

class project -1 with angr capstone

February 26, 2023

```
[1]: import os
import angr
import monkeyhex # this will format numerical results in hexadecimal

filePath = 'gitfm'
proj = angr.Project(filePath)
```

WARNING | 2023-02-26 17:39:42,967 | cle.loader | The main binary is a position-independent executable. It is being loaded with a base address of 0x400000.

```
[ ]:
```

```
[2]: print(proj.loader, '\n\n')

print("Shared Objects:")
for so in (proj.loader.shared_objects):
    print(so)
print()

print(proj.arch)

print('min_addr',proj.loader.min_addr)

print('max_addr',proj.loader.max_addr)

print('main_object',proj.loader.main_object) # we've loaded several binaries_
↳ into this project. Here's the main one!

print('does this binary have an executable stack?: ',proj.loader.main_object.
↳execstack) # sample query: does this binary have an executable stack?

print('is this binary position-independent?',proj.loader.main_object.pic) #_
↳ sample query: is this binary position-independent?
```

<Loaded gitfm, maps [0x400000:0xc07fff]>

Shared Objects:

gitfm
libtinfo.so.6
libc.so.6
ld-linux-x86-64.so.2
extern-address space
cle##tls

<Arch AMD64 (LE)>

min_addr 4194304

max_addr 12615679

main_object <ELF Object gitfm, maps [0x400000:0x42bc87]>

does this binary have an executable stack?: False

is this binary position-independent? True

```
[3]: obj = proj.loader.main_object

print('# The entry point of the object',obj.entry)


print("Retrieve this ELF's segments and sections")

for segment in obj.segments:
    print('segment regions : ',segment)
print('\n\n')

for section in obj.sections:
    print("Section Regions: ",section)

# # You can get an individual segment or section by an address it contains:
print(" You can get an individual segment  by an address it contains:")
print('find_segment_containing(obj.entry)',obj.find_segment_containing(obj.
↪entry))

print('\n')
print(" You can get an individual section by an address it contains:")
```

```
print("obj.find_section_containing(obj.entry)",obj.find_section_containing(obj.  
↪entry))
```

```
# The entry point of the object 4224128  
Retrieve this ELF's segments and sections  
segment regions : <ELFSegment flags=0x4, relro=0x0, vaddr=0x400000,  
memsize=0x37a8, filesize=0x37a8, offset=0x0>  
segment regions : <ELFSegment flags=0x5, relro=0x0, vaddr=0x404000,  
memsize=0x1b04d, filesize=0x1b04d, offset=0x4000>  
segment regions : <ELFSegment flags=0x4, relro=0x0, vaddr=0x420000,  
memsize=0x7340, filesize=0x7340, offset=0x20000>  
segment regions : <ELFSegment flags=0x4, relro=0x1, vaddr=0x428930,  
memsize=0x6d0, filesize=0x6d0, offset=0x27930>  
segment regions : <ELFSegment flags=0x6, relro=0x0, vaddr=0x429000,  
memsize=0x2c88, filesize=0x1830, offset=0x28000>
```

```
Section Regions: <Unnamed | offset 0x0, vaddr 0x400000, size 0x0>  
Section Regions: <.interp | offset 0x318, vaddr 0x400318, size 0x1c>  
Section Regions: <.note.gnu.property | offset 0x338, vaddr 0x400338, size 0x30>  
Section Regions: <.note.gnu.build-id | offset 0x368, vaddr 0x400368, size 0x24>  
Section Regions: <.note.ABI-tag | offset 0x38c, vaddr 0x40038c, size 0x20>  
Section Regions: <.gnu.hash | offset 0x3b0, vaddr 0x4003b0, size 0x60>  
Section Regions: <.dynsym | offset 0x410, vaddr 0x400410, size 0xcc0>  
Section Regions: <.dynstr | offset 0x10d0, vaddr 0x4010d0, size 0x4b9>  
Section Regions: <.gnu.version | offset 0x158a, vaddr 0x40158a, size 0x110>  
Section Regions: <.gnu.version_r | offset 0x16a0, vaddr 0x4016a0, size 0xb0>  
Section Regions: <.rela.dyn | offset 0x1750, vaddr 0x401750, size 0x1548>  
Section Regions: <.rela.plt | offset 0x2c98, vaddr 0x402c98, size 0xb10>  
Section Regions: <.init | offset 0x4000, vaddr 0x404000, size 0x1b>  
Section Regions: <.plt | offset 0x4020, vaddr 0x404020, size 0x770>  
Section Regions: <.plt.got | offset 0x4790, vaddr 0x404790, size 0x30>  
Section Regions: <.plt.sec | offset 0x47c0, vaddr 0x4047c0, size 0x760>  
Section Regions: <.text | offset 0x4f20, vaddr 0x404f20, size 0x1a11e>  
Section Regions: <.fini | offset 0x1f040, vaddr 0x41f040, size 0xd>  
Section Regions: <.rodata | offset 0x20000, vaddr 0x420000, size 0x29ac>  
Section Regions: <.eh_frame_hdr | offset 0x229ac, vaddr 0x4229ac, size 0xb4c>  
Section Regions: <.eh_frame | offset 0x234f8, vaddr 0x4234f8, size 0x3e48>  
Section Regions: <.init_array | offset 0x27930, vaddr 0x428930, size 0x8>  
Section Regions: <.fini_array | offset 0x27938, vaddr 0x428938, size 0x8>  
Section Regions: <.data.rel.ro | offset 0x27940, vaddr 0x428940, size 0xc0>  
Section Regions: <.dynamic | offset 0x27a00, vaddr 0x428a00, size 0x200>  
Section Regions: <.got | offset 0x27c00, vaddr 0x428c00, size 0x400>  
Section Regions: <.data | offset 0x28000, vaddr 0x429000, size 0x1830>  
Section Regions: <.bss | offset 0x29830, vaddr 0x42a840, size 0x1448>  
Section Regions: <.comment | offset 0x29830, vaddr 0x400000, size 0x2b>
```

```

Section Regions: <.debug_aranges | offset 0x2985b, vaddr 0x400000, size 0x4d0>
Section Regions: <.debug_info | offset 0x29d2b, vaddr 0x400000, size 0x2f2cc>
Section Regions: <.debug_abbrev | offset 0x58ff7, vaddr 0x400000, size 0x5669>
Section Regions: <.debug_line | offset 0x5e660, vaddr 0x400000, size 0x15abd>
Section Regions: <.debug_str | offset 0x7411d, vaddr 0x400000, size 0x40bd>
Section Regions: <.debug_line_str | offset 0x781da, vaddr 0x400000, size 0x452>
Section Regions: <.debug_loclists | offset 0x7862c, vaddr 0x400000, size
0x17cd5>
Section Regions: <.debug_rnglists | offset 0x90301, vaddr 0x400000, size
0x28be>
Section Regions: <.symtab | offset 0x92bc0, vaddr 0x400000, size 0x44b8>
Section Regions: <.strtab | offset 0x97078, vaddr 0x400000, size 0x2a76>
Section Regions: <.shstrtab | offset 0x99aee, vaddr 0x400000, size 0x197>
You can get an individual segment by an address it contains:
find_segment_containing(obj.entry) <ELFSegment flags=0x5, relro=0x0,
vaddr=0x404000, memsize=0x1b04d, filesize=0x1b04d, offset=0x4000>

```

```

You can get an individual section by an address it contains:
obj.find_section_containing(obj.entry) <.text | offset 0x4f20, vaddr 0x404f20,
size 0x1a11e>

```

```

[4]: import os

fh = open(filePath, 'rb')
bin_bytearray = bytearray(fh.read())

```

```

[5]: ### SET OF ALL CONDITIONAL JUMPS TAKEN FROM ONLINE SOURCE

JUMP_OPCODES = [
    ↪['JO', 'JNO', 'JS', 'JNS', 'JE', 'JZ', 'JNE', 'JNZ', 'JB', 'JNAE', 'JC', 'JNB', 'JAE', 'JNC', 'JBE', 'JNA',
    'JNBE', 'JL', 'JNGE', 'JGE', 'JNL', 'JLE', 'JNG', 'JG', 'JNLE', 'JP', 'JPE', 'JNP', 'JPO', 'JCXZ', 'JECXZ']

```

```

[6]: # some functions for converting lists

def int_list_to_hex_list(int_list):
    hex_list = []
    for i in int_list:
        hex_list.append(hex(i))
    return hex_list

def hex_list_to_int_list(hex_list):
    int_list = []
    for h in hex_list:
        int_list.append(int(h,0))
    return int_list

```

```

[7]: #took inspiration from capstone github sample code

from capstone import *
from pprint import pprint

code_section = obj.find_section_containing(obj.entry)
textSectionStart = code_section.offset #int(code_section.offset,0)
textSectionLen = code_section.filesize#int(code_section.filesize,0)

textSectionEnd = textSectionStart+textSectionLen

address_instruction_info_dict = {}

print('Start: ',hex(code_section.offset) , "end: ", hex(textSectionEnd))

md = Cs(CS_ARCH_X86, CS_MODE_64)

visited_addresses = []
entry_addresses = []
address_and_instructions = []

entry_addresses.append(textSectionStart)
counter = 0

#will loop as long as there are some unexplored entrypoints in the list
↳entry_addresses
while len(entry_addresses)>0:

    counter+=1

    current_entry_address = entry_addresses.pop(0) #take one entry point

    if current_entry_address in visited_addresses: #if it was already visited,
↳continue
        continue

```



```
print('\t\t->->-> indirect_')
↳ JMP')
break
```

```

Start: 0x4f20 end: 0x1f03e
0x4f20: mov     byte ptr [rsp + 8], 0
0x4f25: jmp     0x107fa
0x107fa
0x4f2a: call    0x4850
0x4f2f: nop
0x4f30: endbr64
0x4f34: push    r15
0x4f36: push    r14
0x4f38: lea     r14, [rip + 0x1b669]
0x4f3f: push    r13
0x4f41: lea     r13, [rip + 0x1b2b4]
0x4f48: push    r12
0x4f4a: push    rbp
0x4f4b: mov     ebp, edi
0x4f4d: push    rbx
0x4f4e: mov     rbx, rsi
0x4f51: sub     rsp, 0x88
0x4f58: mov     rax, qword ptr fs:[0x28]
0x4f61: mov     qword ptr [rsp + 0x78], rax
0x4f66: xor     eax, eax
0x4f68: mov     dword ptr [rsp + 0x34], 0
0x4f70: mov     qword ptr [rsp + 0x50], 0
0x4f79: mov     qword ptr [rsp + 0x58], 0
0x4f82: call    0x18c50
0x4f87: lea     rsi, [rip + 0x1d457]
0x4f8e: mov     edi, 6
0x4f93: call    0x4d00
0x4f98: mov     rax, qword ptr [rbx]
0x4f9b: lea     rdi, [rip + 0x1b5be]
0x4fa2: mov     qword ptr [rip + 0x2593f], rax
0x4fa9: call    0x47e0
0x4fae: lea     rdx, [rip + 0x1c729]
0x4fb5: test    rax, rax
0x4fb8: cmovbe rax, rdx
0x4fbc: mov     qword ptr [rip + 0x2592d], rax
0x4fc3: xor     eax, eax
0x4fc5: call    0x17660
0x4fca: xor     eax, eax
0x4fcc: call    0x18420
0x4fd1: xor     eax, eax
0x4fd3: call    0x18160

```

```

0x4fd8: mov     edi, 0x25
0x4fdd: call    0x1d670
0x4fe2: lea     r8, [rip + 0x1b57c]
0x4fe9: mov     esi, 1
0x4fee: lea     rcx, [rip + 0x1b586]
0x4ff5: mov     rdi, rax
0x4ff8: mov     qword ptr [rip + 0x258d1], rax
0x4fff: xor     eax, eax
0x5001: mov     rdx, -1
0x5008: call    0x4f10
0x500d: lea     rdi, [rip + 0x1b571]
0x5014: call    0x47e0
0x5019: mov     dword ptr [rsp], 1
0x5020: cmp     rax, 1
0x5024: sbb     r12d, r12d
0x5027: xor     r15d, r15d
0x502a: or      r12d, 1
0x502e: mov     rdx, r14
0x5031: mov     rsi, rbx
0x5034: mov     edi, ebp
0x5036: call    0x4de0
0x503b: cmp     eax, -1
0x503e: je      0x510e
0x5044: sub     eax, 0x3f
0x5047: cmp     eax, 0x37
0x504a: ja      0x5057
0x504c: movsxd  rax, dword ptr [r13 + rax*4]
0x5051: add     rax, r13
0x5054: jmp     rax
rax
->->->                                indirect JMP

0x107fa: mov     r15, qword ptr [rax]
0x107fd: mov     rdi, r15
0x10800: call    0x49b0
0x10805: movsxd  rdx, ebp
0x10808: mov     rbp, rdx
0x1080b: sub     rbp, r14
0x1080e: cmp     rax, rbp
0x10811: jbe     0x1082b
0x10813: movsxd  rcx, dword ptr [r13 + 0x74]
0x10817: lea     rsi, [rbp + rcx]
0x1081c: cmp     rax, rsi
0x1081f: jb      0x10a90
0x10825: sub     rax, rcx
0x10828: add     r15, rcx
0x1082b: cmp     rbp, rax
0x1082e: mov     rsi, r15
0x10831: cmova   rbp, rax

```



```

0x10835:    mov     rax, qword ptr [r13 + 0x20]
0x10839:    mov     rdx, rbp
0x1083c:    lea     rdi, [rax + 1]
0x10840:    call    0x4bb0
0x10845:    mov     rax, qword ptr [r13 + 0x20]
0x10849:    mov     rsi, rbp
0x1084c:    lea     rdi, [rax + 1]
0x10850:    call    0x184d0
0x10855:    movzx   ecx, byte ptr [rsp + 8]
0x1085a:    test    cl, cl
0x1085c:    je      0x10866
0x1085e:    mov     rax, qword ptr [r13 + 0x20]
0x10862:    mov     byte ptr [rax + rbp + 1], cl
0x10866:    mov     eax, dword ptr [r13 + 0x64]
0x1086a:    cmp     eax, 0x27
0x1086d:    jle     0x108a0
0x1086f:    mov     edx, dword ptr [r13 + 0x80]
0x10876:    cmp     edx, 4
0x10879:    je      0x108a0
0x1087b:    jg      0x109e0
0x10881:    cmp     edx, 1
0x10884:    jg      0x109c0
0x1088a:    test    edx, edx
0x1088c:    jns     0x10c30
0x10892:    lea     rdi, [rip + 0x10953]
0x10899:    call    0x8b00
0x1089e:    nop
0x108a0:    mov     rax, qword ptr [r13 + 0x38]
0x108a4:    cmp     byte ptr [rax + r12 + 0x49], 0
0x108aa:    je      0x108b9
0x108ac:    movsxd  rax, dword ptr [r13 + 0x64]
0x108b0:    mov     rdx, qword ptr [r13 + 0x20]
0x108b4:    mov     byte ptr [rdx + rax - 3], 0x2a
0x108b9:    cmp     ebx, dword ptr [r13 + 0x28]
0x108bd:    je      0x10948
0x108c3:    mov     rax, qword ptr [r13 + 0x38]
0x108c7:    mov     ecx, dword ptr [rip + 0x1938f]
0x108cd:    movzx   edx, byte ptr [rax + r12 + 0x49]
0x108d3:    test    ecx, ecx
0x108d5:    je      0x108e8
0x108d7:    add     rax, r12
0x108da:    cmp     byte ptr [rax + 0x48], 0
0x108de:    movzx   edx, byte ptr [rax + 0x49]
0x108e2:    jne     0x1097e
0x108e8:    test    dl, dl
0x108ea:    jne     0x109a0
0x108f0:    mov     esi, dword ptr [rip + 0x19a46]
0x108f6:    mov     r9d, dword ptr [rip + 0x19a27]

```

```

0x108fd:    mov     edi, dword ptr [rip + 0x19a3d]
0x10903:    mov     edx, r9d
0x10906:    call    0xa570
0x1090b:    sub     ebx, dword ptr [r13 + 0x2c]
0x1090f:    mov     rdi, qword ptr [r13 + 8]
0x10913:    mov     edx, 1
0x10918:    lea     esi, [rbx + 1]
0x1091b:    call    0xc680
0x10920:    mov     eax, dword ptr [r13 + 0x64]
0x10924:    mov     rsi, qword ptr [r13 + 0x20]
0x10928:    mov     rdi, qword ptr [r13 + 8]
0x1092c:    add     rsp, 0x28
0x10930:    pop     rbx
0x10931:    lea     edx, [rax - 2]
0x10934:    pop     rbp
0x10935:    pop     r12
0x10937:    pop     r13
0x10939:    pop     r14
0x1093b:    pop     r15
0x1093d:    jmp     0xc600
0xc600:
0x10942:    nop     word ptr [rax + rax]
0x10948:    cmp     byte ptr [r13 + 0x6c], 1
0x1094d:    mov     rdx, qword ptr [r13 + 0x20]
0x10951:    sbb     eax, eax
0x10953:    and     eax, 0xffffffffec
0x10956:    add     eax, 0x3e
0x10959:    mov     byte ptr [rdx], al
0x1095b:    mov     rax, qword ptr [r13 + 0x38]
0x1095f:    mov     esi, dword ptr [rip + 0x192f7]
0x10965:    add     rax, r12
0x10968:    test    esi, esi
0x1096a:    je      0x10a50
0x10970:    cmp     byte ptr [rax + 0x48], 0
0x10974:    movzx   edx, byte ptr [rax + 0x49]
0x10978:    je      0x10a50
0x1097e:    test    dl, dl
0x10980:    je      0x10b10
0x10986:    cmp     ebx, dword ptr [r13 + 0x28]
0x1098a:    jne     0x109a0
0x1098c:    cmp     byte ptr [r13 + 0x6c], 1
0x10991:    je      0x10c08
0x10997:    nop     word ptr [rax + rax]
0x109a0:    mov     esi, dword ptr [rip + 0x1998e]
0x109a6:    mov     r9d, dword ptr [rip + 0x19977]
0x109ad:    mov     edi, dword ptr [rip + 0x19985]
0x109b3:    jmp     0x10903
0x10903:

```

```

0x109b8:      nop      dword ptr [rax + rax]
0x109c0:      mov      ecx, 0xb
0x109c5:      mov      esi, ebx
0x109c7:      mov      rdi, r13
0x109ca:      xor      eax, eax
0x109cc:      call     0x10420
0x109d1:      jmp      0x108a0
0x108a0
0x109d6:      nop      word ptr cs:[rax + rax]
0x109e0:      cmp      edx, 5
0x109e3:      jne      0x10892
0x109e9:      cmp      eax, 0x49
0x109ec:      jle      0x108a0
0x109f2:      xor      edx, edx
0x109f4:      mov      esi, ebx
0x109f6:      mov      rdi, r13
0x109f9:      xor      eax, eax
0x109fb:      mov      ecx, 0x36
0x10a00:      call     0x10420
0x10a05:      mov      esi, ebx
0x10a07:      mov      rdi, r13
0x10a0a:      xor      eax, eax
0x10a0c:      mov      ecx, 0x26
0x10a11:      mov      edx, 1
0x10a16:      call     0x10420
0x10a1b:      mov      esi, ebx
0x10a1d:      mov      rdi, r13
0x10a20:      xor      eax, eax
0x10a22:      mov      ecx, 0x16
0x10a27:      mov      edx, 2
0x10a2c:      call     0x10420
0x10a31:      mov      ecx, 0xb
0x10a36:      mov      esi, ebx
0x10a38:      mov      rdi, r13
0x10a3b:      mov      edx, 3
0x10a40:      xor      eax, eax
0x10a42:      call     0x10420
0x10a47:      jmp      0x108a0
0x108a0
0x10a4c:      nop      dword ptr [rax]
0x10a50:      movzx    edx, byte ptr [rax + 0x49]
0x10a54:      cmp      ebx, dword ptr [r13 + 0x28]
0x10a58:      jne      0x108e8
0x10a5e:      cmp      byte ptr [r13 + 0x6c], 1
0x10a63:      jne      0x108e8
0x10a69:      mov      esi, dword ptr [rip + 0x198d9]
0x10a6f:      test     dl, dl
0x10a71:      jne      0x10c08

```

```

0x10a77:      mov     r9d, dword ptr [rip + 0x198ce]
0x10a7e:      test    dl, dl
0x10a80:      je      0x108fd
0x10a86:      jmp     0x109ad
0x109ad:
0x10a8b:      nop     dword ptr [rax + rax]
0x10a90:      add     r14, rax
0x10a93:      mov     rcx, rdx
0x10a96:      sub     rcx, r14
0x10a99:      sub     r14, rdx
0x10a9c:      add     rax, rcx
0x10a9f:      add     r15, r14
0x10aa2:      jmp     0x1082b
0x1082b:
0x10aa7:      nop     word ptr [rax + rax]
0x10ab0:      mov     byte ptr [rsp + 8], 0x3d
0x10ab5:      add     r14, 1
0x10ab9:      jmp     0x107fa
0x107fa:
0x10abe:      nop
0x10ac0:      mov     byte ptr [rsp + 8], 0x7c
0x10ac5:      jmp     0x10ab5
0x10ab5:
0x10ac7:      nop     word ptr [rax + rax]
0x10ad0:      movzx   ecx, byte ptr [rax + 0x4a]
0x10ad4:      mov     byte ptr [rsp + 8], cl
0x10ad8:      test    cl, cl
0x10ada:      je      0x107fa
0x10ae0:      mov     byte ptr [rsp + 8], 0x2a
0x10ae5:      jmp     0x10ab5
0x10ab5:
0x10ae7:      nop     word ptr [rax + rax]
0x10af0:      mov     byte ptr [rsp + 8], 0x40
0x10af5:      jmp     0x10ab5
0x10ab5:
0x10af7:      nop     word ptr [rax + rax]
0x10b00:      mov     byte ptr [rsp + 8], 0x2f
0x10b05:      jmp     0x10ab5
0x10ab5:
0x10b07:      nop     word ptr [rax + rax]
0x10b10:      mov     rdi, qword ptr [r13 + 8]
0x10b14:      mov     esi, ebx
0x10b16:      sub     esi, dword ptr [r13 + 0x2c]
0x10b1a:      mov     edx, 1
0x10b1f:      add     esi, 1
0x10b22:      call    0xc680
0x10b27:      mov     rax, qword ptr [r13 + 0x38]
0x10b2b:      cmp     byte ptr [rax + r12 + 0x49], 0

```

```

0x10b31:    jne     0x10c50
0x10b37:    mov     eax, dword ptr [rip + 0x197ff]
0x10b3d:    mov     r15d, dword ptr [rip + 0x197fc]
0x10b44:    mov     dword ptr [rsp + 0xc], eax
0x10b48:    cmp     ebx, dword ptr [r13 + 0x28]
0x10b4c:    je      0x10c70
0x10b52:    mov     eax, dword ptr [rip + 0x197cc]
0x10b58:    mov     dword ptr [rsp + 8], eax
0x10b5c:    mov     edx, dword ptr [rsp + 8]
0x10b60:    mov     esi, dword ptr [rsp + 0xc]
0x10b64:    mov     edi, r15d
0x10b67:    call    0xa570
0x10b6c:    mov     rax, qword ptr [r13 + 0x20]
0x10b70:    mov     rdi, qword ptr [r13 + 8]
0x10b74:    movsx   esi, byte ptr [rax]
0x10b77:    call    0xc650
0x10b7c:    mov     rax, qword ptr [r13 + 0x38]
0x10b80:    lea     rdx, [rax + r12]
0x10b84:    cmp     byte ptr [rdx + 0x4b], 0
0x10b88:    je      0x10c90
0x10b8e:    movsx   edi, byte ptr [rdx + 0x4e]
0x10b92:    cmp     dil, 2
0x10b96:    je      0x10bb6
0x10b98:    movsx   esi, byte ptr [rdx + 0x4c]
0x10b9c:    cmp     ebx, dword ptr [r13 + 0x28]
0x10ba0:    jne     0x10bad
0x10ba2:    cmp     byte ptr [r13 + 0x6c], 1
0x10ba7:    je      0x10d52
0x10bad:    movsx   edx, byte ptr [rdx + 0x4d]
0x10bb1:    call    0xa570
0x10bb6:    mov     rax, qword ptr [r13 + 0x20]
0x10bba:    mov     rdi, qword ptr [r13 + 8]
0x10bbe:    lea     edx, [rbp + 1]
0x10bc1:    lea     rsi, [rax + 1]
0x10bc5:    call    0xc600
0x10bca:    mov     eax, dword ptr [r13 + 0x64]
0x10bce:    mov     rdi, qword ptr [r13 + 8]
0x10bd2:    lea     edx, [rax - 4]
0x10bd5:    mov     rax, qword ptr [r13 + 0x20]
0x10bd9:    sub     edx, ebp
0x10bdb:    lea     rsi, [rax + rbp + 2]
0x10be0:    call    0xc600
0x10be5:    mov     edx, dword ptr [rsp + 8]
0x10be9:    mov     esi, dword ptr [rsp + 0xc]
0x10bed:    mov     edi, r15d
0x10bf0:    add     rsp, 0x28
0x10bf4:    pop     rbx
0x10bf5:    pop     rbp

```

```

0x10bf6:    pop     r12
0x10bf8:    pop     r13
0x10bfa:    pop     r14
0x10bfc:    pop     r15
0x10bfe:    jmp     0xa570
0xa570
0x10c03:    nop     dword ptr [rax + rax]
0x10c08:    mov     esi, dword ptr [rip + 0x19736]
0x10c0e:    jmp     0x10a77
0x10a77
0x10c13:    nop     dword ptr [rax + rax]
0x10c18:    cmp     byte ptr [rdx + 1], 0
0x10c1c:    jne     0x107fa
0x10c22:    jmp     0x107c6
0x107c6
0x10c27:    nop     word ptr [rax + rax]
0x10c30:    mov     ecx, 0x10
0x10c35:    mov     esi, ebx
0x10c37:    mov     rdi, r13
0x10c3a:    xor     eax, eax
0x10c3c:    call    0x10420
0x10c41:    jmp     0x108a0
0x108a0
0x10c46:    nop     word ptr cs:[rax + rax]
0x10c50:    mov     eax, dword ptr [rip + 0x196de]
0x10c56:    mov     r15d, dword ptr [rip + 0x196db]
0x10c5d:    mov     dword ptr [rsp + 0xc], eax
0x10c61:    jmp     0x10b48
0x10b48
0x10c66:    nop     word ptr cs:[rax + rax]
0x10c70:    cmp     byte ptr [r13 + 0x6c], 1
0x10c75:    jne     0x10b52
0x10c7b:    mov     eax, dword ptr [rip + 0x196cb]
0x10c81:    mov     dword ptr [rsp + 8], eax
0x10c85:    jmp     0x10b5c
0x10b5c
0x10c8a:    nop     word ptr [rax + rax]
0x10c90:    mov     r14, qword ptr [rip + 0x1adf9]
0x10c97:    mov     byte ptr [rdx + 0x4b], 1
0x10c9b:    test    r14, r14
0x10c9e:    je      0x10d49
0x10ca4:    mov     qword ptr [rsp + 0x18], 0
0x10cad:    mov     qword ptr [rsp + 0x10], 0
0x10cb6:    nop     word ptr cs:[rax + rax]
0x10cc0:    mov     rsi, qword ptr [rax + r12]
0x10cc4:    mov     rdi, qword ptr [r14]
0x10cc7:    mov     edx, 0x15
0x10ccc:    call    0x4a40

```

```

0x10cd1:    test    eax, eax
0x10cd3:    jne     0x10cee
0x10cd5:    mov     rdi, qword ptr [r14]
0x10cd8:    call    0x49b0
0x10cdd:    cmp     qword ptr [rsp + 0x10], rax
0x10ce2:    jae     0x10cee
0x10ce4:    mov     qword ptr [rsp + 0x18], r14
0x10ce9:    mov     qword ptr [rsp + 0x10], rax
0x10cee:    mov     r14, qword ptr [r14 + 0x18]
0x10cf2:    mov     rax, qword ptr [r13 + 0x38]
0x10cf6:    test    r14, r14
0x10cf9:    jne     0x10cc0
0x10cfb:    mov     rsi, qword ptr [rsp + 0x18]
0x10d00:    lea     rdx, [rax + r12]
0x10d04:    test    rsi, rsi
0x10d07:    je      0x10d49
0x10d09:    mov     eax, dword ptr [rsi + 0x10]
0x10d0c:    xor     ecx, ecx
0x10d0e:    cmp     eax, -1
0x10d11:    cmovne  eax, ecx
0x10d14:    mov     byte ptr [rdx + 0x4e], al
0x10d17:    movsx   edi, al
0x10d1a:    mov     eax, dword ptr [rsi + 8]
0x10d1d:    mov     ecx, eax
0x10d1f:    cmp     eax, -1
0x10d22:    movzx   eax, byte ptr [rip + 0x19613]
0x10d29:    cmovne  eax, ecx
0x10d2c:    mov     byte ptr [rdx + 0x4c], al
0x10d2f:    mov     eax, dword ptr [rsi + 0xc]
0x10d32:    mov     ecx, eax
0x10d34:    cmp     eax, -1
0x10d37:    movzx   eax, byte ptr [rip + 0x195e6]
0x10d3e:    cmovne  eax, ecx
0x10d41:    mov     byte ptr [rdx + 0x4d], al
0x10d44:    jmp     0x10b92
0x10b92
0x10d49:    mov     byte ptr [rdx + 0x4e], 2
0x10d4d:    jmp     0x10bb6
0x10bb6
0x10d52:    mov     edx, dword ptr [rsp + 8]
0x10d56:    mov     edi, r15d
0x10d59:    call    0xa570
0x10d5e:    jmp     0x10bb6
0x10bb6
0x10d63:    nop     word ptr cs:[rax + rax]
0x10d6e:    nop
0x10d70:    push    r13
0x10d72:    push    r12

```

```

0x10d74:    push    rbp
0x10d75:    push    rbx
0x10d76:    mov     rbx, rdi
0x10d79:    sub     rsp, 0x18
0x10d7d:    mov     rax, qword ptr fs:[0x28]
0x10d86:    mov     qword ptr [rsp + 8], rax
0x10d8b:    xor     eax, eax
0x10d8d:    lea     r12, [rsp + 7]
0x10d92:    mov     rdi, r12
0x10d95:    call    0xa6a0
0x10d9a:    mov     ebp, dword ptr [rbx + 0x2c]
0x10d9d:    mov     eax, ebp
0x10d9f:    cmp     ebp, dword ptr [rbx + 0x68]
0x10da2:    jl      0x10dcB
0x10da4:    jmp     0x10dd9
0x10dd9:
0x10da6:    nop     word ptr cs:[rax + rax]
0x10db0:    cmp     byte ptr [rbx + 0x6d], 0
0x10db4:    je      0x10dc0
0x10db6:    mov     esi, ebp
0x10db8:    mov     rdi, rbx
0x10dbb:    call    0x10760
0x10dc0:    add     ebp, 1
0x10dc3:    cmp     ebp, dword ptr [rbx + 0x68]
0x10dc6:    jge     0x10dd9
0x10dc8:    mov     eax, dword ptr [rbx + 0x2c]
0x10dcb:    mov     edx, ebp
0x10dcd:    sub     edx, eax
0x10dcf:    mov     eax, dword ptr [rbx + 0x60]
0x10dd2:    sub     eax, 2
0x10dd5:    cmp     edx, eax
0x10dd7:    jl      0x10db0
0x10dd9:    mov     edx, dword ptr [rip + 0x19545]
0x10ddf:    xor     edi, edi
0x10de1:    mov     esi, 7
0x10de6:    call    0xa570
0x10deb:    movsxd  rdx, dword ptr [rbx + 0x64]
0x10def:    mov     rdi, qword ptr [rbx + 0x20]
0x10df3:    mov     esi, 0x20
0x10df8:    call    0x4a50
0x10dfd:    mov     eax, dword ptr [rbx + 0x60]
0x10e00:    lea     r13d, [rax - 2]
0x10e04:    mov     eax, dword ptr [rbx + 0x14]
0x10e07:    cmp     r13d, eax
0x10e0a:    cmovg   r13d, eax
0x10e0e:    cmp     r13d, ebp
0x10e11:    jle     0x10e49
0x10e13:    nop     dword ptr [rax + rax]

```



```

0x10e18:    mov     rdi, qword ptr [rbx + 8]
0x10e1c:    mov     esi, ebp
0x10e1e:    sub     esi, dword ptr [rbx + 0x2c]
0x10e21:    mov     edx, 1
0x10e26:    add     esi, 1
0x10e29:    add     ebp, 1
0x10e2c:    call    0xc680
0x10e31:    mov     eax, dword ptr [rbx + 0x64]
0x10e34:    mov     rsi, qword ptr [rbx + 0x20]
0x10e38:    mov     rdi, qword ptr [rbx + 8]
0x10e3c:    lea     edx, [rax - 2]
0x10e3f:    call    0xc600
0x10e44:    cmp     r13d, ebp
0x10e47:    jne     0x10e18
0x10e49:    mov     eax, dword ptr [rbx + 0x68]
0x10e4c:    mov     rdi, r12
0x10e4f:    mov     dword ptr [rbx + 0x14], eax
0x10e52:    call    0xa6b0
0x10e57:    mov     rax, qword ptr [rsp + 8]
0x10e5c:    sub     rax, qword ptr fs:[0x28]
0x10e65:    jne     0x10e72
0x10e67:    add     rsp, 0x18
0x10e6b:    pop     rbx
0x10e6c:    pop     rbp
0x10e6d:    pop     r12
0x10e6f:    pop     r13
0x10e71:    ret
0x10e72:    call    0x49d0
0x10e77:    nop     word ptr [rax + rax]
0x10e80:    endbr64
0x10e84:    cmp     byte ptr [rdi + 0x6d], 0
0x10e88:    jne     0x10e90
0x10e8a:    ret
0x10e8b:    nop     dword ptr [rax + rax]
0x10e90:    xor     eax, eax
0x10e92:    jmp     0x10d70
0x10d70
0x10e97:    nop     word ptr [rax + rax]
0x10ea0:    endbr64
0x10ea4:    cmp     byte ptr [rdi + 0x6d], 0
0x10ea8:    jne     0x10eb0
0x10eaa:    ret
0x10eab:    nop     dword ptr [rax + rax]
0x10eb0:    jmp     0x10760
0x10760
0x10eb5:    nop     word ptr cs:[rax + rax]
0x10ec0:    endbr64
0x10ec4:    push    r13

```

```

0x10ec6:    push    r12
0x10ec8:    push    rbp
0x10ec9:    push    rbx
0x10eca:    mov     rbx, rdi
0x10ecd:    sub     rsp, 0x18
0x10ed1:    movsxd  rdi, dword ptr [rdi + 0x64]
0x10ed5:    mov     rax, qword ptr fs:[0x28]
0x10ede:    mov     qword ptr [rsp + 8], rax
0x10ee3:    xor     eax, eax
0x10ee5:    call    0x1d670
0x10eea:    cmp     byte ptr [rbx + 0x6d], 0
0x10eee:    jne     0x10f10
0x10ef0:    mov     rax, qword ptr [rsp + 8]
0x10ef5:    sub     rax, qword ptr fs:[0x28]
0x10efe:    jne     0x11006
0x10f04:    add     rsp, 0x18
0x10f08:    pop     rbx
0x10f09:    pop     rbp
0x10f0a:    pop     r12
0x10f0c:    pop     r13
0x10f0e:    ret
0x10f0f:    nop
0x10f10:    lea     r13, [rsp + 7]
0x10f15:    mov     r12, rax
0x10f18:    mov     ebp, 1
0x10f1d:    mov     rdi, r13
0x10f20:    call    0xa6a0
0x10f25:    mov     esi, dword ptr [rip + 0x193f5]
0x10f2b:    xor     edi, edi
0x10f2d:    mov     edx, esi
0x10f2f:    call    0xa570
0x10f34:    cmp     dword ptr [rbx + 0x60], 2
0x10f38:    jle     0x10fa4
0x10f3a:    nop     word ptr [rax + rax]
0x10f40:    mov     rdi, qword ptr [rbx + 8]
0x10f44:    xor     edx, edx
0x10f46:    mov     esi, ebp
0x10f48:    add     ebp, 1
0x10f4b:    call    0xc680
0x10f50:    mov     rdi, qword ptr [rbx + 8]
0x10f54:    mov     esi, 0x20
0x10f59:    call    0xc650
0x10f5e:    mov     eax, dword ptr [rbx + 0x60]
0x10f61:    lea     edx, [rax - 1]
0x10f64:    cmp     edx, ebp
0x10f66:    jg      0x10f40
0x10f68:    cmp     eax, 2
0x10f6b:    jle     0x10fa4

```

```

0x10f6d:    mov     ebp, 1
0x10f72:    nop     word ptr [rax + rax]
0x10f78:    mov     eax, dword ptr [rbx + 0x64]
0x10f7b:    mov     rdi, qword ptr [rbx + 8]
0x10f7f:    mov     esi, ebp
0x10f81:    add     ebp, 1
0x10f84:    lea     edx, [rax - 1]
0x10f87:    call    0xc680
0x10f8c:    mov     rdi, qword ptr [rbx + 8]
0x10f90:    mov     esi, 0x20
0x10f95:    call    0xc650
0x10f9a:    mov     eax, dword ptr [rbx + 0x60]
0x10f9d:    sub     eax, 1
0x10fa0:    cmp     eax, ebp
0x10fa2:    jg      0x10f78
0x10fa4:    movsxd  rdx, dword ptr [rbx + 0x64]
0x10fa8:    mov     rdi, r12
0x10fab:    mov     esi, 0x20
0x10fb0:    call    0x4a50
0x10fb5:    mov     rdi, qword ptr [rbx + 8]
0x10fb9:    xor     edx, edx
0x10fbb:    xor     esi, esi
0x10fbd:    call    0xc680
0x10fc2:    mov     edx, dword ptr [rbx + 0x64]
0x10fc5:    mov     rdi, qword ptr [rbx + 8]
0x10fc9:    mov     rsi, r12
0x10fcc:    call    0xc600
0x10fd1:    mov     eax, dword ptr [rbx + 0x60]
0x10fd4:    mov     rdi, qword ptr [rbx + 8]
0x10fd8:    xor     edx, edx
0x10fda:    lea     esi, [rax - 1]
0x10fdd:    call    0xc680
0x10fe2:    mov     edx, dword ptr [rbx + 0x64]
0x10fe5:    mov     rdi, qword ptr [rbx + 8]
0x10fe9:    mov     rsi, r12
0x10fec:    call    0xc600
0x10ff1:    mov     rdi, r12
0x10ff4:    call    0x18d10
0x10ff9:    mov     rdi, r13
0x10ffc:    call    0xa6b0
0x11001:    jmp     0x10ef0
0x10ef0:
0x11006:    call    0x49d0
0x1100b:    nop     dword ptr [rax + rax]
0x11010:    endbr64
0x11014:    push    rbp
0x11015:    xor     eax, eax
0x11017:    mov     rbp, rdi

```

```

0x1101a:    call    0x10ec0
0x1101f:    xor     eax, eax
0x11021:    mov     rdi, rbp
0x11024:    call    0x10160
0x11029:    cmp     byte ptr [rbp + 0x6d], 0
0x1102d:    je      0x11039
0x1102f:    mov     rdi, rbp
0x11032:    xor     eax, eax
0x11034:    call    0xec50
0x11039:    xor     eax, eax
0x1103b:    mov     rdi, rbp
0x1103e:    call    0x10240
0x11043:    cmp     byte ptr [rbp + 0x6d], 0
0x11047:    jne     0x11050
0x11049:    pop     rbp
0x1104a:    ret
0x1104b:    nop     dword ptr [rax + rax]
0x11050:    mov     rdi, rbp
0x11053:    xor     eax, eax
0x11055:    pop     rbp
0x11056:    jmp     0x10d70
0x10d70:
0x1105b:    nop     dword ptr [rax + rax]
0x11060:    endbr64
0x11064:    mov     edx, dword ptr [rdi + 0x68]
0x11067:    test    edx, edx
0x11069:    jle     0x110ab
0x1106b:    sub     edx, 1
0x1106e:    mov     rcx, qword ptr [rdi + 0x38]
0x11072:    lea     rdx, [rdx + rdx*4]
0x11076:    shl     rdx, 4
0x1107a:    lea     rax, [rcx + 0x49]
0x1107e:    lea     rdx, [rcx + rdx + 0x99]
0x11086:    nop     word ptr cs:[rax + rax]
0x11090:    cmp     byte ptr [rax], 0
0x11093:    jne     0x110a2
0x11095:    cmp     byte ptr [rax - 1], 0
0x11099:    je      0x110a2
0x1109b:    mov     byte ptr [rax], 1
0x1109e:    add     dword ptr [rdi + 0x70], 1
0x110a2:    add     rax, 0x50
0x110a6:    cmp     rax, rdx
0x110a9:    jne     0x11090
0x110ab:    ret
0x110ac:    nop     dword ptr [rax]
0x110b0:    endbr64
0x110b4:    mov     edx, dword ptr [rdi + 0x68]
0x110b7:    test    edx, edx

```

```

0x110b9:    jle     0x110ec
0x110bb:    sub     edx, 1
0x110be:    mov     rcx, qword ptr [rdi + 0x38]
0x110c2:    lea     rdx, [rdx + rdx*4]
0x110c6:    shl     rdx, 4
0x110ca:    lea     rax, [rcx + 0x49]
0x110ce:    lea     rdx, [rcx + rdx + 0x99]
0x110d6:    nop     word ptr cs:[rax + rax]
0x110e0:    mov     byte ptr [rax], 0
0x110e3:    add     rax, 0x50
0x110e7:    cmp     rax, rdx
0x110ea:    jne     0x110e0
0x110ec:    mov     dword ptr [rdi + 0x70], 0
0x110f3:    ret
0x110f4:    nop     word ptr cs:[rax + rax]
0x110ff:    nop
0x11100:    endbr64
0x11104:    mov     rax, qword ptr [rdi + 0x18]
0x11108:    ret
0x11109:    nop     dword ptr [rax]
0x11110:    endbr64
0x11114:    mov     eax, dword ptr [rip + 0x1a996]
0x1111a:    test    eax, eax
0x1111c:    jne     0x11120
0x1111e:    ret
0x1111f:    nop
0x11120:    push    rbp
0x11121:    xor     eax, eax
0x11123:    push    rbx
0x11124:    sub     rsp, 8
0x11128:    mov     dword ptr [rip + 0x1a97e], 0
0x11132:    call    0xc8c0
0x11137:    mov     edx, 4
0x1113c:    lea     rsi, [rip + 0xf3ef]
0x11143:    lea     rdi, [rip + 0x100af]
0x1114a:    mov     rbp, rax
0x1114d:    call    0x8070
0x11152:    mov     rdi, rbp
0x11155:    mov     ebx, eax
0x11157:    call    0xc970
0x1115c:    xor     eax, eax
0x1115e:    and     ebx, 0xffffffff
0x11161:    call    0xd870
0x11166:    xor     eax, eax
0x11168:    call    0xd740
0x1116d:    xor     eax, eax
0x1116f:    call    0xa820
0x11174:    xor     eax, eax

```

```

0x11176:      cmp     bl, 0x4e
0x11179:      setne  al
0x1117c:      add     rsp, 8
0x11180:      pop     rbx
0x11181:      pop     rbp
0x11182:      ret
0x11183:      nop     word ptr cs:[rax + rax]
0x1118e:      nop
0x11190:      endbr64
0x11194:      push    rbp
0x11195:      mov     rbp, rsi
0x11198:      sub     rsp, 0x130
0x1119f:      mov     rax, qword ptr fs:[0x28]
0x111a8:      mov     qword ptr [rsp + 0x128], rax
0x111b0:      xor     eax, eax
0x111b2:      mov     rsi, rsp
0x111b5:      call    0x18f40
0x111ba:      test    eax, eax
0x111bc:      je       0x111e0
0x111be:      xor     eax, eax
0x111c0:      mov     rdx, qword ptr [rsp + 0x128]
0x111c8:      sub     rdx, qword ptr fs:[0x28]
0x111d1:      jne     0x11217
0x111d3:      add     rsp, 0x130
0x111da:      pop     rbp
0x111db:      ret
0x111dc:      nop     dword ptr [rax]
0x111e0:      lea     rsi, [rsp + 0x90]
0x111e8:      mov     rdi, rbp
0x111eb:      call    0x18f40
0x111f0:      test    eax, eax
0x111f2:      jne     0x111be
0x111f4:      mov     rax, qword ptr [rsp + 0x90]
0x111fc:      cmp     qword ptr [rsp], rax
0x11200:      jne     0x111be
0x11202:      mov     rax, qword ptr [rsp + 0x98]
0x1120a:      cmp     qword ptr [rsp + 8], rax
0x1120f:      sete    al
0x11212:      movzx   eax, al
0x11215:      jmp     0x111c0
0x111c0:
0x11217:      call    0x49d0
0x1121c:      nop     dword ptr [rax]
0x11220:      endbr64
0x11224:      push    rbx
0x11225:      mov     edx, dword ptr [rdi + 0x70]
0x11228:      mov     rbx, rdi
0x1122b:      mov     ecx, 0x1e

```

```

0x11230:    test    edx, edx
0x11232:    je      0x11268
0x11234:    lea     rdx, [rip + 0xffd8]
0x1123b:    lea     rdi, [rip + 0x107de]
0x11242:    xor     eax, eax
0x11244:    call    0xf760
0x11249:    xor     r8d, r8d
0x1124c:    cmp     al, 0x6f
0x1124e:    je      0x1125e
0x11250:    cmp     al, 0x73
0x11252:    je      0x11280
0x11254:    mov     r8d, 2
0x1125a:    cmp     al, 0x61
0x1125c:    je      0x11298
0x1125e:    mov     eax, r8d
0x11261:    pop     rbx
0x11262:    ret
0x11263:    nop     dword ptr [rax + rax]
0x11268:    lea     rdx, [rip + 0xffa9]
0x1126f:    lea     rdi, [rip + 0x107da]
0x11276:    xor     eax, eax
0x11278:    call    0xf760
0x1127d:    jmp     0x11249
0x11249
0x1127f:    nop
0x11280:    cmp     dword ptr [rbx + 0x70], 1
0x11284:    mov     r8d, 2
0x1128a:    pop     rbx
0x1128b:    sbb     r8d, -1
0x1128f:    mov     eax, r8d
0x11292:    ret
0x11293:    nop     dword ptr [rax + rax]
0x11298:    mov     eax, dword ptr [rbx + 0x70]
0x1129b:    test    eax, eax
0x1129d:    je      0x1125e
0x1129f:    mov     dword ptr [rbx + 0x90], 0
0x112a9:    xor     r8d, r8d
0x112ac:    pop     rbx
0x112ad:    mov     eax, r8d
0x112b0:    ret
0x112b1:    nop     word ptr cs:[rax + rax]
0x112bc:    nop     dword ptr [rax]
0x112c0:    endbr64
0x112c4:    push    rbp
0x112c5:    mov     rbp, rdi
0x112c8:    sub     rsp, 0xa0
0x112cf:    mov     rax, qword ptr fs:[0x28]
0x112d8:    mov     qword ptr [rsp + 0x98], rax

```

```

0x112e0:    xor     eax, eax
0x112e2:    mov     rsi, rsp
0x112e5:    call    0x18f40
0x112ea:    test    eax, eax
0x112ec:    jne     0x112fe
0x112ee:    mov     eax, dword ptr [rsp + 0x18]
0x112f2:    and     eax, 0xf000
0x112f7:    cmp     eax, 0x8000
0x112fc:    je      0x11320
0x112fe:    mov     rax, qword ptr [rsp + 0x98]
0x11306:    sub     rax, qword ptr fs:[0x28]
0x1130f:    jne     0x1132a
0x11311:    add     rsp, 0xa0
0x11318:    pop     rbp
0x11319:    ret
0x1131a:    nop     word ptr [rax + rax]
0x11320:    mov     rdi, rbp
0x11323:    call    0x4870
0x11328:    jmp     0x112fe
0x112fe:
0x1132a:    call    0x49d0
0x1132f:    nop
0x11330:    endbr64
0x11334:    push    r15
0x11336:    push    r14
0x11338:    mov     r14, rdx
0x1133b:    push    r13
0x1133d:    push    r12
0x1133f:    mov     r12d, ecx
0x11342:    push    rbp
0x11343:    and     r12d, 0xf000
0x1134a:    mov     rbp, rdi
0x1134d:    push    rbx
0x1134e:    mov     ebx, ecx
0x11350:    sub     rsp, 0x178
0x11357:    mov     qword ptr [rsp + 8], rsi
0x1135c:    mov     dword ptr [rsp + 0x30], r8d
0x11361:    mov     dword ptr [rsp + 0x34], r9d
0x11366:    mov     rax, qword ptr fs:[0x28]
0x1136f:    mov     qword ptr [rsp + 0x168], rax
0x11377:    xor     eax, eax
0x11379:    cmp     r12d, 0xa000
0x11380:    je      0x11a10
0x11386:    cmp     r12d, 0x4000
0x1138d:    je      0x11828
0x11393:    mov     rdi, r14
0x11396:    call    0x49b0
0x1139b:    mov     rdi, r14

```



```

0x1139e:      mov     r13, rax
0x113a1:      call    0x1d7a0
0x113a6:      lea     rsi, [rsp + 0x40]
0x113ab:      mov     rdi, rax
0x113ae:      mov     qword ptr [rsp + 0x18], rax
0x113b3:      call    0x18f40
0x113b8:      test    eax, eax
0x113ba:      je      0x11630
0x113c0:      mov     rdi, qword ptr [rsp + 0x18]
0x113c5:      call    0x190c0
0x113ca:      cmp     byte ptr [rax], 0
0x113cd:      mov     r13, rax
0x113d0:      je      0x11ae5
0x113d6:      mov     esi, dword ptr [rbp + 0x90]
0x113dc:      test    esi, esi
0x113de:      je      0x113f4
0x113e0:      mov     rdi, qword ptr [rsp + 0x18]
0x113e5:      xor     esi, esi
0x113e7:      call    0x4d90
0x113ec:      test    eax, eax
0x113ee:      je      0x11a60
0x113f4:      mov     rdi, qword ptr [rsp + 8]
0x113f9:      call    0x49b0
0x113fe:      lea     rdi, [rax + 0x20]
0x11402:      call    0x1d670
0x11407:      mov     r8, qword ptr [rsp + 8]
0x1140c:      lea     rcx, [rip + 0xfe34]
0x11413:      mov     r13, rax
0x11416:      cmp     r12d, 0x8000
0x1141d:      je      0x11a48
0x11423:      mov     rdx, -1
0x1142a:      mov     esi, 1
0x1142f:      mov     rdi, r13
0x11432:      xor     eax, eax
0x11434:      call    0x4f10
0x11439:      mov     edx, 1
0x1143e:      mov     esi, 1
0x11443:      mov     rdi, r13
0x11446:      call    0xe480
0x1144b:      xor     eax, eax
0x1144d:      call    0xa820
0x11452:      mov     rdi, r13
0x11455:      call    0x18d10
0x1145a:      mov     rdi, qword ptr [rsp + 8]
0x1145f:      xor     esi, esi
0x11461:      xor     eax, eax
0x11463:      call    0x4d30
0x11468:      mov     r15d, eax

```

```

0x1146b:    cmp     eax, -1
0x1146e:    je      0x11a90
0x11474:    mov     ecx, dword ptr [rbp + 0x10]
0x11477:    test    ecx, ecx
0x11479:    je      0x1147e
0x1147b:    and     ebx, 0xffffffffb6
0x1147e:    mov     rdi, qword ptr [rsp + 0x18]
0x11483:    mov     esi, ebx
0x11485:    call    0x4e00
0x1148a:    mov     edi, r15d
0x1148d:    mov     dword ptr [rsp + 0x28], eax
0x11491:    cmp     eax, -1
0x11494:    je      0x11ae0
0x1149a:    call    0x182f0
0x1149f:    mov     qword ptr [rsp + 0x10], 0x8000
0x114a8:    mov     r13, rax
0x114ab:    cmp     rax, 0x8000
0x114b1:    jle     0x11b38
0x114b7:    mov     eax, ebx
0x114b9:    and     ebx, 0xb000
0x114bf:    and     eax, 0xf000
0x114c4:    mov     dword ptr [rsp + 0x2c], eax
0x114c8:    cmp     ebx, 0x2000
0x114ce:    je      0x11b00
0x114d4:    test    r13, r13
0x114d7:    je      0x11c58
0x114dd:    mov     rdi, qword ptr [rsp + 0x10]
0x114e2:    call    0x1d670
0x114e7:    mov     r14, rax
0x114ea:    test    r13, r13
0x114ed:    jle     0x11aa0
0x114f3:    movabs  rdx, 0xa3d70a3d70a3d70b
0x114fd:    mov     rax, r13
0x11500:    imul    rdx
0x11503:    mov     rax, r13
0x11506:    sar     rax, 0x3f
0x1150a:    add     rdx, r13
0x1150d:    sar     rdx, 6
0x11511:    sub     rdx, rax
0x11514:    mov     qword ptr [rsp + 0x38], rdx
0x11519:    xor     r12d, r12d
0x1151c:    jmp     0x11530
0x11530:
0x1151e:    nop
0x11520:    add     r12, 0x8000
0x11527:    cmp     r12, r13
0x1152a:    jge     0x11aa0
0x11530:    mov     rax, qword ptr [rsp + 0x10]

```

```

0x11535:      mov     rbp, r13
0x11538:      mov     edx, dword ptr [rip + 0x1a572]
0x1153e:      sub     rbp, r12
0x11541:      cmp     rbp, rax
0x11544:      cmovg   rbp, rax
0x11548:      test    edx, edx
0x1154a:      jne     0x11690
0x11550:      mov     edi, 1
0x11555:      call    0x18b70
0x1155a:      mov     edi, r15d
0x1155d:      movsxd  rdx, ebp
0x11560:      mov     rsi, r14
0x11563:      call    0x18d30
0x11568:      xor     edi, edi
0x1156a:      mov     ebx, eax
0x1156c:      call    0x18b70
0x11571:      cmp     ebp, ebx
0x11573:      je      0x11583
0x11575:      test    ebx, ebx
0x11577:      js      0x11bf0
0x1157d:      je      0x11aa0
0x11583:      mov     eax, dword ptr [rip + 0x1a527]
0x11589:      test    eax, eax
0x1158b:      jne     0x11778
0x11591:      movsxd  rbp, ebx
0x11594:      mov     edi, dword ptr [rsp + 0x28]
0x11598:      mov     rsi, r14
0x1159b:      mov     rdx, rbp
0x1159e:      call    0x18da0
0x115a3:      cmp     ebx, eax
0x115a5:      jne     0x11b50
0x115ab:      add     rbp, r12
0x115ae:      cmp     rbp, r13
0x115b1:      jg      0x11520
0x115b7:      mov     rdi, qword ptr [rsp + 8]
0x115bc:      call    0x49b0
0x115c1:      lea     rdi, [rax + 0x20]
0x115c5:      call    0x1d670
0x115ca:      cmp     dword ptr [rsp + 0x2c], 0x8000
0x115d2:      mov     rbx, rax
0x115d5:      je      0x117e0
0x115db:      mov     r9, qword ptr [rsp + 8]
0x115e0:      mov     rdi, rax
0x115e3:      mov     r8, rbp
0x115e6:      mov     esi, 1
0x115eb:      lea     rcx, [rip + 0xfc7b]
0x115f2:      mov     rdx, -1
0x115f9:      xor     eax, eax

```

```

0x115fb:      call    0x4f10
0x11600:      mov     rdi, rbx
0x11603:      mov     edx, 1
0x11608:      mov     esi, 1
0x1160d:      call    0xe480
0x11612:      xor     eax, eax
0x11614:      call    0xa820
0x11619:      mov     rdi, rbx
0x1161c:      call    0x18d10
0x11621:      jmp     0x11520
0x11520
0x11626:      nop     word ptr cs:[rax + rax]
0x11630:      mov     eax, dword ptr [rsp + 0x58]
0x11634:      and     eax, 0xf000
0x11639:      cmp     eax, 0x4000
0x1163e:      jne     0x113c0
0x11644:      mov     r14, qword ptr [rsp + 8]
0x11649:      mov     rdi, r14
0x1164c:      call    0x49b0
0x11651:      mov     rdi, qword ptr [rsp + 0x18]
0x11656:      lea     rsi, [r13 + rax + 2]
0x1165b:      call    0x1d690
0x11660:      mov     rdi, rax
0x11663:      mov     r15, rax
0x11666:      mov     qword ptr [rsp + 0x18], rax
0x1166b:      call    0x49b0
0x11670:      mov     rsi, r14
0x11673:      mov     byte ptr [r15 + rax], 0x2f
0x11678:      lea     rdi, [r15 + rax + 1]
0x1167d:      call    0x48b0
0x11682:      jmp     0x113c0
0x113c0
0x11687:      nop     word ptr [rax + rax]
0x11690:      mov     dword ptr [rip + 0x1a416], 0
0x1169a:      xor     eax, eax
0x1169c:      call    0xc8c0
0x116a1:      mov     edx, 4
0x116a6:      lea     rsi, [rip + 0xee85]
0x116ad:      lea     rdi, [rip + 0xfb45]
0x116b4:      mov     qword ptr [rsp + 0x20], rax
0x116b9:      call    0x8070
0x116be:      mov     rdi, qword ptr [rsp + 0x20]
0x116c3:      mov     ebx, eax
0x116c5:      call    0xc970
0x116ca:      xor     eax, eax
0x116cc:      and     ebx, 0xffffffff
0x116cf:      call    0xd870
0x116d4:      xor     eax, eax

```

```

0x116d6:      call    0xd740
0x116db:      xor     eax, eax
0x116dd:      call    0xa820
0x116e2:      cmp     bl, 0x4e
0x116e5:      je      0x11550
0x116eb:      mov     edi, r15d
0x116ee:      call    0x4ab0
0x116f3:      mov     edi, dword ptr [rsp + 0x28]
0x116f7:      call    0x4ab0
0x116fc:      mov     rdi, qword ptr [rsp + 0x18]
0x11701:      lea     rsi, [rsp + 0xd0]
0x11709:      call    0x18f40
0x1170e:      test    eax, eax
0x11710:      jne     0x11729
0x11712:      mov     eax, dword ptr [rsp + 0xe8]
0x11719:      and     eax, 0xf000
0x1171e:      cmp     eax, 0x8000
0x11723:      je      0x11ced
0x11729:      mov     rdi, r14
0x1172c:      mov     r13d, 2
0x11732:      call    0x18d10
0x11737:      mov     rdi, qword ptr [rsp + 0x18]
0x1173c:      call    0x18d10
0x11741:      nop     dword ptr [rax]
0x11748:      mov     rax, qword ptr [rsp + 0x168]
0x11750:      sub     rax, qword ptr fs:[0x28]
0x11759:      jne     0x11cfc
0x1175f:      add     rsp, 0x178
0x11766:      mov     eax, r13d
0x11769:      pop     rbx
0x1176a:      pop     rbp
0x1176b:      pop     r12
0x1176d:      pop     r13
0x1176f:      pop     r14
0x11771:      pop     r15
0x11773:      ret
0x11774:      nop     dword ptr [rax]
0x11778:      mov     dword ptr [rip + 0x1a32e], 0
0x11782:      xor     eax, eax
0x11784:      call    0xc8c0
0x11789:      mov     edx, 4
0x1178e:      lea     rsi, [rip + 0xed9d]
0x11795:      lea     rdi, [rip + 0xfa5d]
0x1179c:      mov     qword ptr [rsp + 0x20], rax
0x117a1:      call    0x8070
0x117a6:      mov     rdi, qword ptr [rsp + 0x20]
0x117ab:      mov     ebp, eax
0x117ad:      call    0xc970

```

```

0x117b2:    xor     eax, eax
0x117b4:    and     ebp, 0xffffffff
0x117b7:    call    0xd870
0x117bc:    xor     eax, eax
0x117be:    call    0xd740
0x117c3:    xor     eax, eax
0x117c5:    call    0xa820
0x117ca:    cmp     bpl, 0x4e
0x117ce:    je      0x11591
0x117d4:    jmp     0x116eb
0x116eb
0x117d9:    nop     dword ptr [rax]
0x117e0:    cmp     r13, 0x1869f
0x117e7:    jle     0x119f0
0x117ed:    mov     rax, rbp
0x117f0:    cqo
0x117f2:    idiv    qword ptr [rsp + 0x38]
0x117f7:    mov     r8, rax
0x117fa:    mov     r9, qword ptr [rsp + 8]
0x117ff:    mov     esi, 1
0x11804:    mov     rdi, rbx
0x11807:    xor     eax, eax
0x11809:    lea     rcx, [rip + 0xfa4b]
0x11810:    mov     rdx, -1
0x11817:    call    0x4f10
0x1181c:    jmp     0x11600
0x11600
0x11821:    nop     dword ptr [rax]
0x11828:    lea     rsi, [rsp + 0x40]
0x1182d:    mov     rdi, r14
0x11830:    call    0x18f40
0x11835:    test    eax, eax
0x11837:    jne     0x118c3
0x1183d:    mov     eax, dword ptr [rsp + 0x58]
0x11841:    mov     r13d, 0xb
0x11847:    and     eax, 0xf000
0x1184c:    cmp     eax, 0x4000
0x11851:    jne     0x11748
0x11857:    mov     rdi, r14
0x1185a:    call    0x49b0
0x1185f:    mov     r15, qword ptr [rsp + 8]
0x11864:    mov     rbx, rax
0x11867:    mov     rdi, r15
0x1186a:    call    0x49b0
0x1186f:    lea     rdi, [rbx + rax + 2]
0x11874:    call    0x1d670
0x11879:    mov     r9, r15
0x1187c:    mov     r8, r14

```

```

0x1187f:    mov     esi, 1
0x11884:    mov     rdi, rax
0x11887:    mov     r12, rax
0x1188a:    xor     eax, eax
0x1188c:    mov     rdx, -1
0x11893:    lea     rcx, [rip + 0xee67]
0x1189a:    call    0x4f10
0x1189f:    mov     edi, dword ptr [rbp + 0x90]
0x118a5:    test    edi, edi
0x118a7:    je      0x118bb
0x118a9:    xor     esi, esi
0x118ab:    mov     rdi, r12
0x118ae:    call    0x4d90
0x118b3:    test    eax, eax
0x118b5:    je      0x11cbb
0x118bb:    mov     rdi, r12
0x118be:    call    0x18d10
0x118c3:    mov     r15, qword ptr [rsp + 8]
0x118c8:    mov     r13d, 6
0x118ce:    mov     rdi, r15
0x118d1:    call    0x49b0
0x118d6:    mov     rdi, r14
0x118d9:    mov     rbx, rax
0x118dc:    call    0x49b0
0x118e1:    lea     rdi, [rbx + rax + 0x21]
0x118e6:    call    0x1d670
0x118eb:    mov     rdi, r14
0x118ee:    mov     r12, rax
0x118f1:    call    0x190c0
0x118f6:    cmp     byte ptr [rax], 0
0x118f9:    je      0x11748
0x118ff:    mov     rdi, r15
0x11902:    call    0x49b0
0x11907:    mov     rdi, r14
0x1190a:    mov     rbx, rax
0x1190d:    call    0x49b0
0x11912:    lea     rdi, [rbx + rax + 0x20]
0x11917:    call    0x1d670
0x1191c:    mov     r9, r14
0x1191f:    mov     r8, r15
0x11922:    mov     esi, 1
0x11927:    lea     rcx, [rip + 0xf8ed]
0x1192e:    mov     r13, rax
0x11931:    mov     rdi, rax
0x11934:    xor     eax, eax
0x11936:    mov     rdx, -1
0x1193d:    call    0x4f10
0x11942:    mov     edx, 1

```

```

0x11947:      mov     esi, 1
0x1194c:      mov     rdi, r13
0x1194f:      call    0xe480
0x11954:      xor     eax, eax
0x11956:      call    0xa820
0x1195b:      mov     rdi, r13
0x1195e:      mov     r13d, 0xc
0x11964:      call    0x18d10
0x11969:      mov     r9, r14
0x1196c:      mov     r8, r15
0x1196f:      mov     esi, 1
0x11974:      lea     rcx, [rip + 0xf8a7]
0x1197b:      mov     rdx, -1
0x11982:      mov     rdi, r12
0x11985:      xor     eax, eax
0x11987:      call    0x4f10
0x1198c:      mov     esi, 1
0x11991:      mov     rdi, r12
0x11994:      call    0x193f0
0x11999:      mov     rdi, r12
0x1199c:      mov     ebx, eax
0x1199e:      call    0x18d10
0x119a3:      mov     rdi, qword ptr [rbp + 0x18]
0x119a7:      call    0xc4b0
0x119ac:      mov     eax, ebx
0x119ae:      and     eax, 0x7f
0x119b1:      add     eax, 1
0x119b4:      cmp     al, 1
0x119b6:      jg      0x11748
0x119bc:      mov     r13d, 1
0x119c2:      test    bl, 0x7f
0x119c5:      jne     0x11748
0x119cb:      and     bh, 0xff
0x119ce:      je      0x11748
0x119d4:      lea     rdi, [rip + 0xf857]
0x119db:      mov     r13d, 9
0x119e1:      call    0x195e0
0x119e6:      jmp     0x11748
0x11748:
0x119eb:      nop     dword ptr [rax + rax]
0x119f0:      lea     rax, [rbp + rbp*4]
0x119f5:      lea     r8, [rax + rax*4]
0x119f9:      shl     r8, 2
0x119fd:      mov     rax, r8
0x11a00:      cqo
0x11a02:      idiv    r13
0x11a05:      mov     r8, rax
0x11a08:      jmp     0x117fa

```



```

0x117fa
0x11a0d:    nop     dword ptr [rax]
0x11a10:    mov     rdi, qword ptr [rsp + 8]
0x11a15:    lea     rsi, [rsp + 0xd0]
0x11a1d:    call    0x18f40
0x11a22:    cmp     eax, -1
0x11a25:    je      0x11393
0x11a2b:    mov     ebx, dword ptr [rsp + 0xe8]
0x11a32:    mov     r12d, ebx
0x11a35:    and     r12d, 0xf000
0x11a3c:    jmp     0x11386
0x11386
0x11a41:    nop     dword ptr [rax]
0x11a48:    mov     r8, qword ptr [rsp + 8]
0x11a4d:    lea     rcx, [rip + 0xf7e1]
0x11a54:    jmp     0x11423
0x11423
0x11a59:    nop     dword ptr [rax]
0x11a60:    call    0xe4e0
0x11a65:    mov     rsi, r13
0x11a68:    xor     eax, eax
0x11a6a:    mov     rdi, rbp
0x11a6d:    call    0x11220
0x11a72:    mov     r13d, eax
0x11a75:    xor     eax, eax
0x11a77:    call    0xa820
0x11a7c:    test    r13d, r13d
0x11a7f:    je      0x113f4
0x11a85:    jmp     0x11748
0x11748
0x11a8a:    nop     word ptr [rax + rax]
0x11a90:    mov     r13d, 4
0x11a96:    jmp     0x11748
0x11748
0x11a9b:    nop     dword ptr [rax + rax]
0x11aa0:    call    0x49e0
0x11aa5:    test    eax, eax
0x11aa7:    je      0x11c90
0x11aad:    mov     edi, r15d
0x11ab0:    mov     r13d, 1
0x11ab6:    call    0x4ab0
0x11abb:    mov     edi, dword ptr [rsp + 0x28]
0x11abf:    call    0x4ab0
0x11ac4:    mov     rdi, r14
0x11ac7:    call    0x18d10
0x11acc:    mov     rdi, qword ptr [rsp + 0x18]
0x11ad1:    call    0x18d10
0x11ad6:    jmp     0x11748

```

```

0x11748
0x11adb:      nop      dword ptr [rax + rax]
0x11ae0:      call     0x4ab0
0x11ae5:      mov      rdi, qword ptr [rsp + 0x18]
0x11aea:      mov      r13d, 6
0x11af0:      call     0x18d10
0x11af5:      jmp      0x11748
0x11748
0x11afa:      nop      word ptr [rax + rax]
0x11b00:      movabs    r13, 0x7fffffffffffffff
0x11b0a:      mov      edi, 0x8000
0x11b0f:      call     0x1d670
0x11b14:      mov      qword ptr [rsp + 0x10], 0x8000
0x11b1d:      mov      r14, rax
0x11b20:      movabs    rax, 0x147ae147ae147ae
0x11b2a:      mov      qword ptr [rsp + 0x38], rax
0x11b2f:      jmp      0x11519
0x11519
0x11b34:      nop      dword ptr [rax]
0x11b38:      mov      edi, r15d
0x11b3b:      call     0x182f0
0x11b40:      mov      qword ptr [rsp + 0x10], rax
0x11b45:      mov      r13, rax
0x11b48:      jmp      0x114b7
0x114b7
0x11b4d:      nop      dword ptr [rax]
0x11b50:      mov      dword ptr [rsp + 8], eax
0x11b54:      call     0x4860
0x11b59:      mov      edi, r15d
0x11b5c:      mov      ebx, dword ptr [rax]
0x11b5e:      call     0x4ab0
0x11b63:      mov      edi, dword ptr [rsp + 0x28]
0x11b67:      call     0x4ab0
0x11b6c:      mov      rdi, qword ptr [rsp + 0x18]
0x11b71:      lea      rsi, [rsp + 0xd0]
0x11b79:      call     0x18f40
0x11b7e:      mov      edx, dword ptr [rsp + 8]
0x11b82:      test     eax, eax
0x11b84:      jne      0x11bb0
0x11b86:      mov      eax, dword ptr [rsp + 0xe8]
0x11b8d:      and      eax, 0xf000
0x11b92:      cmp      eax, 0x8000
0x11b97:      jne      0x11bb0
0x11b99:      mov      rdi, qword ptr [rsp + 0x18]
0x11b9e:      call     0x4870
0x11ba3:      mov      edx, dword ptr [rsp + 8]
0x11ba7:      nop      word ptr [rax + rax]
0x11bb0:      mov      rdi, r14

```

```

0x11bb3:    mov     dword ptr [rsp + 8], edx
0x11bb7:    call   0x18d10
0x11bbc:    mov     rdi, qword ptr [rsp + 0x18]
0x11bc1:    call   0x18d10
0x11bc6:    mov     edx, dword ptr [rsp + 8]
0x11bca:    cmp     ebx, 0x1c
0x11bcd:    sete    r13b
0x11bd1:    test    edx, edx
0x11bd3:    setg    al
0x11bd6:    or      r13d, eax
0x11bd9:    movzx   r13d, r13b
0x11bdd:    add     r13d, 7
0x11be1:    jmp     0x11748
0x11748:
0x11be6:    nop     word ptr cs:[rax + rax]
0x11bf0:    mov     edi, r15d
0x11bf3:    call   0x4ab0
0x11bf8:    mov     edi, dword ptr [rsp + 0x28]
0x11bfc:    call   0x4ab0
0x11c01:    mov     rdi, qword ptr [rsp + 0x18]
0x11c06:    lea     rsi, [rsp + 0xd0]
0x11c0e:    call   0x18f40
0x11c13:    test    eax, eax
0x11c15:    jne     0x11c38
0x11c17:    mov     eax, dword ptr [rsp + 0xe8]
0x11c1e:    and     eax, 0xf000
0x11c23:    cmp     eax, 0x8000
0x11c28:    jne     0x11c38
0x11c2a:    mov     rdi, qword ptr [rsp + 0x18]
0x11c2f:    call   0x4870
0x11c34:    nop     dword ptr [rax]
0x11c38:    mov     rdi, r14
0x11c3b:    mov     r13d, 5
0x11c41:    call   0x18d10
0x11c46:    mov     rdi, qword ptr [rsp + 0x18]
0x11c4b:    call   0x18d10
0x11c50:    jmp     0x11748
0x11748:
0x11c55:    nop     dword ptr [rax]
0x11c58:    call   0x49e0
0x11c5d:    test    eax, eax
0x11c5f:    je      0x11ca7
0x11c61:    mov     edi, r15d
0x11c64:    mov     r13d, 1
0x11c6a:    call   0x4ab0
0x11c6f:    mov     edi, dword ptr [rsp + 0x28]
0x11c73:    call   0x4ab0
0x11c78:    mov     rdi, qword ptr [rsp + 0x18]

```

```

0x11c7d:      call    0x18d10
0x11c82:      jmp     0x11748
0x11748
0x11c87:      nop     word ptr [rax + rax]
0x11c90:      mov     edx, dword ptr [rsp + 0x34]
0x11c94:      mov     esi, dword ptr [rsp + 0x30]
0x11c98:      mov     rdi, qword ptr [rsp + 0x18]
0x11c9d:      call    0x4cc0
0x11ca2:      jmp     0x11aad
0x11aad
0x11ca7:      mov     edx, dword ptr [rsp + 0x34]
0x11cab:      mov     esi, dword ptr [rsp + 0x30]
0x11caf:      mov     rdi, qword ptr [rsp + 0x18]
0x11cb4:      call    0x4cc0
0x11cb9:      jmp     0x11c61
0x11c61
0x11cbb:      call    0xe4e0
0x11cc0:      mov     rdi, rbp
0x11cc3:      mov     rsi, r12
0x11cc6:      xor     eax, eax
0x11cc8:      call    0x11220
0x11ccd:      mov     r13d, eax
0x11cd0:      xor     eax, eax
0x11cd2:      call    0xa820
0x11cd7:      mov     rdi, r12
0x11cda:      call    0x18d10
0x11cdf:      test    r13d, r13d
0x11ce2:      je      0x118c3
0x11ce8:      jmp     0x11748
0x11748
0x11ced:      mov     rdi, qword ptr [rsp + 0x18]
0x11cf2:      call    0x4870
0x11cf7:      jmp     0x11729
0x11729
0x11cfc:      call    0x49d0
0x11d01:      nop     word ptr cs:[rax + rax]
0x11d0c:      nop     dword ptr [rax]
0x11d10:      endbr64
0x11d14:      push    r15
0x11d16:      and     ecx, 0xf000
0x11d1c:      push    r14
0x11d1e:      mov     r14, rdi
0x11d21:      push    r13
0x11d23:      push    r12
0x11d25:      mov     r12, rdx
0x11d28:      push    rbp
0x11d29:      mov     rbp, rsi
0x11d2c:      push    rbx

```

```

0x11d2d:    sub     rsp, 0x148
0x11d34:    mov     rax, qword ptr fs:[0x28]
0x11d3d:    mov     qword ptr [rsp + 0x138], rax
0x11d45:    xor     eax, eax
0x11d47:    cmp     ecx, 0x4000
0x11d4d:    je      0x11f20
0x11d53:    mov     rdi, rdx
0x11d56:    lea     rbx, [rsp + 0x10]
0x11d5b:    call    0x49b0
0x11d60:    mov     rdi, r12
0x11d63:    mov     r13, rax
0x11d66:    call    0x1d7a0
0x11d6b:    mov     rsi, rbx
0x11d6e:    mov     rdi, rax
0x11d71:    mov     r12, rax
0x11d74:    call    0x18f40
0x11d79:    test    eax, eax
0x11d7b:    je      0x11ed0
0x11d81:    mov     rdi, r12
0x11d84:    call    0x190c0
0x11d89:    cmp     byte ptr [rax], 0
0x11d8c:    mov     r15, rax
0x11d8f:    je      0x12170
0x11d95:    cmp     r12, rax
0x11d98:    je      0x120e8
0x11d9e:    mov     edx, dword ptr [r14 + 0x90]
0x11da5:    test    edx, edx
0x11da7:    je      0x11dbb
0x11da9:    xor     esi, esi
0x11dab:    mov     rdi, r12
0x11dae:    call    0x4d90
0x11db3:    test    eax, eax
0x11db5:    je      0x12140
0x11dbb:    mov     rdi, rbp
0x11dbe:    call    0x49b0
0x11dc3:    lea     rdi, [rax + 0x20]
0x11dc7:    call    0x1d670
0x11dcc:    mov     r8, rbp
0x11dcf:    mov     esi, 1
0x11dd4:    lea     rcx, [rip + 0xf4c2]
0x11ddb:    mov     r13, rax
0x11dde:    mov     rdi, rax
0x11de1:    mov     rdx, -1
0x11de8:    xor     eax, eax
0x11dea:    call    0x4f10
0x11def:    mov     edx, 1
0x11df4:    mov     esi, 1
0x11df9:    mov     rdi, r13

```

```

0x11dfc:    call    0xe480
0x11e01:    xor     eax, eax
0x11e03:    call    0xa820
0x11e08:    mov     rdi, r13
0x11e0b:    mov     r13d, 6
0x11e11:    call    0x18d10
0x11e16:    lea     rsi, [rsp + 0xa0]
0x11e1e:    mov     rdi, rbp
0x11e21:    call    0x18f40
0x11e26:    cmp     eax, -1
0x11e29:    je      0x11ea1
0x11e2b:    mov     rsi, rbx
0x11e2e:    mov     rdi, r12
0x11e31:    call    0x18f40
0x11e36:    cmp     eax, -1
0x11e39:    je      0x121c8
0x11e3f:    mov     rax, qword ptr [rsp + 0xa0]
0x11e47:    cmp     qword ptr [rsp + 0x10], rax
0x11e4c:    jne     0x11e5a
0x11e4e:    mov     eax, dword ptr [r14 + 0x10]
0x11e52:    test    eax, eax
0x11e54:    je      0x12188
0x11e5a:    mov     r9d, dword ptr [rsp + 0xc0]
0x11e62:    mov     rdx, r12
0x11e65:    mov     rsi, rbp
0x11e68:    xor     eax, eax
0x11e6a:    mov     r8d, dword ptr [rsp + 0xbc]
0x11e72:    mov     ecx, dword ptr [rsp + 0xb8]
0x11e79:    mov     rdi, r14
0x11e7c:    call    0x11330
0x11e81:    mov     r13d, eax
0x11e84:    cmp     eax, 3
0x11e87:    je      0x11ea1
0x11e89:    jg      0x120d0
0x11e8f:    cmp     eax, 1
0x11e92:    je      0x121a8
0x11e98:    cmp     eax, 2
0x11e9b:    jne     0x12210
0x11ea1:    mov     rax, qword ptr [rsp + 0x138]
0x11ea9:    sub     rax, qword ptr fs:[0x28]
0x11eb2:    jne     0x122c1
0x11eb8:    add     rsp, 0x148
0x11ebf:    mov     eax, r13d
0x11ec2:    pop     rbx
0x11ec3:    pop     rbp
0x11ec4:    pop     r12
0x11ec6:    pop     r13
0x11ec8:    pop     r14

```

```

0x11eca:    pop     r15
0x11ecc:    ret
0x11ecd:    nop     dword ptr [rax]
0x11ed0:    mov     eax, dword ptr [rsp + 0x28]
0x11ed4:    and     eax, 0xf000
0x11ed9:    cmp     eax, 0x4000
0x11ede:    jne     0x11d81
0x11ee4:    mov     rdi, rbp
0x11ee7:    call    0x49b0
0x11eec:    mov     rdi, r12
0x11eef:    lea     rsi, [r13 + rax + 2]
0x11ef4:    call    0x1d690
0x11ef9:    mov     rdi, rax
0x11efc:    mov     r12, rax
0x11eff:    call    0x49b0
0x11f04:    mov     rsi, rbp
0x11f07:    mov     byte ptr [r12 + rax], 0x2f
0x11f0c:    lea     rdi, [r12 + rax + 1]
0x11f11:    call    0x48b0
0x11f16:    jmp     0x11d81
0x11d81:
0x11f1b:    nop     dword ptr [rax + rax]
0x11f20:    lea     rsi, [rsp + 0x10]
0x11f25:    mov     rdi, rdx
0x11f28:    call    0x18f40
0x11f2d:    test    eax, eax
0x11f2f:    jne     0x11fb7
0x11f35:    mov     eax, dword ptr [rsp + 0x28]
0x11f39:    mov     r13d, 9
0x11f3f:    and     eax, 0xf000
0x11f44:    cmp     eax, 0x4000
0x11f49:    jne     0x11ea1
0x11f4f:    mov     rdi, r12
0x11f52:    call    0x49b0
0x11f57:    mov     rdi, rbp
0x11f5a:    mov     rbx, rax
0x11f5d:    call    0x49b0
0x11f62:    lea     rdi, [rbx + rax + 2]
0x11f67:    call    0x1d670
0x11f6c:    mov     esi, 1
0x11f71:    mov     r9, rbp
0x11f74:    mov     r8, r12
0x11f77:    mov     rdi, rax
0x11f7a:    mov     r15, rax
0x11f7d:    xor     eax, eax
0x11f7f:    mov     rdx, -1
0x11f86:    lea     rcx, [rip + 0xe774]
0x11f8d:    call    0x4f10

```

```

0x11f92:    mov     esi, dword ptr [r14 + 0x90]
0x11f99:    test    esi, esi
0x11f9b:    je      0x11faf
0x11f9d:    xor     esi, esi
0x11f9f:    mov     rdi, r15
0x11fa2:    call    0x4d90
0x11fa7:    test    eax, eax
0x11fa9:    je      0x12278
0x11faf:    mov     rdi, r15
0x11fb2:    call    0x18d10
0x11fb7:    mov     rdi, rbp
0x11fba:    call    0x49b0
0x11fbf:    mov     rdi, r12
0x11fc2:    mov     rbx, rax
0x11fc5:    call    0x49b0
0x11fca:    lea     rdi, [rbx + rax + 0x21]
0x11fcf:    call    0x1d670
0x11fd4:    mov     rdi, r12
0x11fd7:    mov     r13, rax
0x11fda:    call    0x190c0
0x11fdf:    cmp     byte ptr [rax], 0
0x11fe2:    je      0x12178
0x11fe8:    mov     rdi, rbp
0x11feb:    call    0x49b0
0x11ff0:    mov     rdi, r12
0x11ff3:    mov     rbx, rax
0x11ff6:    call    0x49b0
0x11ffb:    lea     rdi, [rbx + rax + 0x20]
0x12000:    call    0x1d670
0x12005:    mov     r9, r12
0x12008:    mov     r8, rbp
0x1200b:    mov     esi, 1
0x12010:    lea     rcx, [rip + 0xf26c]
0x12017:    mov     r15, rax
0x1201a:    mov     rdi, rax
0x1201d:    xor     eax, eax
0x1201f:    mov     rdx, -1
0x12026:    call    0x4f10
0x1202b:    mov     edx, 1
0x12030:    mov     esi, 1
0x12035:    mov     rdi, r15
0x12038:    call    0xe480
0x1203d:    xor     eax, eax
0x1203f:    call    0xa820
0x12044:    mov     rdi, r15
0x12047:    call    0x18d10
0x1204c:    mov     r9, r12
0x1204f:    mov     r8, rbp

```



```

0x12052:      mov     rdi, r13
0x12055:      lea     rcx, [rip + 0xf22e]
0x1205c:      mov     rdx, -1
0x12063:      mov     esi, 1
0x12068:      xor     eax, eax
0x1206a:      call    0x4f10
0x1206f:      mov     esi, 1
0x12074:      mov     rdi, r13
0x12077:      call    0x193f0
0x1207c:      mov     rdi, r13
0x1207f:      mov     r13d, 0xc
0x12085:      mov     ebx, eax
0x12087:      call    0x18d10
0x1208c:      mov     rdi, qword ptr [r14 + 0x18]
0x12090:      call    0xc4b0
0x12095:      mov     eax, ebx
0x12097:      and     eax, 0x7f
0x1209a:      add     eax, 1
0x1209d:      cmp     al, 1
0x1209f:      jg      0x11ea1
0x120a5:      mov     r13d, 8
0x120ab:      test    bl, 0x7f
0x120ae:      jne     0x11ea1
0x120b4:      and     bh, 0xff
0x120b7:      jne     0x122b0
0x120bd:      mov     r13d, 1
0x120c3:      jmp     0x11ea1
0x11ea1
0x120c8:      nop     dword ptr [rax + rax]
0x120d0:      cmp     eax, 8
0x120d3:      jne     0x12210
0x120d9:      mov     r13d, 0xa
0x120df:      jmp     0x11ea1
0x11ea1
0x120e4:      nop     dword ptr [rax]
0x120e8:      mov     rdi, r12
0x120eb:      call    0x49b0
0x120f0:      lea     rdi, [rax + 3]
0x120f4:      mov     r13, rax
0x120f7:      call    0x1d670
0x120fc:      mov     ecx, 0x2f2e
0x12101:      lea     rdx, [r13 + 1]
0x12105:      mov     rsi, r15
0x12108:      mov     word ptr [rax], cx
0x1210b:      lea     rcx, [rax + 2]
0x1210f:      mov     r12, rax
0x12112:      mov     rdi, rcx
0x12115:      mov     qword ptr [rsp + 8], rcx

```

```

0x1211a:      call    0x4bb0
0x1211f:      mov     rdi, r15
0x12122:      call    0x18d10
0x12127:      mov     edx, dword ptr [r14 + 0x90]
0x1212e:      mov     r15, qword ptr [rsp + 8]
0x12133:      test    edx, edx
0x12135:      jne     0x11da9
0x1213b:      jmp     0x11dbb
0x11dbb
0x12140:      call    0xe4e0
0x12145:      xor     eax, eax
0x12147:      mov     rsi, r15
0x1214a:      mov     rdi, r14
0x1214d:      call    0x11220
0x12152:      mov     r13d, eax
0x12155:      xor     eax, eax
0x12157:      call    0xa820
0x1215c:      test    r13d, r13d
0x1215f:      je      0x11dbb
0x12165:      jmp     0x11ea1
0x11ea1
0x1216a:      nop     word ptr [rax + rax]
0x12170:      mov     rdi, r12
0x12173:      call    0x18d10
0x12178:      mov     r13d, 4
0x1217e:      jmp     0x11ea1
0x11ea1
0x12183:      nop     dword ptr [rax + rax]
0x12188:      xor     eax, eax
0x1218a:      call    0x18ab0
0x1218f:      mov     eax, dword ptr [rsp + 0x28]
0x12193:      and     eax, 0xf000
0x12198:      sub     eax, 0x4000
0x1219d:      test    eax, 0xffffb000
0x121a2:      je      0x12250
0x121a8:      mov     rdi, rbp
0x121ab:      mov     r13d, 5
0x121b1:      call    0x4870
0x121b6:      cmp     eax, -1
0x121b9:      jne     0x120bd
0x121bf:      jmp     0x11ea1
0x11ea1
0x121c4:      nop     dword ptr [rax]
0x121c8:      movzx   r13d, byte ptr [r15 - 1]
0x121cd:      mov     byte ptr [r15 - 1], 0
0x121d2:      mov     rsi, rbx
0x121d5:      cmp     byte ptr [r12], 0
0x121da:      je      0x12200

```

```

0x121dc:      mov     rdi, r12
0x121df:      call    0x18f40
0x121e4:      mov     byte ptr [r15 - 1], r13b
0x121e8:      cmp     eax, -1
0x121eb:      jne     0x11e3f
0x121f1:      mov     r13d, 7
0x121f7:      jmp     0x11ea1
0x11ea1
0x121fc:      nop     dword ptr [rax]
0x12200:      lea     rdi, [rip + 0xf0a0]
0x12207:      call    0x18f40
0x1220c:      jmp     0x121e4
0x121e4
0x1220e:      nop
0x12210:      sub     r13d, 1
0x12214:      lea     rax, [rip + 0x18205]
0x1221b:      xor     ecx, ecx
0x1221d:      mov     rsi, rbp
0x12220:      movsxd  r13, r13d
0x12223:      mov     r8d, 0xe
0x12229:      lea     rdi, [rip + 0xf079]
0x12230:      mov     rdx, qword ptr [rax + r13*8]
0x12234:      xor     eax, eax
0x12236:      mov     r13d, 0xb
0x1223c:      call    0xf7e0
0x12241:      jmp     0x11ea1
0x11ea1
0x12246:      nop     word ptr cs:[rax + rax]
0x12250:      mov     rdi, r12
0x12253:      call    0x4870
0x12258:      mov     rsi, r12
0x1225b:      mov     rdi, rbp
0x1225e:      call    0x4e10
0x12263:      cmp     eax, -1
0x12266:      jne     0x121a8
0x1226c:      jmp     0x12178
0x12178
0x12271:      nop     dword ptr [rax]
0x12278:      call    0xe4e0
0x1227d:      mov     rdi, r14
0x12280:      mov     rsi, r15
0x12283:      xor     eax, eax
0x12285:      call    0x11220
0x1228a:      mov     r13d, eax
0x1228d:      xor     eax, eax
0x1228f:      call    0xa820
0x12294:      mov     rdi, r15
0x12297:      call    0x18d10

```

```

0x1229c:      test    r13d, r13d
0x1229f:      je      0x11fb7
0x122a5:      jmp     0x11ea1
0x11ea1:
0x122aa:      nop     word ptr [rax + rax]
0x122b0:      lea     rdi, [rip + 0xefe3]
0x122b7:      call    0x195e0
0x122bc:      jmp     0x11ea1
0x11ea1:
0x122c1:      call    0x49d0
0x122c6:      nop     word ptr cs:[rax + rax]
0x122d0:      endbr64
0x122d4:      push    r15
0x122d6:      push    r14
0x122d8:      push    r13
0x122da:      mov     r13, rsi
0x122dd:      push    r12
0x122df:      mov     r12, rdi
0x122e2:      mov     rdi, rsi
0x122e5:      push    rbp
0x122e6:      push    rbx
0x122e7:      sub     rsp, 0x18
0x122eb:      call    0x49b0
0x122f0:      lea     rdi, [rax + 1]
0x122f4:      mov     rbx, rax
0x122f7:      call    0x1d670
0x122fc:      movsxd  r14, dword ptr [r12 + 0x30]
0x12301:      mov     rsi, r13
0x12304:      mov     rbp, rax
0x12307:      mov     rdi, rax
0x1230a:      cmp     r14, rbx
0x1230d:      cmova   r14, rbx
0x12311:      mov     rdx, r14
0x12314:      call    0x4880
0x12319:      mov     byte ptr [rbp + r14], 0
0x1231f:      mov     ebx, dword ptr [r12 + 0x68]
0x12324:      test    ebx, ebx
0x12326:      jle     0x12450
0x1232c:      mov     r15, qword ptr [r12 + 0x38]
0x12331:      xor     r12d, r12d
0x12334:      mov     qword ptr [rsp + 8], r15
0x12339:      mov     r13, r15
0x1233c:      jmp     0x1234d
0x1234d:
0x1233e:      nop
0x12340:      add     r12d, 1
0x12344:      add     r15, 0x50
0x12348:      cmp     ebx, r12d

```

```

0x1234b:    je      0x12380
0x1234d:    mov     rsi, qword ptr [r15]
0x12350:    mov     rdi, rbp
0x12353:    call    0x4b30
0x12358:    test    eax, eax
0x1235a:    jne     0x12340
0x1235c:    mov     rdi, rbp
0x1235f:    call    0x18d10
0x12364:    add     rsp, 0x18
0x12368:    mov     eax, r12d
0x1236b:    pop     rbx
0x1236c:    pop     rbp
0x1236d:    pop     r12
0x1236f:    pop     r13
0x12371:    pop     r14
0x12373:    pop     r15
0x12375:    ret
0x12376:    nop     word ptr cs:[rax + rax]
0x12380:    xor     r12d, r12d
0x12383:    jmp     0x12395
0x12395:
0x12385:    nop     dword ptr [rax]
0x12388:    add     r12d, 1
0x1238c:    add     r13, 0x50
0x12390:    cmp     ebx, r12d
0x12393:    je      0x123b0
0x12395:    mov     rsi, qword ptr [r13]
0x12399:    mov     rdi, rbp
0x1239c:    call    0x4830
0x123a1:    test    eax, eax
0x123a3:    jne     0x12388
0x123a5:    cmp     ebx, r12d
0x123a8:    jne     0x1235c
0x123aa:    nop     word ptr [rax + rax]
0x123b0:    mov     r15, qword ptr [rsp + 8]
0x123b5:    xor     r12d, r12d
0x123b8:    jmp     0x123cd
0x123cd:
0x123ba:    nop     word ptr [rax + rax]
0x123c0:    add     r12d, 1
0x123c4:    add     r15, 0x50
0x123c8:    cmp     r12d, ebx
0x123cb:    je      0x12400
0x123cd:    mov     r13, qword ptr [r15]
0x123d0:    mov     rdi, r13
0x123d3:    call    0x49b0
0x123d8:    mov     rsi, r13
0x123db:    mov     rdi, rbp

```

```

0x123de:      cmp     rax, r14
0x123e1:      cmova   rax, r14
0x123e5:      mov     rdx, rax
0x123e8:      call    0x4890
0x123ed:      test    eax, eax
0x123ef:      jne     0x123c0
0x123f1:      cmp     r12d, ebx
0x123f4:      jne     0x1235c
0x123fa:      nop     word ptr [rax + rax]
0x12400:      xor     r12d, r12d
0x12403:      jmp     0x1241a
0x1241a:
0x12405:      nop     dword ptr [rax]
0x12408:      add     r15, 0x50
0x1240c:      add     r12d, 1
0x12410:      mov     qword ptr [rsp + 8], r15
0x12415:      cmp     r12d, ebx
0x12418:      je      0x12450
0x1241a:      mov     r15, qword ptr [rsp + 8]
0x1241f:      mov     r13, qword ptr [r15]
0x12422:      mov     rdi, r13
0x12425:      call    0x49b0
0x1242a:      mov     rsi, r13
0x1242d:      mov     rdi, rbp
0x12430:      cmp     rax, r14
0x12433:      cmova   rax, r14
0x12437:      mov     rdx, rax
0x1243a:      call    0x4c30
0x1243f:      test    eax, eax
0x12441:      jne     0x12408
0x12443:      cmp     r12d, ebx
0x12446:      jne     0x1235c
0x1244c:      nop     dword ptr [rax]
0x12450:      xor     r12d, r12d
0x12453:      jmp     0x1235c
0x1235c:
0x12458:      nop     dword ptr [rax + rax]
0x12460:      endbr64
0x12464:      sub     rsp, 0xa8
0x1246b:      mov     rax, qword ptr fs:[0x28]
0x12474:      mov     qword ptr [rsp + 0x98], rax
0x1247c:      xor     eax, eax
0x1247e:      mov     rsi, rsp
0x12481:      call    0x4b90
0x12486:      mov     r8d, eax
0x12489:      xor     eax, eax
0x1248b:      test    r8d, r8d
0x1248e:      js      0x124a4

```

```

0x12490:    mov     eax, dword ptr [rsp + 0x18]
0x12494:    and     eax, 0xf000
0x12499:    cmp     eax, 0x4000
0x1249e:    sete    al
0x124a1:    movzx   eax, al
0x124a4:    mov     rdx, qword ptr [rsp + 0x98]
0x124ac:    sub     rdx, qword ptr fs:[0x28]
0x124b5:    jne     0x124bf
0x124b7:    add     rsp, 0xa8
0x124be:    ret
0x124bf:    call    0x49d0
0x124c4:    nop     word ptr cs:[rax + rax]
0x124cf:    nop
0x124d0:    endbr64
0x124d4:    push    r13
0x124d6:    push    r12
0x124d8:    mov     r12d, esi
0x124db:    push    rbp
0x124dc:    push    rbx
0x124dd:    sub     rsp, 0xa8
0x124e4:    mov     rax, qword ptr fs:[0x28]
0x124ed:    mov     qword ptr [rsp + 0x98], rax
0x124f5:    xor     eax, eax
0x124f7:    mov     r13, rsp
0x124fa:    call    0x1d7a0
0x124ff:    cmp     byte ptr [rax], 0
0x12502:    mov     rbp, rax
0x12505:    mov     rbx, rax
0x12508:    je      0x12560
0x1250a:    nop     word ptr [rax + rax]
0x12510:    movzx   eax, byte ptr [rbx + 1]
0x12514:    add     rbx, 1
0x12518:    test    al, al
0x1251a:    je      0x12560
0x1251c:    cmp     rbp, rbx
0x1251f:    je      0x12510
0x12521:    cmp     al, 0x2f
0x12523:    jne     0x12510
0x12525:    mov     byte ptr [rbx], 0
0x12528:    mov     rsi, r13
0x1252b:    mov     rdi, rbp
0x1252e:    call    0x4b90
0x12533:    test    eax, eax
0x12535:    js      0x12590
0x12537:    mov     eax, dword ptr [rsp + 0x18]
0x1253b:    and     eax, 0xf000
0x12540:    cmp     eax, 0x4000
0x12545:    jne     0x12590

```

```

0x12547:      mov     byte ptr [rbx], 0x2f
0x1254a:      movzx   eax, byte ptr [rbx + 1]
0x1254e:      add     rbx, 1
0x12552:      test    al, al
0x12554:      jne      0x1251c
0x12556:      nop      word ptr cs:[rax + rax]
0x12560:      mov     rdi, rbp
0x12563:      call    0x18d10
0x12568:      xor     eax, eax
0x1256a:      mov     rdx, qword ptr [rsp + 0x98]
0x12572:      sub     rdx, qword ptr fs:[0x28]
0x1257b:      jne      0x125ae
0x1257d:      add     rsp, 0xa8
0x12584:      pop     rbx
0x12585:      pop     rbp
0x12586:      pop     r12
0x12588:      pop     r13
0x1258a:      ret
0x1258b:      nop      dword ptr [rax + rax]
0x12590:      mov     esi, r12d
0x12593:      mov     rdi, rbp
0x12596:      call    0x48c0
0x1259b:      test    eax, eax
0x1259d:      je       0x12547
0x1259f:      mov     rdi, rbp
0x125a2:      call    0x18d10
0x125a7:      mov     eax, 0xffffffff
0x125ac:      jmp     0x1256a
0x1256a:
0x125ae:      call    0x49d0
0x125b3:      nop      word ptr cs:[rax + rax]
0x125be:      nop
0x125c0:      endbr64
0x125c4:      push    r15
0x125c6:      movsxd  rsi, esi
0x125c9:      mov     r10, rdx
0x125cc:      movsxd  r8, r8d
0x125cf:      push    r14
0x125d1:      lea     rax, [rsi + rsi*4]
0x125d5:      mov     rsi, rdi
0x125d8:      push    r13
0x125da:      shl     rax, 4
0x125de:      push    r12
0x125e0:      push    rbp
0x125e1:      push    rbx
0x125e2:      sub     rsp, 0x68
0x125e6:      mov     qword ptr [rsp + 0x48], rdi
0x125eb:      mov     rdi, rax

```



```

0x125ee:    mov     qword ptr [rsp + 0x50], rax
0x125f3:    add     rax, qword ptr [rsi + 0x38]
0x125f7:    mov     qword ptr [rsp + 0x20], rdx
0x125fc:    mov     rdx, rax
0x125ff:    mov     rax, qword ptr [rax]
0x12602:    mov     qword ptr [rsp + 0x38], rcx
0x12607:    mov     rcx, qword ptr [rcx + 0x38]
0x1260b:    mov     qword ptr [rsp + 0x10], rax
0x12610:    lea     rax, [r8 + r8*4]
0x12614:    shl     rax, 4
0x12618:    mov     qword ptr [rsp + 0x28], r9
0x1261d:    lea     rsi, [rcx + rax]
0x12621:    mov     qword ptr [rsp + 0x58], rax
0x12626:    mov     eax, dword ptr [rdx + 0x10]
0x12629:    mov     ecx, dword ptr [rcx + rdi + 0x10]
0x1262d:    mov     rbx, qword ptr [rsi]
0x12630:    mov     qword ptr [r9], 0
0x12637:    mov     r8d, eax
0x1263a:    mov     qword ptr [r10], 0
0x12641:    mov     edi, ecx
0x12643:    and     r8d, 0xf000
0x1264a:    and     edi, 0xf000
0x12650:    cmp     r8d, 0x4000
0x12657:    je      0x12669
0x12659:    and     eax, 0xd000
0x1265e:    cmp     eax, 0x8000
0x12663:    jne     0x12928
0x12669:    mov     rax, qword ptr [rdx + 8]
0x1266d:    mov     qword ptr [rsp + 8], rax
0x12672:    cmp     edi, 0x4000
0x12678:    je      0x12ac0
0x1267e:    and     ecx, 0xd000
0x12684:    mov     dword ptr [rsp + 0x40], 0
0x1268c:    mov     dword ptr [rsp + 0x44], 1
0x12694:    cmp     ecx, 0x8000
0x1269a:    je      0x12ac0
0x126a0:    cmp     qword ptr [rsp + 8], 0
0x126a6:    je      0x12c90
0x126ac:    mov     r14, qword ptr [rsp + 0x38]
0x126b1:    mov     rdi, qword ptr [r14 + 0x18]
0x126b5:    call    0x49b0
0x126ba:    mov     rdi, rbx
0x126bd:    mov     rbp, rax
0x126c0:    call    0x49b0
0x126c5:    lea     rdi, [rbp + rax + 2]
0x126ca:    call    0x1d670
0x126cf:    mov     r8, qword ptr [r14 + 0x18]
0x126d3:    mov     r9, rbx

```

```

0x126d6:    lea     rcx, [rip + 0xe024]
0x126dd:    mov     rdx, -1
0x126e4:    mov     esi, 1
0x126e9:    mov     rdi, rax
0x126ec:    mov     r12, rax
0x126ef:    xor     eax, eax
0x126f1:    call    0x4f10
0x126f6:    mov     rbx, qword ptr [rsp + 0x10]
0x126fb:    mov     rdi, rbx
0x126fe:    call    0x49b0
0x12703:    lea     rdi, [rax + 0x21]
0x12707:    call    0x1d670
0x1270c:    mov     r8, rbx
0x1270f:    mov     esi, 1
0x12714:    lea     rcx, [rip + 0xebbd]
0x1271b:    mov     rbp, rax
0x1271e:    mov     rdi, rax
0x12721:    mov     rdx, -1
0x12728:    xor     eax, eax
0x1272a:    call    0x4f10
0x1272f:    mov     edx, 1
0x12734:    mov     esi, 1
0x12739:    mov     rdi, rbp
0x1273c:    call    0xe480
0x12741:    xor     eax, eax
0x12743:    call    0xa820
0x12748:    mov     rdi, rbp
0x1274b:    call    0x18d10
0x12750:    mov     rdi, qword ptr [rsp + 0x10]
0x12755:    xor     esi, esi
0x12757:    xor     eax, eax
0x12759:    call    0x4d30
0x1275e:    mov     r15d, eax
0x12761:    cmp     eax, -1
0x12764:    je      0x12c6b
0x1276a:    mov     rdi, r12
0x1276d:    xor     esi, esi
0x1276f:    xor     eax, eax
0x12771:    call    0x4d30
0x12776:    mov     rdi, r12
0x12779:    mov     dword ptr [rsp + 0x1c], eax
0x1277d:    cmp     eax, -1
0x12780:    je      0x12c77
0x12786:    call    0x18d10
0x1278b:    mov     edi, 0x8000
0x12790:    xor     r12d, r12d
0x12793:    call    0x1d670
0x12798:    mov     edi, 0x8000

```

```

0x1279d:    mov     rbp, rax
0x127a0:    call   0x1d670
0x127a5:    mov     rdi, qword ptr [rsp + 0x10]
0x127aa:    mov     r13, rax
0x127ad:    call   0x49b0
0x127b2:    lea     rdi, [rax + 0x21]
0x127b6:    call   0x1d670
0x127bb:    mov     dword ptr [rsp + 0x18], r15d
0x127c0:    movabs  rdx, 0xa3d70a3d70a3d70b
0x127ca:    mov     r14, rax
0x127cd:    mov     rax, qword ptr [rsp + 8]
0x127d2:    imul    rdx
0x127d5:    mov     rax, qword ptr [rsp + 8]
0x127da:    add     rdx, rax
0x127dd:    sar     rax, 0x3f
0x127e1:    sar     rdx, 6
0x127e5:    sub     rdx, rax
0x127e8:    mov     qword ptr [rsp + 0x30], rdx
0x127ed:    jmp     0x1282b
0x1282b:
0x127ef:    nop
0x127f0:    mov     r8, qword ptr [rsp + 0x10]
0x127f5:    mov     esi, 1
0x127fa:    mov     rdi, r14
0x127fd:    xor     eax, eax
0x127ff:    lea     rcx, [rip + 0xeab8]
0x12806:    mov     rdx, -1
0x1280d:    call   0x4f10
0x12812:    mov     edx, 1
0x12817:    mov     esi, 1
0x1281c:    mov     rdi, r14
0x1281f:    call   0xe480
0x12824:    xor     eax, eax
0x12826:    call   0xa820
0x1282b:    mov     ecx, dword ptr [rip + 0x1927f]
0x12831:    test    ecx, ecx
0x12833:    jne     0x12a30
0x12839:    mov     edi, 1
0x1283e:    call   0x18b70
0x12843:    mov     edi, dword ptr [rsp + 0x18]
0x12847:    mov     edx, 0x8000
0x1284c:    mov     rsi, rbp
0x1284f:    call   0x47a0
0x12854:    xor     edi, edi
0x12856:    mov     rbx, rax
0x12859:    call   0x18b70
0x1285e:    test    ebx, ebx
0x12860:    js      0x12af0

```

```

0x12866:      mov     rcx, qword ptr [rsp + 0x20]
0x1286b:      movsxd  rax, ebx
0x1286e:      mov     edi, 1
0x12873:      add     qword ptr [rcx], rax
0x12876:      call    0x18b70
0x1287b:      mov     edi, dword ptr [rsp + 0x1c]
0x1287f:      mov     edx, 0x8000
0x12884:      mov     rsi, r13
0x12887:      call    0x47a0
0x1288c:      xor     edi, edi
0x1288e:      mov     r15, rax
0x12891:      call    0x18b70
0x12896:      test    r15d, r15d
0x12899:      js      0x12b20
0x1289f:      mov     rax, qword ptr [rsp + 0x28]
0x128a4:      movsxd  rdx, r15d
0x128a7:      add     qword ptr [rax], rdx
0x128aa:      cmp     r15d, ebx
0x128ad:      cmovle   ebx, r15d
0x128b1:      test    ebx, ebx
0x128b3:      je      0x12b40
0x128b9:      movsxd  r15, ebx
0x128bc:      mov     rsi, r13
0x128bf:      mov     rdi, rbp
0x128c2:      mov     rdx, r15
0x128c5:      call    0x4af0
0x128ca:      test    eax, eax
0x128cc:      jne     0x12b70
0x128d2:      add     r12, r15
0x128d5:      cmp     qword ptr [rsp + 8], 0
0x128db:      je      0x127f0
0x128e1:      cmp     qword ptr [rsp + 8], 0x1869f
0x128ea:      jle     0x12a10
0x128f0:      mov     rax, r12
0x128f3:      cqo
0x128f5:      idiv    qword ptr [rsp + 0x30]
0x128fa:      mov     r8, rax
0x128fd:      mov     r9, qword ptr [rsp + 0x10]
0x12902:      mov     esi, 1
0x12907:      mov     rdi, r14
0x1290a:      xor     eax, eax
0x1290c:      lea     rcx, [rip + 0xe9b4]
0x12913:      mov     rdx, -1
0x1291a:      call    0x4f10
0x1291f:      jmp     0x12812
0x12812:
0x12924:      nop     dword ptr [rax]
0x12928:      cmp     edi, 0x4000

```

```

0x1292e:      je      0x129f0
0x12934:      and     ecx, 0xd000
0x1293a:      cmp     ecx, 0x8000
0x12940:      je      0x129f0
0x12946:      mov     r14, qword ptr [rsp + 0x38]
0x1294b:      mov     rdi, qword ptr [r14 + 0x18]
0x1294f:      call    0x49b0
0x12954:      mov     rdi, rbx
0x12957:      mov     rbp, rax
0x1295a:      call    0x49b0
0x1295f:      lea     rdi, [rbp + rax + 2]
0x12964:      call    0x1d670
0x12969:      mov     r8, qword ptr [r14 + 0x18]
0x1296d:      mov     r9, rbx
0x12970:      lea     rcx, [rip + 0xdd8a]
0x12977:      mov     rdx, -1
0x1297e:      mov     esi, 1
0x12983:      mov     rdi, rax
0x12986:      mov     r12, rax
0x12989:      xor     eax, eax
0x1298b:      call    0x4f10
0x12990:      mov     rbx, qword ptr [rsp + 0x10]
0x12995:      mov     rdi, rbx
0x12998:      call    0x49b0
0x1299d:      lea     rdi, [rax + 0x21]
0x129a1:      call    0x1d670
0x129a6:      mov     r8, rbx
0x129a9:      mov     esi, 1
0x129ae:      lea     rcx, [rip + 0xe909]
0x129b5:      mov     rdi, rax
0x129b8:      mov     rbp, rax
0x129bb:      mov     rdx, -1
0x129c2:      xor     eax, eax
0x129c4:      call    0x4f10
0x129c9:      mov     dword ptr [rsp + 0x44], 1
0x129d1:      mov     qword ptr [rsp + 8], 0
0x129da:      mov     dword ptr [rsp + 0x40], 1
0x129e2:      jmp     0x1272f
0x1272f
0x129e7:      nop     word ptr [rax + rax]
0x129f0:      mov     rax, qword ptr [rsi + 8]
0x129f4:      mov     dword ptr [rsp + 0x40], 1
0x129fc:      mov     dword ptr [rsp + 0x44], 0
0x12a04:      mov     qword ptr [rsp + 8], rax
0x12a09:      jmp     0x126a0
0x126a0
0x12a0e:      nop
0x12a10:      lea     rax, [r12 + r12*4]

```

```

0x12a14:    lea     rax, [rax + rax*4]
0x12a18:    shl     rax, 2
0x12a1c:    cqo
0x12a1e:    idiv    qword ptr [rsp + 8]
0x12a23:    mov     r8, rax
0x12a26:    jmp     0x128fd
0x128fd
0x12a2b:    nop     dword ptr [rax + rax]
0x12a30:    mov     dword ptr [rip + 0x19076], 0
0x12a3a:    xor     eax, eax
0x12a3c:    call    0xc8c0
0x12a41:    mov     edx, 4
0x12a46:    lea     rsi, [rip + 0xdae5]
0x12a4d:    lea     rdi, [rip + 0xe7a5]
0x12a54:    mov     r15, rax
0x12a57:    call    0x8070
0x12a5c:    mov     rdi, r15
0x12a5f:    mov     ebx, eax
0x12a61:    call    0xc970
0x12a66:    xor     eax, eax
0x12a68:    and     ebx, 0xffffffffdf
0x12a6b:    call    0xd870
0x12a70:    xor     eax, eax
0x12a72:    call    0xd740
0x12a77:    xor     eax, eax
0x12a79:    call    0xa820
0x12a7e:    cmp     bl, 0x4e
0x12a81:    je      0x12839
0x12a87:    mov     rdi, r14
0x12a8a:    mov     r15d, dword ptr [rsp + 0x18]
0x12a8f:    mov     r12, -1
0x12a96:    call    0x18d10
0x12a9b:    mov     rdi, rbp
0x12a9e:    call    0x18d10
0x12aa3:    mov     rdi, r13
0x12aa6:    call    0x18d10
0x12aab:    mov     edi, r15d
0x12aae:    call    0x4ab0
0x12ab3:    mov     edi, dword ptr [rsp + 0x1c]
0x12ab7:    call    0x4ab0
0x12abc:    jmp     0x12b09
0x12b09
0x12abe:    nop
0x12ac0:    mov     rcx, qword ptr [rsp + 8]
0x12ac5:    mov     rax, qword ptr [rsi + 8]
0x12ac9:    mov     dword ptr [rsp + 0x40], 0
0x12ad1:    mov     dword ptr [rsp + 0x44], 0
0x12ad9:    cmp     rcx, rax

```

```

0x12adc:      cmovge  rax, rcx
0x12ae0:      mov     qword ptr [rsp + 8], rax
0x12ae5:      jmp     0x126a0
0x126a0
0x12aea:      nop     word ptr [rax + rax]
0x12af0:      mov     edi, dword ptr [rsp + 0x18]
0x12af4:      mov     r12, -2
0x12afb:      call    0x4ab0
0x12b00:      mov     edi, dword ptr [rsp + 0x1c]
0x12b04:      call    0x4ab0
0x12b09:      add     rsp, 0x68
0x12b0d:      mov     rax, r12
0x12b10:      pop     rbx
0x12b11:      pop     rbp
0x12b12:      pop     r12
0x12b14:      pop     r13
0x12b16:      pop     r14
0x12b18:      pop     r15
0x12b1a:      ret
0x12b1b:      nop     dword ptr [rax + rax]
0x12b20:      mov     edi, dword ptr [rsp + 0x18]
0x12b24:      mov     r12, -3
0x12b2b:      call    0x4ab0
0x12b30:      mov     edi, dword ptr [rsp + 0x1c]
0x12b34:      call    0x4ab0
0x12b39:      jmp     0x12b09
0x12b09
0x12b3b:      nop     dword ptr [rax + rax]
0x12b40:      mov     r15d, dword ptr [rsp + 0x18]
0x12b45:      mov     rdi, r14
0x12b48:      call    0x18d10
0x12b4d:      mov     rdi, rbp
0x12b50:      call    0x18d10
0x12b55:      mov     rdi, r13
0x12b58:      call    0x18d10
0x12b5d:      mov     edi, r15d
0x12b60:      call    0x4ab0
0x12b65:      mov     edi, dword ptr [rsp + 0x1c]
0x12b69:      call    0x4ab0
0x12b6e:      jmp     0x12b09
0x12b09
0x12b70:      mov     r15d, dword ptr [rsp + 0x18]
0x12b75:      xor     eax, eax
0x12b77:      nop     word ptr [rax + rax]
0x12b80:      movzx   ecx, byte ptr [r13 + rax]
0x12b86:      mov     edx, eax
0x12b88:      cmp     byte ptr [rbp + rax], cl
0x12b8c:      jne     0x12b99

```

```

0x12b8e:    lea     edx, [rax + 1]
0x12b91:    add     rax, 1
0x12b95:    cmp     ebx, eax
0x12b97:    jg      0x12b80
0x12b99:    movsxd  rdx, edx
0x12b9c:    add     r12, rdx
0x12b9f:    mov     edx, dword ptr [rsp + 0x40]
0x12ba3:    test    edx, edx
0x12ba5:    jne     0x12bea
0x12ba7:    mov     rax, qword ptr [rsp + 0x48]
0x12bac:    mov     rcx, qword ptr [rsp + 0x50]
0x12bb1:    mov     rax, qword ptr [rax + 0x38]
0x12bb5:    mov     rax, qword ptr [rax + rcx + 8]
0x12bba:    mov     rcx, qword ptr [rsp + 0x20]
0x12bbf:    mov     qword ptr [rcx], rax
0x12bc2:    mov     eax, dword ptr [rsp + 0x44]
0x12bc6:    test    eax, eax
0x12bc8:    jne     0x12c24
0x12bca:    mov     rax, qword ptr [rsp + 0x38]
0x12bcf:    mov     rcx, qword ptr [rsp + 0x58]
0x12bd4:    mov     rax, qword ptr [rax + 0x38]
0x12bd8:    mov     rax, qword ptr [rax + rcx + 8]
0x12bdd:    mov     rcx, qword ptr [rsp + 0x28]
0x12be2:    mov     qword ptr [rcx], rax
0x12be5:    jmp     0x12b45
0x12b45:
0x12bea:    mov     edi, 1
0x12bef:    call    0x18b70
0x12bf4:    mov     rbx, qword ptr [rsp + 0x20]
0x12bf9:    jmp     0x12c07
0x12c07:
0x12bfb:    nop     dword ptr [rax + rax]
0x12c00:    cdqe
0x12c02:    add     qword ptr [rbx], rax
0x12c05:    je      0x12c1b
0x12c07:    mov     edx, 0x8000
0x12c0c:    mov     rsi, rbp
0x12c0f:    mov     edi, r15d
0x12c12:    call    0x47a0
0x12c17:    test    eax, eax
0x12c19:    jg      0x12c00
0x12c1b:    xor     edi, edi
0x12c1d:    call    0x18b70
0x12c22:    jmp     0x12bc2
0x12bc2:
0x12c24:    mov     edi, 1
0x12c29:    call    0x18b70
0x12c2e:    mov     qword ptr [rsp + 8], rbp

```



```

0x12c33:      mov     rbx, qword ptr [rsp + 0x28]
0x12c38:      mov     ebp, dword ptr [rsp + 0x1c]
0x12c3c:      jmp     0x12c47
0x12c47:
0x12c3e:      nop
0x12c40:      cdqe
0x12c42:      add     qword ptr [rbx], rax
0x12c45:      je      0x12c5a
0x12c47:      mov     edx, 0x8000
0x12c4c:      mov     rsi, r13
0x12c4f:      mov     edi, ebp
0x12c51:      call    0x47a0
0x12c56:      test    eax, eax
0x12c58:      jg      0x12c40
0x12c5a:      xor     edi, edi
0x12c5c:      mov     rbp, qword ptr [rsp + 8]
0x12c61:      call    0x18b70
0x12c66:      jmp     0x12b45
0x12b45:
0x12c6b:      mov     r12, -4
0x12c72:      jmp     0x12b09
0x12b09:
0x12c77:      call    0x18d10
0x12c7c:      mov     edi, r15d
0x12c7f:      mov     r12, -5
0x12c86:      call    0x4ab0
0x12c8b:      jmp     0x12b09
0x12b09:
0x12c90:      xor     r12d, r12d
0x12c93:      jmp     0x12b09
0x12b09:
0x12c98:      nop     dword ptr [rax + rax]
0x12ca0:      endbr64
0x12ca4:      push    r15
0x12ca6:      push    r14
0x12ca8:      push    r13
0x12caa:      push    r12
0x12cac:      push    rbp
0x12cad:      mov     rbp, rsi
0x12cb0:      push    rbx
0x12cb1:      mov     rbx, rdi
0x12cb4:      sub     rsp, 0x38
0x12cb8:      mov     r13d, dword ptr [rdi + 0x28]
0x12cbc:      mov     rdi, qword ptr [rip + 0x1780d]
0x12cc3:      mov     rax, qword ptr fs:[0x28]
0x12ccc:      mov     qword ptr [rsp + 0x28], rax
0x12cd1:      xor     eax, eax
0x12cd3:      mov     r14d, dword ptr [rsi + 0x28]

```

```

0x12cd7:    call    0xdb60
0x12cdc:    xor     eax, eax
0x12cde:    call    0xa820
0x12ce3:    mov     r15, qword ptr [rbx + 0x18]
0x12ce7:    mov     rsi, qword ptr [rbp + 0x18]
0x12ceb:    mov     rdi, r15
0x12cee:    call    0x4b30
0x12cf3:    mov     rdx, qword ptr [rbx + 0x38]
0x12cf7:    mov     r8d, eax
0x12cfa:    movsxd  rax, r13d
0x12cfd:    lea     r12, [rax + rax*4]
0x12d01:    shl     r12, 4
0x12d05:    add     rdx, r12
0x12d08:    test    r8d, r8d
0x12d0b:    jne     0x12d39
0x12d0d:    movsxd  rax, r14d
0x12d10:    mov     rdi, qword ptr [rdx]
0x12d13:    mov     qword ptr [rsp + 8], rdx
0x12d18:    lea     rax, [rax + rax*4]
0x12d1c:    shl     rax, 4
0x12d20:    add     rax, qword ptr [rbp + 0x38]
0x12d24:    mov     rsi, qword ptr [rax]
0x12d27:    call    0x4b30
0x12d2c:    mov     rdx, qword ptr [rsp + 8]
0x12d31:    test    eax, eax
0x12d33:    je      0x12e40
0x12d39:    cmp     byte ptr [rdx + 0x48], 1
0x12d3d:    jne     0x12d57
0x12d3f:    movsxd  rax, r14d
0x12d42:    lea     r10, [rax + rax*4]
0x12d46:    mov     rax, qword ptr [rbp + 0x38]
0x12d4a:    shl     r10, 4
0x12d4e:    add     rax, r10
0x12d51:    cmp     byte ptr [rax + 0x48], 1
0x12d55:    je      0x12da0
0x12d57:    mov     ecx, 0x1e
0x12d5c:    xor     edx, edx
0x12d5e:    mov     rsi, r15
0x12d61:    xor     eax, eax
0x12d63:    lea     rdi, [rip + 0xed6]
0x12d6a:    call    0xf760
0x12d6f:    xor     eax, eax
0x12d71:    call    0xe4e0
0x12d76:    mov     rax, qword ptr [rsp + 0x28]
0x12d7b:    sub     rax, qword ptr fs:[0x28]
0x12d84:    jne     0x12fbc
0x12d8a:    add     rsp, 0x38
0x12d8e:    xor     eax, eax

```

```

0x12d90:    pop     rbx
0x12d91:    pop     rbp
0x12d92:    pop     r12
0x12d94:    pop     r13
0x12d96:    pop     r14
0x12d98:    pop     r15
0x12d9a:    jmp     0xa820
0xa820
0x12d9f:    nop
0x12da0:    mov     rcx, qword ptr [rax + 8]
0x12da4:    cmp     qword ptr [rdx + 8], rcx
0x12da8:    je      0x12df2
0x12daa:    mov     edx, dword ptr [rdx + 0x10]
0x12dad:    mov     ecx, edx
0x12daf:    and     ecx, 0xf000
0x12db5:    cmp     ecx, 0x4000
0x12dbb:    je      0x12dcb
0x12dbd:    and     edx, 0xd000
0x12dc3:    cmp     edx, 0x8000
0x12dc9:    jne     0x12df2
0x12dcb:    mov     eax, dword ptr [rax + 0x10]
0x12dce:    mov     edx, eax
0x12dd0:    and     edx, 0xf000
0x12dd6:    cmp     edx, 0x4000
0x12ddc:    je      0x12ee8
0x12de2:    and     eax, 0xd000
0x12de7:    cmp     eax, 0x8000
0x12dec:    je      0x12ee8
0x12df2:    mov     esi, r13d
0x12df5:    lea     rdx, [rsp + 0x18]
0x12dfa:    lea     r9, [rsp + 0x20]
0x12dff:    mov     r8d, r14d
0x12e02:    mov     rcx, rbp
0x12e05:    mov     rdi, rbx
0x12e08:    xor     eax, eax
0x12e0a:    mov     qword ptr [rsp + 8], r10
0x12e0f:    call    0x125c0
0x12e14:    mov     r13, rax
0x12e17:    lea     eax, [rax + 5]
0x12e1a:    cmp     r13d, -5
0x12e1e:    jnb     0x12f17
0x12e24:    lea     rdx, [rip + 0xeac5]
0x12e2b:    mov     r10, qword ptr [rsp + 8]
0x12e30:    movsxd  rax, dword ptr [rdx + rax*4]
0x12e34:    add     rax, rdx
0x12e37:    jmp     rax
rax
->->->                                indirect JMP

```

```

0x510e: movsxd  rax, dword ptr [rip + 0x25733]
0x5115: lea      r13, [rip + 0x1c5c2]
0x511c: cmp      eax, ebp
0x511e: jl       0x6f96
0x5124: mov      r14, r13
0x5127: mov      edi, 0xa
0x512c: lea      rbp, [rip + 0x1b51c]
0x5133: call     0x4810
0x5138: lea      rax, [rip + 0x1b4a5]
0x513f: lea      r9, [rip + 0x1b469]
0x5146: mov      edi, 1
0x514b: push     rsi
0x514c: lea      r8, [rip + 0x1b465]
0x5153: lea      rcx, [rip + 0x1b435]
0x515a: push     rax
0x515b: lea      rdx, [rip + 0x1b403]
0x5162: lea      rsi, [rip + 0x1b468]
0x5169: xor      eax, eax
0x516b: call     0x4d20
0x5170: mov      rsi, qword ptr [rip + 0x24b09]
0x5177: mov      edi, 1
0x517c: xor      eax, eax
0x517e: call     0x4d20
0x5183: xor      eax, eax
0x5185: lea      rdx, [rip + 0x1b464]
0x518c: lea      rsi, [rip + 0x1b464]
0x5193: lea      rdi, [rip + 0x1b459]
0x519a: call     0x92b0
0x519f: xor      eax, eax
0x51a1: lea      rdx, [rip + 0x1b456]
0x51a8: lea      rsi, [rip + 0x1b458]
0x51af: lea      rdi, [rip + 0x1b44d]
0x51b6: call     0x92b0
0x51bb: xor      esi, esi
0x51bd: xor      eax, eax
0x51bf: lea      rdx, [rip + 0x1b447]
0x51c6: lea      rdi, [rip + 0x1b445]
0x51cd: call     0x92b0
0x51d2: xor      eax, eax
0x51d4: lea      rdx, [rip + 0x1b443]
0x51db: lea      rsi, [rip + 0x1b448]
0x51e2: lea      rdi, [rip + 0x1b43d]
0x51e9: call     0x92b0
0x51ee: xor      esi, esi
0x51f0: xor      eax, eax
0x51f2: lea      rdx, [rip + 0x1b437]
0x51f9: lea      rdi, [rip + 0x1b435]
0x5200: call     0x92b0

```

```

0x5205: lea     rdx, [rip + 0x1b433]
0x520c: xor      esi, esi
0x520e: xor      eax, eax
0x5210: lea     rdi, [rip + 0x1b42d]
0x5217: call    0x92b0
0x521c: xor      edi, edi
0x521e: call    0xc3d0
0x5223: xor      eax, eax
0x5225: call    0x179a0
0x522a: mov     rdi, rbp
0x522d: call    0x17c50
0x5232: xor      eax, eax
0x5234: call    0x18660
0x5239: lea     rdi, [rip + 0x1b41b]
0x5240: call    0x17c50
0x5245: lea     rax, [rsp + 0x44]
0x524a: xor      edi, edi
0x524c: mov     rsi, rax
0x524f: mov     qword ptr [rsp + 0x20], rax
0x5254: xor      eax, eax
0x5256: call    0x9310
0x525b: mov     dword ptr [rsp + 0x1c], eax
0x525f: mov     eax, dword ptr [rsp + 0x44]
0x5263: add     dword ptr [rip + 0x25673], eax
0x5269: xor      eax, eax
0x526b: call    0x17280
0x5270: xor      eax, eax
0x5272: call    0x17a30
0x5277: pop     rdi
0x5278: lea     rdi, [rip + 0x1b3e9]
0x527f: pop     r8
0x5281: cmp     eax, 1
0x5284: adc     dword ptr [rip + 0x25651], 0
0x528b: call    0x47e0
0x5290: mov     rbx, rax
0x5293: test    rax, rax
0x5296: je      0x7011
0x529c: mov     rdi, rbx
0x529f: call    0x49b0
0x52a4: lea     rdi, [rax + 0x21]
0x52a8: call    0x1d670
0x52ad: mov     rdi, rbx
0x52b0: mov     qword ptr [rip + 0x26819], rax
0x52b7: call    0x49b0
0x52bc: lea     rdi, [rax + 0x21]
0x52c0: call    0x1d670
0x52c5: xor      edi, edi
0x52c7: mov     qword ptr [rip + 0x267fa], rax

```

```

0x52ce: call    0x4be0
0x52d3: mov     edi, eax
0x52d5: call    0x4ae0
0x52da: call    0x4ee0
0x52df: mov     dword ptr [rsp + 0x1c], eax
0x52e3: call    0x4ee0
0x52e8: mov     dword ptr [rsp + 0x18], eax
0x52ec: call    0x4920
0x52f1: pxor    xmm0, xmm0
0x52f5: mov     r8, rbx
0x52f8: mov     rdi, qword ptr [rip + 0x267d1]
0x52ff: mov     r9d, eax
0x5302: push    rax
0x5303: mov     edx, dword ptr [rsp + 0x24]
0x5307: lea     rcx, [rip + 0x1b361]
0x530e: mov     esi, 1
0x5313: xor     eax, eax
0x5315: cvtsi2sd    xmm0, edx
0x5319: mulsd   xmm0, qword ptr [rip + 0x1b89f]
0x5321: mulsd   xmm0, qword ptr [rip + 0x1b89f]
0x5329: cvtttsd2si    edx, xmm0
0x532d: add     edx, 1
0x5330: push    rdx
0x5331: or      rdx, 0xffffffffffffffff
0x5335: call    0x4f10
0x533a: call    0x4920
0x533f: mov     r10d, dword ptr [rsp + 0x28]
0x5344: pxor    xmm0, xmm0
0x5348: mov     rdi, qword ptr [rip + 0x26779]
0x534f: mov     r9d, eax
0x5352: mov     r8, rbx
0x5355: mov     esi, 1
0x535a: xor     eax, eax
0x535c: cvtsi2sd    xmm0, r10d
0x5361: mulsd   xmm0, qword ptr [rip + 0x1b857]
0x5369: lea     rcx, [rip + 0x1b30e]
0x5370: mulsd   xmm0, qword ptr [rip + 0x1b850]
0x5378: cvtttsd2si    edx, xmm0
0x537c: add     edx, 1
0x537f: mov     dword ptr [rsp], edx
0x5382: or      rdx, 0xffffffffffffffff
0x5386: call    0x4f10
0x538b: lea     rdi, [rip + 0x1b2fb]
0x5392: call    0x17c50
0x5397: pop     rdx
0x5398: pop     rcx
0x5399: cmp     r12d, -1
0x539d: je      0x7092

```

```

0x53a3: cmp     dword ptr [rsp], 0
0x53a7: mov     dword ptr [rip + 0x248b2], r12d
0x53ae: jne     0x6f80
0x53b4: mov     eax, dword ptr [rsp]
0x53b7: mov     edi, eax
0x53b9: mov     dword ptr [rip + 0x25519], eax
0x53bf: call    0x9af0
0x53c4: lea     rdi, [rip + 0x1b2e7]
0x53cb: call    0x17c50
0x53d0: xor     eax, eax
0x53d2: cmp     dword ptr [rip + 0x24887], 1
0x53d9: je      0x7050
0x53df: xor     esi, esi
0x53e1: lea     rdi, [rip + 0x1b2e8]
0x53e8: mov     dword ptr [rip + 0x2486e], eax
0x53ee: call    0x17e50
0x53f3: lea     rsi, [rip + 0x1cfeb]
0x53fa: lea     rdi, [rip + 0x1b2dd]
0x5401: mov     dword ptr [rip + 0x25501], eax
0x5407: call    0x17f10
0x540c: lea     rsi, [rip + 0x1cfd2]
0x5413: lea     rdi, [rip + 0x1b2d3]
0x541a: mov     qword ptr [rip + 0x2484f], rax
0x5421: call    0x17f10
0x5426: cmp     dword ptr [rip + 0x24833], 0
0x542d: lea     rdi, [rip + 0x24814]
0x5434: mov     qword ptr [rip + 0x2482d], rax
0x543b: lea     rax, [rip + 0x247ee]
0x5442: cmov     rdi, rax
0x5446: call    0x17c50
0x544b: mov     edx, 8
0x5450: lea     rsi, [rip + 0x25229]
0x5457: lea     rdi, [rip + 0x251e2]
0x545e: call    0x17f70
0x5463: mov     rdi, rbp
0x5466: call    0x17c50
0x546b: xor     eax, eax
0x546d: call    0x18660
0x5472: lea     rdi, [rip + 0x1b1e2]
0x5479: call    0x17c50
0x547e: mov     rsi, qword ptr [rsp + 0x10]
0x5483: mov     edi, dword ptr [rsp + 0xc]
0x5487: xor     eax, eax
0x5489: call    0x9310
0x548e: mov     edx, dword ptr [rsp + 0x34]
0x5492: add     dword ptr [rip + 0x25444], edx
0x5498: cmp     eax, 0x800
0x549d: je      0x7066

```

```

0x54a3: call    0x49e0
0x54a8: test    eax, eax
0x54aa: jne     0x54b3
0x54ac: mov     byte ptr [rip + 0x24753], 0x23
0x54b3: xor     eax, eax
0x54b5: call    0x18fe0
0x54ba: mov     rbp, rax
0x54bd: test    rax, rax
0x54c0: je      0x73ec
0x54c6: xor     eax, eax
0x54c8: call    0x9e00
0x54cd: xor     eax, eax
0x54cf: call    0x16c10
0x54d4: xor     eax, eax
0x54d6: call    0xd060
0x54db: mov     rdi, qword ptr [rip + 0x2478e]
0x54e2: call    0xddc0
0x54e7: cmp     byte ptr [r13], 0x2f
0x54ec: je      0x7001
0x54f2: mov     rdi, rbp
0x54f5: call    0x49b0
0x54fa: mov     rdi, r13
0x54fd: mov     r12, rax
0x5500: call    0x49b0
0x5505: lea     rdi, [r12 + rax + 2]
0x550a: call    0x1d670
0x550f: mov     r9, r13
0x5512: mov     r8, rbp
0x5515: or      rdx, 0xffffffffffffffff
0x5519: mov     r12, rax
0x551c: mov     rdi, rax
0x551f: mov     esi, 1
0x5524: xor     eax, eax
0x5526: lea     rcx, [rip + 0x1b1d4]
0x552d: call    0x4f10
0x5532: mov     rdi, r12
0x5535: call    0xf160
0x553a: mov     rdi, r12
0x553d: mov     qword ptr [rip + 0x2537c], rax
0x5544: call    0x18d10
0x5549: cmp     byte ptr [r14], 0x2f
0x554d: je      0x6f70
0x5553: mov     rdi, rbp
0x5556: call    0x49b0
0x555b: mov     rdi, r14
0x555e: mov     r12, rax
0x5561: call    0x49b0
0x5566: lea     rdi, [r12 + rax + 2]

```



```

0x556b: call    0x1d670
0x5570: mov     r9, r14
0x5573: mov     r8, rbp
0x5576: or      rdx, 0xffffffffffffffff
0x557a: mov     r12, rax
0x557d: mov     rdi, rax
0x5580: mov     esi, 1
0x5585: xor     eax, eax
0x5587: lea     rcx, [rip + 0x1b173]
0x558e: call    0x4f10
0x5593: mov     rdi, r12
0x5596: xor     r14d, r14d
0x5599: lea     rbx, [rsp + 0x28]
0x559e: call    0xf160
0x55a3: mov     rdi, r12
0x55a6: mov     qword ptr [rip + 0x2530b], rax
0x55ad: call    0x18d10
0x55b2: mov     rdi, rbp
0x55b5: mov     ebp, 1
0x55ba: call    0x18d10
0x55bf: xor     eax, eax
0x55c1: call    0x17280
0x55c6: mov     rax, qword ptr [rip + 0x252f3]
0x55cd: xor     edi, edi
0x55cf: mov     qword ptr [rip + 0x252da], rax
0x55d6: mov     rax, qword ptr [rip + 0x252db]
0x55dd: mov     qword ptr [rip + 0x252c4], rax
0x55e4: xor     eax, eax
0x55e6: call    0x7890
0x55eb: mov     rdi, qword ptr [rip + 0x252d6]
0x55f2: call    0xbb10
0x55f7: mov     edi, 1
0x55fc: call    0x9b00
0x5601: xor     eax, eax
0x5603: call    0xa6c0
0x5608: mov     edi, 1
0x560d: mov     qword ptr [rip + 0x25280], 0
0x5618: mov     dword ptr [rip + 0x25272], 0
0x5622: mov     dword ptr [rip + 0x25264], 0
0x562c: call    0x18b90
0x5631: mov     rdi, qword ptr [rip + 0x25278]
0x5638: call    0x11100
0x563d: mov     rdi, rax
0x5640: call    0xc4b0
0x5645: lea     rax, [rsp + 0x50]
0x564a: mov     dword ptr [rsp + 0x10], r15d
0x564f: mov     qword ptr [rsp], 0
0x5657: mov     r15, rax

```

```

0x565a: cmp     dword ptr [rip + 0x2527b], 0
0x5661: jne     0x6d10
0x5667: mov     rdi, qword ptr [rip + 0x25242]
0x566e: call    0x11100
0x5673: mov     rdi, rax
0x5676: call    0xc4b0
0x567b: xor     eax, eax
0x567d: call    0x18330
0x5682: mov     edi, 0x3c
0x5687: sub     edi, dword ptr [rax]
0x5689: call    0x4aa0
0x568e: cmp     dword ptr [rip + 0x2524b], 0
0x5695: je      0x6ba2
0x569b: mov     rax, qword ptr [rip + 0x25216]
0x56a2: mov     qword ptr [rip + 0x25207], rax
0x56a9: mov     rax, qword ptr [rip + 0x25210]
0x56b0: cmp     dword ptr [rip + 0x252e5], 6
0x56b7: mov     qword ptr [rip + 0x251ea], rax
0x56be: jle     0x6d75
0x56c4: xor     eax, eax
0x56c6: call    0x16d50
0x56cb: xor     eax, eax
0x56cd: call    0xe4e0
0x56d2: mov     rdi, r14
0x56d5: call    0xc970
0x56da: xor     eax, eax
0x56dc: call    0xa820
0x56e1: xor     eax, eax
0x56e3: call    0xc8c0
0x56e8: mov     r12, rax
0x56eb: xor     eax, eax
0x56ed: call    0x96c0
0x56f2: xor     edi, edi
0x56f4: xor     eax, eax
0x56f6: call    0x7bd0
0x56fb: mov     rdi, r12
0x56fe: call    0xc970
0x5703: mov     rdi, qword ptr [rip + 0x251a6]
0x570a: mov     esi, 1
0x570f: call    0x14d90
0x5714: test    ebp, ebp
0x5716: jne     0x6d5a
0x571c: xor     eax, eax
0x571e: call    0x9640
0x5723: xor     eax, eax
0x5725: call    0xc8c0
0x572a: mov     r14, rax
0x572d: nop     dword ptr [rax]

```

```

0x5730: mov     rdi, r14
0x5733: call    0xc970
0x5738: xor     eax, eax
0x573a: call    0xc8c0
0x573f: mov     r14, rax
0x5742: xor     eax, eax
0x5744: call    0xd870
0x5749: xor     eax, eax
0x574b: call    0xd740
0x5750: xor     eax, eax
0x5752: call    0xa820
0x5757: mov     rdi, r15
0x575a: call    0xdb10
0x575f: mov     dword ptr [rip + 0x26347], 0
0x5769: jmp     0x5779
0x5779:
0x576b: nop     dword ptr [rax + rax]
0x5770: xor     edi, edi
0x5772: xor     eax, eax
0x5774: call    0x7d10
0x5779: mov     rdi, rbx
0x577c: call    0xb180
0x5781: test    rax, rax
0x5784: je      0x5770
0x5786: mov     r12, rax
0x5789: xor     eax, eax
0x578b: call    0xdea0
0x5790: mov     rax, qword ptr [r12]
0x5794: mov     r13, qword ptr [r12 + 0x10]
0x5799: movzx   ebp, byte ptr [rax]
0x579c: test    r13, r13
0x579f: je      0x57dc
0x57a1: cmp     byte ptr [r13 + 0x1b], 0
0x57a6: mov     rax, qword ptr [r13]
0x57aa: je      0x67c0
0x57b0: lea     rdx, [rip + 0x23869]
0x57b7: sub     rax, rdx
0x57ba: movabs  rdx, 0xea0ea0ea0ea0ea1
0x57c4: mov     rcx, rax
0x57c7: imul    rdx
0x57ca: mov     rax, rcx
0x57cd: sar     rax, 0x3f
0x57d1: sar     rdx, 1
0x57d4: mov     rbp, rdx
0x57d7: sub     rbp, rax
0x57da: not     ebp
0x57dc: lea     eax, [rbp + 0x57]
0x57df: cmp     eax, 0x5e

```

```

0x57e2: ja      0x57f5
0x57e4: lea      rcx, [rip + 0x1aaf1]
0x57eb: movsxd   rax, dword ptr [rcx + rax*4]
0x57ef: add      rax, rcx
0x57f2: jmp      rax
rax
                                ->->->                                indirect JMP
0x5057: mov      rcx, qword ptr [rip + 0x2588a]
0x505e: mov      rdi, qword ptr [rip + 0x2581b]
0x5065: lea      rdx, [rip + 0x1b529]
0x506c: xor      eax, eax
0x506e: mov      esi, 1
0x5073: mov      r15d, 1
0x5079: call     0x4e60
0x507e: mov      rax, qword ptr [rsp + 0x78]
0x5083: sub      rax, qword ptr fs:[0x28]
0x508c: jne      0x7475
0x5092: add      rsp, 0x88
0x5099: mov      eax, r15d
0x509c: pop      rbx
0x509d: pop      rbp
0x509e: pop      r12
0x50a0: pop      r13
0x50a2: pop      r14
0x50a4: pop      r15
0x50a6: ret
0x50a7: lea      rcx, [rip + 0x1b4e1]
0x50ae: mov      edi, 1
0x50b3: xor      eax, eax
0x50b5: xor      r15d, r15d
0x50b8: lea      rdx, [rip + 0x1b4a6]
0x50bf: lea      rsi, [rip + 0x1b517]
0x50c6: call     0x4d20
0x50cb: jmp      0x507e
0x507e
0x50cd: mov      r15d, 1
0x50d3: jmp      0x502e
0x502e
0x50d8: xor      eax, eax
0x50da: xor      r15d, r15d
0x50dd: call     0x9720
0x50e2: jmp      0x507e
0x507e
0x50e4: mov      r12d, 1
0x50ea: jmp      0x502e
0x502e
0x50ef: mov      dword ptr [rsp], 0
0x50f6: jmp      0x502e

```

```

0x502e
0x50fb: mov     r15d, 1
0x5101: jmp     0x507e
0x507e
0x5106: xor     r12d, r12d
0x5109: jmp     0x502e
0x502e
0x510e: movsxd  rax, dword ptr [rip + 0x25733]
0x5115: lea     r13, [rip + 0x1c5c2]
0x511c: cmp     eax, ebp
0x511e: jl      0x6f96
0x5124: mov     r14, r13
0x5127: mov     edi, 0xa
0x512c: lea     rbp, [rip + 0x1b51c]
0x5133: call    0x4810
0x5138: lea     rax, [rip + 0x1b4a5]
0x513f: lea     r9, [rip + 0x1b469]
0x5146: mov     edi, 1
0x514b: push    rsi
0x514c: lea     r8, [rip + 0x1b465]
0x5153: lea     rcx, [rip + 0x1b435]
0x515a: push    rax
0x515b: lea     rdx, [rip + 0x1b403]
0x5162: lea     rsi, [rip + 0x1b468]
0x5169: xor     eax, eax
0x516b: call    0x4d20
0x5170: mov     rsi, qword ptr [rip + 0x24b09]
0x5177: mov     edi, 1
0x517c: xor     eax, eax
0x517e: call    0x4d20
0x5183: xor     eax, eax
0x5185: lea     rdx, [rip + 0x1b464]
0x518c: lea     rsi, [rip + 0x1b464]
0x5193: lea     rdi, [rip + 0x1b459]
0x519a: call    0x92b0
0x519f: xor     eax, eax
0x51a1: lea     rdx, [rip + 0x1b456]
0x51a8: lea     rsi, [rip + 0x1b458]
0x51af: lea     rdi, [rip + 0x1b44d]
0x51b6: call    0x92b0
0x51bb: xor     esi, esi
0x51bd: xor     eax, eax
0x51bf: lea     rdx, [rip + 0x1b447]
0x51c6: lea     rdi, [rip + 0x1b445]
0x51cd: call    0x92b0
0x51d2: xor     eax, eax
0x51d4: lea     rdx, [rip + 0x1b443]
0x51db: lea     rsi, [rip + 0x1b448]

```

```

0x51e2: lea     rdi, [rip + 0x1b43d]
0x51e9: call    0x92b0
0x51ee: xor     esi, esi
0x51f0: xor     eax, eax
0x51f2: lea     rdx, [rip + 0x1b437]
0x51f9: lea     rdi, [rip + 0x1b435]
0x5200: call    0x92b0
0x5205: lea     rdx, [rip + 0x1b433]
0x520c: xor     esi, esi
0x520e: xor     eax, eax
0x5210: lea     rdi, [rip + 0x1b42d]
0x5217: call    0x92b0
0x521c: xor     edi, edi
0x521e: call    0xc3d0
0x5223: xor     eax, eax
0x5225: call    0x179a0
0x522a: mov     rdi, rbp
0x522d: call    0x17c50
0x5232: xor     eax, eax
0x5234: call    0x18660
0x5239: lea     rdi, [rip + 0x1b41b]
0x5240: call    0x17c50
0x5245: lea     rax, [rsp + 0x44]
0x524a: xor     edi, edi
0x524c: mov     rsi, rax
0x524f: mov     qword ptr [rsp + 0x20], rax
0x5254: xor     eax, eax
0x5256: call    0x9310
0x525b: mov     dword ptr [rsp + 0x1c], eax
0x525f: mov     eax, dword ptr [rsp + 0x44]
0x5263: add     dword ptr [rip + 0x25673], eax
0x5269: xor     eax, eax
0x526b: call    0x17280
0x5270: xor     eax, eax
0x5272: call    0x17a30
0x5277: pop     rdi
0x5278: lea     rdi, [rip + 0x1b3e9]
0x527f: pop     r8
0x5281: cmp     eax, 1
0x5284: adc     dword ptr [rip + 0x25651], 0
0x528b: call    0x47e0
0x5290: mov     rbx, rax
0x5293: test    rax, rax
0x5296: je      0x7011
0x529c: mov     rdi, rbx
0x529f: call    0x49b0
0x52a4: lea     rdi, [rax + 0x21]
0x52a8: call    0x1d670

```

```

0x52ad: mov     rdi, rbx
0x52b0: mov     qword ptr [rip + 0x26819], rax
0x52b7: call    0x49b0
0x52bc: lea     rdi, [rax + 0x21]
0x52c0: call    0x1d670
0x52c5: xor     edi, edi
0x52c7: mov     qword ptr [rip + 0x267fa], rax
0x52ce: call    0x4be0
0x52d3: mov     edi, eax
0x52d5: call    0x4ae0
0x52da: call    0x4ee0
0x52df: mov     dword ptr [rsp + 0x1c], eax
0x52e3: call    0x4ee0
0x52e8: mov     dword ptr [rsp + 0x18], eax
0x52ec: call    0x4920
0x52f1: pxor    xmm0, xmm0
0x52f5: mov     r8, rbx
0x52f8: mov     rdi, qword ptr [rip + 0x267d1]
0x52ff: mov     r9d, eax
0x5302: push    rax
0x5303: mov     edx, dword ptr [rsp + 0x24]
0x5307: lea     rcx, [rip + 0x1b361]
0x530e: mov     esi, 1
0x5313: xor     eax, eax
0x5315: cvtsi2sd    xmm0, edx
0x5319: mulsd   xmm0, qword ptr [rip + 0x1b89f]
0x5321: mulsd   xmm0, qword ptr [rip + 0x1b89f]
0x5329: cvttsd2si    edx, xmm0
0x532d: add     edx, 1
0x5330: push    rdx
0x5331: or      rdx, 0xffffffffffffffff
0x5335: call    0x4f10
0x533a: call    0x4920
0x533f: mov     r10d, dword ptr [rsp + 0x28]
0x5344: pxor    xmm0, xmm0
0x5348: mov     rdi, qword ptr [rip + 0x26779]
0x534f: mov     r9d, eax
0x5352: mov     r8, rbx
0x5355: mov     esi, 1
0x535a: xor     eax, eax
0x535c: cvtsi2sd    xmm0, r10d
0x5361: mulsd   xmm0, qword ptr [rip + 0x1b857]
0x5369: lea     rcx, [rip + 0x1b30e]
0x5370: mulsd   xmm0, qword ptr [rip + 0x1b850]
0x5378: cvttsd2si    edx, xmm0
0x537c: add     edx, 1
0x537f: mov     dword ptr [rsp], edx
0x5382: or      rdx, 0xffffffffffffffff

```

```

0x5386: call    0x4f10
0x538b: lea      rdi, [rip + 0x1b2fb]
0x5392: call    0x17c50
0x5397: pop      rdx
0x5398: pop      rcx
0x5399: cmp      r12d, -1
0x539d: je       0x7092
0x53a3: cmp      dword ptr [rsp], 0
0x53a7: mov      dword ptr [rip + 0x248b2], r12d
0x53ae: jne      0x6f80
0x53b4: mov      eax, dword ptr [rsp]
0x53b7: mov      edi, eax
0x53b9: mov      dword ptr [rip + 0x25519], eax
0x53bf: call    0x9af0
0x53c4: lea      rdi, [rip + 0x1b2e7]
0x53cb: call    0x17c50
0x53d0: xor      eax, eax
0x53d2: cmp      dword ptr [rip + 0x24887], 1
0x53d9: je       0x7050
0x53df: xor      esi, esi
0x53e1: lea      rdi, [rip + 0x1b2e8]
0x53e8: mov      dword ptr [rip + 0x2486e], eax
0x53ee: call    0x17e50
0x53f3: lea      rsi, [rip + 0x1cfeb]
0x53fa: lea      rdi, [rip + 0x1b2dd]
0x5401: mov      dword ptr [rip + 0x25501], eax
0x5407: call    0x17f10
0x540c: lea      rsi, [rip + 0x1cfd2]
0x5413: lea      rdi, [rip + 0x1b2d3]
0x541a: mov      qword ptr [rip + 0x2484f], rax
0x5421: call    0x17f10
0x5426: cmp      dword ptr [rip + 0x24833], 0
0x542d: lea      rdi, [rip + 0x24814]
0x5434: mov      qword ptr [rip + 0x2482d], rax
0x543b: lea      rax, [rip + 0x247ee]
0x5442: cmove    rdi, rax
0x5446: call    0x17c50
0x544b: mov      edx, 8
0x5450: lea      rsi, [rip + 0x25229]
0x5457: lea      rdi, [rip + 0x251e2]
0x545e: call    0x17f70
0x5463: mov      rdi, rbp
0x5466: call    0x17c50
0x546b: xor      eax, eax
0x546d: call    0x18660
0x5472: lea      rdi, [rip + 0x1b1e2]
0x5479: call    0x17c50
0x547e: mov      rsi, qword ptr [rsp + 0x10]

```



```

0x5483: mov     edi, dword ptr [rsp + 0xc]
0x5487: xor      eax, eax
0x5489: call     0x9310
0x548e: mov     edx, dword ptr [rsp + 0x34]
0x5492: add     dword ptr [rip + 0x25444], edx
0x5498: cmp     eax, 0x800
0x549d: je      0x7066
0x54a3: call     0x49e0
0x54a8: test    eax, eax
0x54aa: jne     0x54b3
0x54ac: mov     byte ptr [rip + 0x24753], 0x23
0x54b3: xor     eax, eax
0x54b5: call     0x18fe0
0x54ba: mov     rbp, rax
0x54bd: test    rax, rax
0x54c0: je      0x73ec
0x54c6: xor     eax, eax
0x54c8: call     0x9e00
0x54cd: xor     eax, eax
0x54cf: call     0x16c10
0x54d4: xor     eax, eax
0x54d6: call     0xd060
0x54db: mov     rdi, qword ptr [rip + 0x2478e]
0x54e2: call     0xddc0
0x54e7: cmp     byte ptr [r13], 0x2f
0x54ec: je      0x7001
0x54f2: mov     rdi, rbp
0x54f5: call     0x49b0
0x54fa: mov     rdi, r13
0x54fd: mov     r12, rax
0x5500: call     0x49b0
0x5505: lea     rdi, [r12 + rax + 2]
0x550a: call     0x1d670
0x550f: mov     r9, r13
0x5512: mov     r8, rbp
0x5515: or      rdx, 0xffffffffffffffff
0x5519: mov     r12, rax
0x551c: mov     rdi, rax
0x551f: mov     esi, 1
0x5524: xor     eax, eax
0x5526: lea     rcx, [rip + 0x1b1d4]
0x552d: call     0x4f10
0x5532: mov     rdi, r12
0x5535: call     0xf160
0x553a: mov     rdi, r12
0x553d: mov     qword ptr [rip + 0x2537c], rax
0x5544: call     0x18d10
0x5549: cmp     byte ptr [r14], 0x2f

```

```

0x554d: je      0x6f70
0x5553: mov     rdi, rbp
0x5556: call    0x49b0
0x555b: mov     rdi, r14
0x555e: mov     r12, rax
0x5561: call    0x49b0
0x5566: lea     rdi, [r12 + rax + 2]
0x556b: call    0x1d670
0x5570: mov     r9, r14
0x5573: mov     r8, rbp
0x5576: or      rdx, 0xffffffffffffffff
0x557a: mov     r12, rax
0x557d: mov     rdi, rax
0x5580: mov     esi, 1
0x5585: xor     eax, eax
0x5587: lea     rcx, [rip + 0x1b173]
0x558e: call    0x4f10
0x5593: mov     rdi, r12
0x5596: xor     r14d, r14d
0x5599: lea     rbx, [rsp + 0x28]
0x559e: call    0xf160
0x55a3: mov     rdi, r12
0x55a6: mov     qword ptr [rip + 0x2530b], rax
0x55ad: call    0x18d10
0x55b2: mov     rdi, rbp
0x55b5: mov     ebp, 1
0x55ba: call    0x18d10
0x55bf: xor     eax, eax
0x55c1: call    0x17280
0x55c6: mov     rax, qword ptr [rip + 0x252f3]
0x55cd: xor     edi, edi
0x55cf: mov     qword ptr [rip + 0x252da], rax
0x55d6: mov     rax, qword ptr [rip + 0x252db]
0x55dd: mov     qword ptr [rip + 0x252c4], rax
0x55e4: xor     eax, eax
0x55e6: call    0x7890
0x55eb: mov     rdi, qword ptr [rip + 0x252d6]
0x55f2: call    0xbb10
0x55f7: mov     edi, 1
0x55fc: call    0x9b00
0x5601: xor     eax, eax
0x5603: call    0xa6c0
0x5608: mov     edi, 1
0x560d: mov     qword ptr [rip + 0x25280], 0
0x5618: mov     dword ptr [rip + 0x25272], 0
0x5622: mov     dword ptr [rip + 0x25264], 0
0x562c: call    0x18b90
0x5631: mov     rdi, qword ptr [rip + 0x25278]

```

```

0x5638: call    0x11100
0x563d: mov     rdi, rax
0x5640: call    0xc4b0
0x5645: lea     rax, [rsp + 0x50]
0x564a: mov     dword ptr [rsp + 0x10], r15d
0x564f: mov     qword ptr [rsp], 0
0x5657: mov     r15, rax
0x565a: cmp     dword ptr [rip + 0x2527b], 0
0x5661: jne     0x6d10
0x5667: mov     rdi, qword ptr [rip + 0x25242]
0x566e: call    0x11100
0x5673: mov     rdi, rax
0x5676: call    0xc4b0
0x567b: xor     eax, eax
0x567d: call    0x18330
0x5682: mov     edi, 0x3c
0x5687: sub     edi, dword ptr [rax]
0x5689: call    0x4aa0
0x568e: cmp     dword ptr [rip + 0x2524b], 0
0x5695: je      0x6ba2
0x569b: mov     rax, qword ptr [rip + 0x25216]
0x56a2: mov     qword ptr [rip + 0x25207], rax
0x56a9: mov     rax, qword ptr [rip + 0x25210]
0x56b0: cmp     dword ptr [rip + 0x252e5], 6
0x56b7: mov     qword ptr [rip + 0x251ea], rax
0x56be: jle     0x6d75
0x56c4: xor     eax, eax
0x56c6: call    0x16d50
0x56cb: xor     eax, eax
0x56cd: call    0xe4e0
0x56d2: mov     rdi, r14
0x56d5: call    0xc970
0x56da: xor     eax, eax
0x56dc: call    0xa820
0x56e1: xor     eax, eax
0x56e3: call    0xc8c0
0x56e8: mov     r12, rax
0x56eb: xor     eax, eax
0x56ed: call    0x96c0
0x56f2: xor     edi, edi
0x56f4: xor     eax, eax
0x56f6: call    0x7bd0
0x56fb: mov     rdi, r12
0x56fe: call    0xc970
0x5703: mov     rdi, qword ptr [rip + 0x251a6]
0x570a: mov     esi, 1
0x570f: call    0x14d90
0x5714: test    ebp, ebp

```

```

0x5716: jne      0x6d5a
0x571c: xor        eax, eax
0x571e: call       0x9640
0x5723: xor        eax, eax
0x5725: call       0xc8c0
0x572a: mov        r14, rax
0x572d: nop        dword ptr [rax]
0x5730: mov        rdi, r14
0x5733: call       0xc970
0x5738: xor        eax, eax
0x573a: call       0xc8c0
0x573f: mov        r14, rax
0x5742: xor        eax, eax
0x5744: call       0xd870
0x5749: xor        eax, eax
0x574b: call       0xd740
0x5750: xor        eax, eax
0x5752: call       0xa820
0x5757: mov        rdi, r15
0x575a: call       0xdb10
0x575f: mov        dword ptr [rip + 0x26347], 0
0x5769: jmp        0x5779
0x5779:
0x576b: nop        dword ptr [rax + rax]
0x5770: xor        edi, edi
0x5772: xor        eax, eax
0x5774: call       0x7d10
0x5779: mov        rdi, rbx
0x577c: call       0xb180
0x5781: test       rax, rax
0x5784: je         0x5770
0x5786: mov        r12, rax
0x5789: xor        eax, eax
0x578b: call       0xdea0
0x5790: mov        rax, qword ptr [r12]
0x5794: mov        r13, qword ptr [r12 + 0x10]
0x5799: movzx     ebp, byte ptr [rax]
0x579c: test       r13, r13
0x579f: je         0x57dc
0x57a1: cmp        byte ptr [r13 + 0x1b], 0
0x57a6: mov        rax, qword ptr [r13]
0x57aa: je         0x67c0
0x57b0: lea        rdx, [rip + 0x23869]
0x57b7: sub        rax, rdx
0x57ba: movabs     rdx, 0xea0ea0ea0ea0ea1
0x57c4: mov        rcx, rax
0x57c7: imul       rdx
0x57ca: mov        rax, rcx

```

```

0x57cd: sar    rax, 0x3f
0x57d1: sar    rdx, 1
0x57d4: mov    rbp, rdx
0x57d7: sub    rbp, rax
0x57da: not    ebp
0x57dc: lea    eax, [rbp + 0x57]
0x57df: cmp    eax, 0x5e
0x57e2: ja     0x57f5
0x57e4: lea    rcx, [rip + 0x1aaf1]
0x57eb: movsxd  rax, dword ptr [rcx + rax*4]
0x57ef: add    rax, rcx
0x57f2: jmp    rax
rax
                                ->->->                                indirect JMP
0xc600: endbr64
0xc604: mov    rax, rdi
0xc607: mov    rdi, rsi
0xc60a: mov    esi, edx
0xc60c: mov    ecx, dword ptr [rax + 0x10]
0xc60f: lea    r8d, [rcx + rdx]
0xc613: mov    edx, dword ptr [rax + 0xc]
0xc616: mov    dword ptr [rax + 0x10], r8d
0xc61a: cmp    edx, ecx
0xc61c: jle    0xc640
0xc61e: mov    r9d, dword ptr [rax + 8]
0xc622: cmp    dword ptr [rax + 0x14], r9d
0xc626: jge    0xc640
0xc628: cmp    r8d, edx
0xc62b: jle    0xc631
0xc62d: sub    edx, ecx
0xc62f: mov    esi, edx
0xc631: jmp    0xa1a0
0xa1a0
0xc636: nop    word ptr cs:[rax + rax]
0xc640: xor    eax, eax
0xc642: ret
0xc643: nop    word ptr cs:[rax + rax]
0xc64e: nop
0xc650: endbr64
0xc654: mov    ecx, dword ptr [rdi + 0x10]
0xc657: lea    edx, [rcx + 1]
0xc65a: mov    dword ptr [rdi + 0x10], edx
0xc65d: cmp    edx, dword ptr [rdi + 0xc]
0xc660: jg     0xc66a
0xc662: mov    ecx, dword ptr [rdi + 8]
0xc665: cmp    dword ptr [rdi + 0x14], ecx
0xc668: jl     0xc670
0xc66a: xor    eax, eax

```

```

0xc66c: ret
0xc66d: nop      dword ptr [rax]
0xc670: mov      edi, esi
0xc672: jmp      0xa230
0xa230
0xc677: nop      word ptr [rax + rax]
0xc680: endbr64
0xc684: mov      rax, rdi
0xc687: mov      edi, esi
0xc689: mov      dword ptr [rax + 0x10], edx
0xc68c: add      edx, dword ptr [rax]
0xc68e: mov      dword ptr [rax + 0x14], esi
0xc691: add      edi, dword ptr [rax + 4]
0xc694: mov      esi, edx
0xc696: jmp      0xa4b0
0xa4b0
0xc69b: nop      dword ptr [rax + rax]
0xc6a0: endbr64
0xc6a4: mov      eax, dword ptr [rdi]
0xc6a6: ret
0xc6a7: nop      word ptr [rax + rax]
0xc6b0: endbr64
0xc6b4: mov      eax, dword ptr [rdi + 4]
0xc6b7: ret
0xc6b8: nop      dword ptr [rax + rax]
0xc6c0: endbr64
0xc6c4: mov      eax, dword ptr [rdi + 8]
0xc6c7: ret
0xc6c8: nop      dword ptr [rax + rax]
0xc6d0: endbr64
0xc6d4: mov      eax, dword ptr [rdi + 0xc]
0xc6d7: ret
0xc6d8: nop      dword ptr [rax + rax]
0xc6e0: push     r14
0xc6e2: push     r13
0xc6e4: push     r12
0xc6e6: mov      r12, qword ptr [rip + 0x1ed1b]
0xc6ed: push     rbp
0xc6ee: push     rbx
0xc6ef: mov      r13, qword ptr [r12 + 0x20]
0xc6f4: mov      rbx, qword ptr [r12 + 0x18]
0xc6f9: cmp      r13, rbx
0xc6fc: je       0xc761
0xc6fe: mov      ebp, edi
0xc700: jbe      0xc770
0xc702: test     bpl, 1
0xc706: je       0xc75c
0xc708: mov      rdi, qword ptr [r12 + 0x60]

```

```

0xc70d: test    rdi, rdi
0xc710: je        0xc71e
0xc712: call     0x18d10
0xc717: mov     r12, qword ptr [rip + 0x1ecea]
0xc71e: mov     r14, r13
0xc721: sub     r14, rbx
0xc724: lea     rdi, [r14 + 1]
0xc728: call    0x1d670
0xc72d: mov     rdx, r14
0xc730: mov     qword ptr [r12 + 0x60], rax
0xc735: mov     rax, qword ptr [rip + 0x1eccc]
0xc73c: mov     rsi, qword ptr [rax + 0x58]
0xc740: mov     rdi, qword ptr [rax + 0x60]
0xc744: add     rsi, rbx
0xc747: call    0x4bb0
0xc74c: mov     rax, qword ptr [rip + 0x1ecb5]
0xc753: mov     rax, qword ptr [rax + 0x60]
0xc757: mov     byte ptr [rax + r14], 0
0xc75c: and     ebp, 2
0xc75f: jne     0xc788
0xc761: pop     rbx
0xc762: pop     rbp
0xc763: pop     r12
0xc765: pop     r13
0xc767: pop     r14
0xc769: ret
0xc76a: nop     word ptr [rax + rax]
0xc770: mov     rax, rbx
0xc773: mov     rbx, r13
0xc776: mov     r13, rax
0xc779: test    bpl, 1
0xc77d: je      0xc75c
0xc77f: jmp     0xc708
0xc708:
0xc781: nop     dword ptr [rax]
0xc788: mov     rax, qword ptr [rip + 0x1ec79]
0xc78f: mov     rdx, rbx
0xc792: sub     rdx, r13
0xc795: add     qword ptr [rax + 0x48], rdx
0xc799: mov     rdi, qword ptr [rax + 0x58]
0xc79d: add     qword ptr [rax + 0x38], rdx
0xc7a1: mov     qword ptr [rax + 0x18], rbx
0xc7a5: lea     rsi, [rdi + r13]
0xc7a9: add     rdi, rbx
0xc7ac: mov     qword ptr [rax + 0x20], rbx
0xc7b0: call    0x48b0
0xc7b5: mov     rbx, qword ptr [rip + 0x1ec4c]
0xc7bc: mov     rax, qword ptr [rbx + 0x38]

```

```

0xc7c0: mov     rdi, qword ptr [rbx + 0x58]
0xc7c4: lea     rsi, [rax + 1]
0xc7c8: mov     qword ptr [rbx + 0x50], rsi
0xc7cc: call    0x1d690
0xc7d1: mov     qword ptr [rbx + 0x58], rax
0xc7d5: pop     rbx
0xc7d6: pop     rbp
0xc7d7: pop     r12
0xc7d9: pop     r13
0xc7db: pop     r14
0xc7dd: ret
0xc7de: nop
0xc7e0: endbr64
0xc7e4: test    rdi, rdi
0xc7e7: je      0xc818
0xc7e9: push    rbp
0xc7ea: mov     rbp, rdi
0xc7ed: mov     rdi, qword ptr [rdi + 0x58]
0xc7f1: test    rdi, rdi
0xc7f4: je      0xc7fb
0xc7f6: call    0x18d10
0xc7fb: mov     rdi, qword ptr [rbp + 0x60]
0xc7ff: test    rdi, rdi
0xc802: je      0xc809
0xc804: call    0x18d10
0xc809: mov     rdi, rbp
0xc80c: pop     rbp
0xc80d: jmp     0x18d10
0x18d10
0xc812: nop     word ptr [rax + rax]
0xc818: ret
0xc819: nop     dword ptr [rax]
0xc820: endbr64
0xc824: mov     rax, qword ptr [rip + 0x1ebdd]
0xc82b: push    rbp
0xc82c: mov     rdi, qword ptr [rax + 0x68]
0xc830: xor     eax, eax
0xc832: call    0x1aac0
0xc837: mov     rax, qword ptr [rip + 0x1ebca]
0xc83e: mov     rdi, qword ptr [rax]
0xc841: call    0xc5c0
0xc846: mov     rbp, qword ptr [rip + 0x1ebbb]
0xc84d: test    rbp, rbp
0xc850: je      0xc876
0xc852: mov     rdi, qword ptr [rbp + 0x58]
0xc856: test    rdi, rdi
0xc859: je      0xc860
0xc85b: call    0x18d10

```



```

0xc860: mov     rdi, qword ptr [rbp + 0x60]
0xc864: test    rdi, rdi
0xc867: je      0xc86e
0xc869: call   0x18d10
0xc86e: mov     rdi, rbp
0xc871: call   0x18d10
0xc876: mov     qword ptr [rip + 0x1eb87], 0
0xc881: pop     rbp
0xc882: ret
0xc883: nop     word ptr cs:[rax + rax]
0xc88e: nop
0xc890: endbr64
0xc894: mov     rax, qword ptr [rip + 0x1eb6d]
0xc89b: movsxd  rcx, edi
0xc89e: mov     edx, esi
0xc8a0: mov     r8, rcx
0xc8a3: mov     qword ptr [rax + 0x28], rcx
0xc8a7: movsxd  rcx, esi
0xc8aa: mov     rdi, qword ptr [rax]
0xc8ad: xor     esi, esi
0xc8af: mov     qword ptr [rax + 0x30], rcx
0xc8b3: mov     ecx, 1
0xc8b8: jmp     0xc5e0
0xc5e0
0xc8bd: nop     dword ptr [rax]
0xc8c0: endbr64
0xc8c4: push    r12
0xc8c6: mov     edi, 0x70
0xc8cb: push    rbx
0xc8cc: sub     rsp, 8
0xc8d0: call   0x1d670
0xc8d5: mov     rbx, qword ptr [rip + 0x1eb2c]
0xc8dc: mov     r12, rax
0xc8df: movdqu  xmm0, xmmword ptr [rbx]
0xc8e3: movups  xmmword ptr [rax], xmm0
0xc8e6: movdqu  xmm1, xmmword ptr [rbx + 0x10]
0xc8eb: movups  xmmword ptr [rax + 0x10], xmm1
0xc8ef: movdqu  xmm2, xmmword ptr [rbx + 0x20]
0xc8f4: movups  xmmword ptr [rax + 0x20], xmm2
0xc8f8: movdqu  xmm3, xmmword ptr [rbx + 0x30]
0xc8fd: movups  xmmword ptr [rax + 0x30], xmm3
0xc901: movdqu  xmm4, xmmword ptr [rbx + 0x40]
0xc906: movups  xmmword ptr [rax + 0x40], xmm4
0xc90a: movdqu  xmm5, xmmword ptr [rbx + 0x50]
0xc90f: movups  xmmword ptr [rax + 0x50], xmm5
0xc913: movdqu  xmm6, xmmword ptr [rbx + 0x60]
0xc918: cmp     qword ptr [rax + 0x58], 0
0xc91d: movups  xmmword ptr [rax + 0x60], xmm6

```

```

0xc921: je      0xc94a
0xc923: mov     rdi, qword ptr [rax + 0x50]
0xc927: call    0x1d670
0xc92c: mov     rdx, qword ptr [r12 + 0x50]
0xc931: mov     qword ptr [rbx + 0x58], rax
0xc935: mov     rax, qword ptr [rip + 0x1eacc]
0xc93c: mov     rsi, qword ptr [r12 + 0x58]
0xc941: mov     rdi, qword ptr [rax + 0x58]
0xc945: call    0x4bb0
0xc94a: mov     rdi, qword ptr [r12 + 0x60]
0xc94f: test    rdi, rdi
0xc952: je      0xc964
0xc954: mov     rbx, qword ptr [rip + 0x1eaad]
0xc95b: call    0x1d7a0
0xc960: mov     qword ptr [rbx + 0x60], rax
0xc964: add     rsp, 8
0xc968: mov     rax, r12
0xc96b: pop     rbx
0xc96c: pop     r12
0xc96e: ret
0xc96f: nop
0xc970: endbr64
0xc974: test    rdi, rdi
0xc977: je      0xc9f0
0xc979: push    r13
0xc97b: push    r12
0xc97d: push    rbp
0xc97e: push    rbx
0xc97f: mov     rbx, rdi
0xc982: sub     rsp, 8
0xc986: mov     rbp, qword ptr [rip + 0x1ea7b]
0xc98d: mov     rdi, qword ptr [rbp + 0x58]
0xc991: mov     r13, qword ptr [rbp + 0x28]
0xc995: mov     r12, qword ptr [rbp + 0x30]
0xc999: test    rdi, rdi
0xc99c: je      0xc9a3
0xc99e: call    0x18d10
0xc9a3: mov     rdi, qword ptr [rbp + 0x60]
0xc9a7: test    rdi, rdi
0xc9aa: je      0xc9b1
0xc9ac: call    0x18d10
0xc9b1: mov     rdi, rbp
0xc9b4: call    0x18d10
0xc9b9: movsxd  rax, r13d
0xc9bc: mov     edx, r12d
0xc9bf: movsxd  r12, r12d
0xc9c2: mov     qword ptr [rbx + 0x28], rax
0xc9c6: mov     rdi, qword ptr [rbx]

```

```

0xc9c9: mov     r8d, r13d
0xc9cc: mov     ecx, 1
0xc9d1: mov     qword ptr [rbx + 0x30], r12
0xc9d5: xor     esi, esi
0xc9d7: mov     qword ptr [rip + 0x1ea2a], rbx
0xc9de: add     rsp, 8
0xc9e2: pop     rbx
0xc9e3: pop     rbp
0xc9e4: pop     r12
0xc9e6: pop     r13
0xc9e8: jmp     0xc5e0
0xc5e0
0xc9ed: nop     dword ptr [rax]
0xc9f0: ret
0xc9f1: nop     word ptr cs:[rax + rax]
0xc9fc: nop     dword ptr [rax]
0xca00: endbr64
0xca04: mov     rdx, qword ptr [rip + 0x1e9fd]
0xca0b: mov     rax, qword ptr [rdx + 0x18]
0xca0f: sub     rax, qword ptr [rdx + 0x40]
0xca13: ret
0xca14: nop     word ptr cs:[rax + rax]
0xca1f: nop
0xca20: endbr64
0xca24: mov     rax, qword ptr [rip + 0x1e9dd]
0xca2b: mov     r8d, dword ptr [rax + 8]
0xca2f: mov     dword ptr [rax + 8], edi
0xca32: mov     eax, r8d
0xca35: ret
0xca36: nop     word ptr cs:[rax + rax]
0xca40: endbr64
0xca44: mov     rax, qword ptr [rip + 0x1e9bd]
0xca4b: cmp     qword ptr [rax + 0x48], 0
0xca50: sete    al
0xca53: movzx   eax, al
0xca56: ret
0xca57: nop     word ptr [rax + rax]
0xca60: endbr64
0xca64: mov     rax, qword ptr [rip + 0x1e99d]
0xca6b: mov     rdx, qword ptr [rax + 0x18]
0xca6f: mov     qword ptr [rax + 0x20], rdx
0xca73: ret
0xca74: nop     word ptr cs:[rax + rax]
0xca7f: nop
0xca80: endbr64
0xca84: sub     rsp, 8
0xca88: xor     eax, eax
0xca8a: mov     edi, 3

```

```

0xca8f: call    0xc6e0
0xca94: mov     rax, qword ptr [rip + 0x1e96d]
0xca9b: mov     dword ptr [rax + 0x10], 0x18
0xcaa2: add     rsp, 8
0xcaa6: ret
0xcaa7: nop     word ptr [rax + rax]
0xcab0: endbr64
0xcab4: sub     rsp, 8
0xcab8: xor     eax, eax
0xcaba: mov     edi, 1
0xcabf: call    0xc6e0
0xcac4: mov     rax, qword ptr [rip + 0x1e93d]
0xcacb: mov     dword ptr [rax + 0x10], 0x19
0xcad2: add     rsp, 8
0xcad6: ret
0xcad7: nop     word ptr [rax + rax]
0xcae0: endbr64
0xcae4: mov     rax, qword ptr [rip + 0x1e91d]
0xcaeb: mov     rdx, qword ptr [rax + 0x18]
0xcaef: mov     rcx, qword ptr [rax + 0x20]
0xcaf3: movsxd  rdx, edx
0xcaf6: mov     qword ptr [rax + 0x18], rcx
0xcafa: mov     qword ptr [rax + 0x20], rdx
0xcafe: ret
0xcaff: nop
0xcb00: endbr64
0xcb04: mov     rax, qword ptr [rip + 0x1e8fd]
0xcb0b: mov     rdx, qword ptr [rax + 0x18]
0xcb0f: cmp     rdx, qword ptr [rax + 0x40]
0xcb13: jbe     0xcb24
0xcb15: sub     rdx, 1
0xcb19: mov     dword ptr [rax + 0x10], 1
0xcb20: mov     qword ptr [rax + 0x18], rdx
0xcb24: ret
0xcb25: nop     word ptr cs:[rax + rax]
0xcb30: endbr64
0xcb34: mov     rax, qword ptr [rip + 0x1e8cd]
0xcb3b: mov     rdx, qword ptr [rax + 0x18]
0xcb3f: cmp     rdx, qword ptr [rax + 0x38]
0xcb43: jae     0xcb54
0xcb45: add     rdx, 1
0xcb49: mov     dword ptr [rax + 0x10], 2
0xcb50: mov     qword ptr [rax + 0x18], rdx
0xcb54: ret
0xcb55: nop     word ptr cs:[rax + rax]
0xcb60: endbr64
0xcb64: mov     rdx, qword ptr [rip + 0x1e89d]
0xcb6b: mov     rax, qword ptr [rdx + 0x18]

```

```

0xcb6f: mov     rdi, qword ptr [rdx + 0x40]
0xcb73: cmp     rax, rdi
0xcb76: jbe     0xcc20
0xcb7c: mov     r8, qword ptr [rdx + 0x58]
0xcb80: jmp     0xcba9
0xcba9:
0xcb82: nop     word ptr [rax + rax]
0xcb88: lea     ecx, [rsi - 0x30]
0xcb8b: cmp     cl, 9
0xcb8e: jbe     0xcbbf
0xcb90: sub     esi, 0x24
0xcb93: cmp     sil, 1
0xcb97: jbe     0xcbbf
0xcb99: mov     qword ptr [rdx + 0x18], rax
0xcb9d: mov     dword ptr [rdx + 0x10], 1
0xcba4: cmp     rdi, rax
0xcba7: jae     0xcbef
0xcba9: sub     rax, 1
0xcbad: movzx   esi, byte ptr [r8 + rax]
0xcbb2: mov     ecx, esi
0xcbb4: and     ecx, 0xffffffff
0xcbb7: sub     ecx, 0x41
0xcbb8: cmp     cl, 0x19
0xcbbd: ja      0xcb88
0xcbbf: mov     rax, qword ptr [rdx + 0x18]
0xcbc3: cmp     rdi, rax
0xcbc6: jae     0xcbef
0xcbc8: sub     rax, 1
0xcbcc: movzx   esi, byte ptr [r8 + rax]
0xcbd1: mov     ecx, esi
0xcbd3: and     ecx, 0xffffffff
0xcbd6: sub     ecx, 0x41
0xcbd9: cmp     cl, 0x19
0xcbdc: jbe     0xcc00
0xcbde: lea     ecx, [rsi - 0x30]
0xcbe1: cmp     cl, 9
0xcbe4: jbe     0xcc00
0xcbe6: sub     esi, 0x24
0xcbe9: cmp     sil, 1
0xcbed: jbe     0xcc00
0xcbef: mov     dword ptr [rdx + 0x10], 3
0xcbf6: ret
0xcbf7: nop     word ptr [rax + rax]
0xcc00: mov     qword ptr [rdx + 0x18], rax
0xcc04: mov     dword ptr [rdx + 0x10], 1
0xcc0b: cmp     rdi, rax
0xcc0e: jb      0xcbc8
0xcc10: mov     dword ptr [rdx + 0x10], 3

```

```

0xcc17: ret
0xcc18: nop      dword ptr [rax + rax]
0xcc20: ret
0xcc21: nop      word ptr cs:[rax + rax]
0xcc2c: nop      dword ptr [rax]
0xcc30: endbr64
0xcc34: mov      rdx, qword ptr [rip + 0x1e7cd]
0xcc3b: mov      rax, qword ptr [rdx + 0x18]
0xcc3f: mov      rdi, qword ptr [rdx + 0x38]
0xcc43: cmp      rax, rdi
0xcc46: jae      0xcce8
0xcc4c: mov      r8, qword ptr [rdx + 0x58]
0xcc50: jmp      0xcc7d
0xcc52: nop      word ptr [rax + rax]
0xcc58: lea      ecx, [rsi - 0x30]
0xcc5b: cmp      cl, 9
0xcc5e: jbe      0xcc8f
0xcc60: sub      esi, 0x24
0xcc63: cmp      sil, 1
0xcc67: jbe      0xcc8f
0xcc69: add      rax, 1
0xcc6d: mov      dword ptr [rdx + 0x10], 2
0xcc74: mov      qword ptr [rdx + 0x18], rax
0xcc78: cmp      rdi, rax
0xcc7b: je       0xccbb
0xcc7d: movzx    esi, byte ptr [r8 + rax]
0xcc82: mov      ecx, esi
0xcc84: and      ecx, 0xffffffff
0xcc87: sub      ecx, 0x41
0xcc8a: cmp      cl, 0x19
0xcc8d: ja       0xcc58
0xcc8f: mov      rax, qword ptr [rdx + 0x18]
0xcc93: cmp      rdi, rax
0xcc96: jbe      0xccbb
0xcc98: movzx    esi, byte ptr [r8 + rax]
0xcc9d: mov      ecx, esi
0xcc9f: and      ecx, 0xffffffff
0xcca2: sub      ecx, 0x41
0xcca5: cmp      cl, 0x19
0xcca8: jbe      0xcc88
0xccaa: lea      ecx, [rsi - 0x30]
0ccad: cmp      cl, 9
0xccb0: jbe      0xcc88
0xccb2: sub      esi, 0x24
0xccb5: cmp      sil, 1
0xccb9: jbe      0xcc88
0xccbb: mov      dword ptr [rdx + 0x10], 4

```

```

0xccc2: ret
0xccc3: nop      dword ptr [rax + rax]
0xccc8: add      rax, 1
0xcccc: mov      dword ptr [rdx + 0x10], 2
0xccd3: mov      qword ptr [rdx + 0x18], rax
0xccd7: cmp      rdi, rax
0xccda: jne      0xcc98
0xccdc: mov      dword ptr [rdx + 0x10], 4
0xcce3: ret
0xcce4: nop      dword ptr [rax]
0xcce8: ret
0xcce9: nop      dword ptr [rax]
0xccf0: endbr64
0xccf4: mov      rax, qword ptr [rip + 0x1e70d]
0xccfb: mov      rdx, qword ptr [rax + 0x40]
0xccff: mov      dword ptr [rax + 0x10], 5
0xcd06: mov      qword ptr [rax + 0x18], rdx
0xcd0a: ret
0xcd0b: nop      dword ptr [rax + rax]
0xcd10: endbr64
0xcd14: mov      rax, qword ptr [rip + 0x1e6ed]
0xcd1b: mov      rdx, qword ptr [rax + 0x38]
0xcd1f: mov      dword ptr [rax + 0x10], 6
0xcd26: mov      qword ptr [rax + 0x18], rdx
0xcd2a: ret
0xcd2b: nop      dword ptr [rax + rax]
0xcd30: endbr64
0xcd34: push     rbp
0xcd35: push     rbx
0xcd36: mov      ebx, edi
0xcd38: sub      rsp, 8
0xcd3c: call     0x4f00
0xcd41: movsxd   rdx, ebx
0xcd44: mov      rax, qword ptr [rax]
0xcd47: test     byte ptr [rax + rdx*2 + 1], 0x40
0xcd4c: je       0xcdab
0xcd4e: mov      rbp, qword ptr [rip + 0x1e6b3]
0xcd55: mov      rsi, qword ptr [rbp + 0x38]
0xcd59: lea      rdx, [rsi + 1]
0xcd5d: cmp      rdx, qword ptr [rbp + 0x50]
0xcd61: jae      0xcdb8
0xcd63: mov      rax, qword ptr [rbp + 0x18]
0xcd67: mov      rcx, qword ptr [rbp + 0x58]
0xcd6b: sub      rdx, rax
0xcd6e: lea      rsi, [rcx + rax]
0xcd72: lea      rdi, [rcx + rax + 1]
0xcd77: call     0x4d40
0xcd7c: mov      rax, qword ptr [rip + 0x1e685]

```

```

0xcd83: mov     rdx, qword ptr [rax + 0x58]
0xcd87: mov     rax, qword ptr [rax + 0x18]
0xcd8b: mov     byte ptr [rdx + rax], bl
0xcd8e: mov     rax, qword ptr [rip + 0x1e673]
0xcd95: add     qword ptr [rax + 0x18], 1
0xcd9a: add     qword ptr [rax + 0x38], 1
0xcd9f: add     qword ptr [rax + 0x48], 1
0xcda4: mov     dword ptr [rax + 0x10], 7
0xcdab: add     rsp, 8
0xcdaf: pop     rbx
0xcdb0: pop     rbp
0xcdb1: ret
0xcdb2: nop     word ptr [rax + rax]
0xcdb8: add     rsi, 0x21
0xcdbc: mov     rdi, qword ptr [rbp + 0x58]
0xcdc0: mov     qword ptr [rbp + 0x50], rsi
0xcdc4: call    0x1d690
0xcdc9: mov     qword ptr [rbp + 0x58], rax
0xcdcd: mov     rbp, qword ptr [rip + 0x1e634]
0xcdd4: mov     rax, qword ptr [rbp + 0x38]
0xcdd8: lea     rdx, [rax + 1]
0xcddc: jmp     0xcd63
0xcd63:
0xcdde: nop
0xcde0: endbr64
0xcde4: mov     rax, qword ptr [rip + 0x1e61d]
0xcdeb: mov     rcx, qword ptr [rax + 0x18]
0xcdef: mov     rdx, qword ptr [rax + 0x38]
0xcdf3: cmp     rcx, rdx
0xcdf6: jnb     0xce00
0xcdf8: ret
0xcdf9: nop     dword ptr [rax]
0xce00: push    rbx
0xce01: mov     rax, qword ptr [rax + 0x58]
0xce05: add     rdx, 1
0xce09: sub     rdx, rcx
0xce0c: lea     rsi, [rax + rcx + 1]
0xce11: lea     rdi, [rax + rcx]
0xce15: call    0x4bb0
0xce1a: mov     rbx, qword ptr [rip + 0x1e5e7]
0xce21: mov     rsi, qword ptr [rbx + 0x38]
0xce25: sub     qword ptr [rbx + 0x48], 1
0xce2a: lea     rax, [rsi - 1]
0xce2e: mov     qword ptr [rbx + 0x38], rax
0xce32: test    al, 0xf
0xce34: je      0xce40
0xce36: mov     dword ptr [rbx + 0x10], 8
0xce3d: pop     rbx

```



```

0xce3e: ret
0xce3f: nop
0xce40: mov     qword ptr [rbx + 0x50], rsi
0xce44: mov     rdi, qword ptr [rbx + 0x58]
0xce48: call    0x1d690
0xce4d: mov     qword ptr [rbx + 0x58], rax
0xce51: mov     rbx, qword ptr [rip + 0x1e5b0]
0xce58: mov     dword ptr [rbx + 0x10], 8
0xce5f: pop     rbx
0xce60: ret
0xce61: nop     word ptr cs:[rax + rax]
0xce6c: nop     dword ptr [rax]
0xce70: endbr64
0xce74: mov     rax, qword ptr [rip + 0x1e58d]
0xce7b: mov     rcx, qword ptr [rax + 0x18]
0xce7f: cmp     rcx, qword ptr [rax + 0x40]
0xce83: ja      0xce90
0xce85: ret
0xce86: nop     word ptr cs:[rax + rax]
0xce90: push    rbx
0xce91: mov     rdx, qword ptr [rax + 0x38]
0xce95: mov     rsi, qword ptr [rax + 0x58]
0xce99: add     rdx, 1
0xce9d: lea     rdi, [rsi + rcx - 1]
0xcea2: sub     rdx, rcx
0xcea5: add     rsi, rcx
0xcea8: call    0x4bb0
0xcead: mov     rbx, qword ptr [rip + 0x1e554]
0xceb4: mov     rsi, qword ptr [rbx + 0x38]
0xceb8: sub     qword ptr [rbx + 0x18], 1
0xcebd: sub     qword ptr [rbx + 0x48], 1
0xcec2: lea     rax, [rsi - 1]
0xcec6: mov     qword ptr [rbx + 0x38], rax
0xceca: test    al, 0xf
0xcecc: je      0xcee0
0xcece: mov     dword ptr [rbx + 0x10], 9
0xcd5: pop     rbx
0xcd6: ret
0xcd7: nop     word ptr [rax + rax]
0xcee0: mov     qword ptr [rbx + 0x50], rsi
0xcee4: mov     rdi, qword ptr [rbx + 0x58]
0xcee8: call    0x1d690
0xceed: mov     qword ptr [rbx + 0x58], rax
0xcef1: mov     rbx, qword ptr [rip + 0x1e510]
0xcef8: mov     dword ptr [rbx + 0x10], 9
0xceff: pop     rbx
0xcf00: ret
0xcf01: nop     word ptr cs:[rax + rax]

```

```

0xcf0c: nop      dword ptr [rax]
0xcf10: endbr64
0xcf14: push     rbp
0xcf15: push     rbx
0xcf16: sub      rsp, 8
0xcf1a: mov      r9, qword ptr [rip + 0x1e4e7]
0xcf21: mov      rax, qword ptr [r9 + 0x18]
0xcf25: mov      rbx, qword ptr [r9 + 0x20]
0xcf29: mov      qword ptr [r9 + 0x20], rax
0xcf2d: xor      eax, eax
0xcf2f: call     0xcc30
0xcf34: xor      eax, eax
0xcf36: mov      edi, 3
0xcf3b: mov      rbp, qword ptr [r9 + 0x18]
0xcf3f: call     0xc6e0
0xcf44: mov      rax, qword ptr [rip + 0x1e4bd]
0xcf4b: mov      rdx, qword ptr [rax + 0x18]
0xcf4f: cmp      rdx, rbx
0xcf52: jae      0xcf65
0xcf54: lea      rcx, [rdx + rbx]
0xcf58: sub      rcx, rbp
0xcf5b: cmp      rbx, rbp
0xcf5e: cmovae   rdx, rcx
0xcf62: mov      rbx, rdx
0xcf65: mov      qword ptr [rax + 0x20], rbx
0xcf69: mov      dword ptr [rax + 0x10], 0xa
0xcf70: add      rsp, 8
0xcf74: pop      rbx
0xcf75: pop      rbp
0xcf76: ret
0xcf77: nop      word ptr [rax + rax]
0xcf80: endbr64
0xcf84: push     rbp
0xcf85: push     rbx
0xcf86: sub      rsp, 8
0xcf8a: mov      rax, qword ptr [rip + 0x1e477]
0xcf91: mov      rbp, qword ptr [rax + 0x18]
0xcf95: mov      rbx, qword ptr [rax + 0x20]
0xcf99: mov      qword ptr [rax + 0x20], rbp
0xcf9d: xor      eax, eax
0xcf9f: call     0xcb60
0xcfa4: xor      eax, eax
0xcfa6: mov      edi, 3
0xcfab: call     0xc6e0
0xcfb0: mov      rax, qword ptr [rip + 0x1e451]
0xcfb7: mov      rdx, qword ptr [rax + 0x18]
0xcfb7: cmp      rdx, rbx
0xcfbe: jae      0xcfd1

```

```

0xcfc0: lea      rcx, [rdx + rbx]
0xcfc4: sub      rcx, rbp
0xcfc7: cmp      rbx, rbp
0xcfc9: cmovae   rdx, rcx
0xcfce: mov      rbx, rdx
0xcfd1: mov      qword ptr [rax + 0x20], rbx
0xcfd5: mov      dword ptr [rax + 0x10], 0xb
0xcfdc: add      rsp, 8
0xcfe0: pop      rbx
0xcfe1: pop      rbp
0xcfe2: ret
0xcfe3: nop      word ptr cs:[rax + rax]
0xcfee: nop
0xcff0: endbr64
0xcff4: push     rbx
0xcff5: mov      rbx, qword ptr [rip + 0x1e40c]
0xcffc: mov      esi, 1
0xd001: mov      qword ptr [rbx + 0x18], 0
0xd009: mov      rdi, qword ptr [rbx + 0x58]
0xd00d: mov      qword ptr [rbx + 0x20], 0
0xd015: mov      qword ptr [rbx + 0x38], 0
0xd01d: mov      qword ptr [rbx + 0x40], 0
0xd025: mov      qword ptr [rbx + 0x48], 0
0xd02d: mov      qword ptr [rbx + 0x50], 1
0xd035: call     0x1d690
0xd03a: mov      qword ptr [rbx + 0x58], rax
0xd03e: mov      rax, qword ptr [rip + 0x1e3c3]
0xd045: mov      rax, qword ptr [rax + 0x58]
0xd049: mov      byte ptr [rax], 0
0xd04c: mov      rax, qword ptr [rip + 0x1e3b5]
0xd053: mov      dword ptr [rax + 0x10], 0xc
0xd05a: pop      rbx
0xd05b: ret
0xd05c: nop      dword ptr [rax]
0xd060: endbr64
0xd064: push     rbx
0xd065: mov      edi, 0x70
0xd06a: sub      rsp, 0x10
0xd06e: mov      rax, qword ptr fs:[0x28]
0xd077: mov      qword ptr [rsp + 8], rax
0xd07c: xor      eax, eax
0xd07e: call     0x1d670
0xd083: mov      qword ptr [rax + 8], 1
0xd08b: mov      qword ptr [rax + 0x58], 0
0xd093: mov      qword ptr [rax + 0x60], 0
0xd09b: mov      qword ptr [rax + 0x28], 0
0xd0a3: mov      qword ptr [rax + 0x30], 0
0xd0ab: mov      qword ptr [rip + 0x1e356], rax

```

```

0xd0b2: xor     eax, eax
0xd0b4: call    0xcff0
0xd0b9: mov     rbx, qword ptr [rip + 0x1e348]
0xd0c0: xor     eax, eax
0xd0c2: call    0xc590
0xd0c7: lea     rdi, [rip + 0x135e4]
0xd0ce: mov     qword ptr [rbx], rax
0xd0d1: call    0x17c50
0xd0d6: mov     ecx, 1
0xd0db: mov     rsi, rsp
0xd0de: mov     edx, 1
0xd0e3: lea     rdi, [rip + 0x13e42]
0xd0ea: call    0x17320
0xd0ef: mov     rdi, qword ptr [rsp]
0xd0f3: lea     rax, [rip + 0x13e24]
0xd0fa: mov     rbx, qword ptr [rip + 0x1e307]
0xd101: test    rdi, rdi
0xd104: cmov     rdi, rax
0xd108: xor     eax, eax
0xd10a: call    0x19940
0xd10f: lea     rdi, [rip + 0x1cb32]
0xd116: mov     qword ptr [rbx + 0x68], rax
0xd11a: mov     eax, dword ptr [rip + 0x1cb40]
0xd120: test    eax, eax
0xd122: lea     rax, [rip + 0x1cb07]
0xd129: cmov     rdi, rax
0xd12d: call    0x17c50
0xd132: lea     rdi, [rip + 0x1d0c7]
0xd139: mov     edx, 6
0xd13e: lea     rsi, [rip + 0x1d0db]
0xd145: call    0x17f70
0xd14a: xor     eax, eax
0xd14c: call    0x1a310
0xd151: mov     rax, qword ptr [rip + 0x1e2b0]
0xd158: mov     rdi, qword ptr [rax + 0x68]
0xd15c: xor     eax, eax
0xd15e: call    0x1a900
0xd163: nop     dword ptr [rax + rax]
0xd168: xor     eax, eax
0xd16a: call    0x1ab40
0xd16f: test    rax, rax
0xd172: jne     0xd168
0xd174: mov     edi, 0x200
0xd179: call    0x1a620
0xd17e: mov     rax, qword ptr [rsp + 8]
0xd183: sub     rax, qword ptr fs:[0x28]
0xd18c: jne     0xd194
0xd18e: add     rsp, 0x10

```

```

0xd192: pop      rbx
0xd193: ret
0xd194: call     0x49d0
0xd199: nop      dword ptr [rax]
0xd1a0: endbr64
0xd1a4: sub      rsp, 8
0xd1a8: mov      rax, qword ptr [rip + 0x1e259]
0xd1af: or       edi, 2
0xd1b2: mov      rdx, qword ptr [rax + 0x40]
0xd1b6: mov      dword ptr [rax + 0x10], 6
0xd1bd: mov      qword ptr [rax + 0x20], rdx
0xd1c1: mov      rdx, qword ptr [rax + 0x38]
0xd1c5: mov      qword ptr [rax + 0x18], rdx
0xd1c9: xor      eax, eax
0xd1cb: call     0xc6e0
0xd1d0: mov      rax, qword ptr [rip + 0x1e231]
0xd1d7: mov      rdx, qword ptr [rax + 0x18]
0xd1db: mov      dword ptr [rax + 0x10], 0xd
0xd1e2: mov      qword ptr [rax + 0x20], rdx
0xd1e6: add      rsp, 8
0xd1ea: ret
0xd1eb: nop      dword ptr [rax + rax]
0xd1f0: endbr64
0xd1f4: mov      rax, qword ptr [rip + 0x1e20d]
0xd1fb: push     rbx
0xd1fc: mov      edi, 3
0xd201: mov      rdx, qword ptr [rax + 0x18]
0xd205: mov      rsi, qword ptr [rax + 0x20]
0xd209: mov      dword ptr [rax + 0x10], 5
0xd210: mov      rcx, qword ptr [rax + 0x40]
0xd214: mov      qword ptr [rax + 0x20], rdx
0xd218: mov      rbx, rsi
0xd21b: mov      qword ptr [rax + 0x18], rcx
0xd21f: sub      rbx, rdx
0xd222: cmp      rsi, rdx
0xd225: cmovbe   rbx, rcx
0xd229: xor      eax, eax
0xd22b: call     0xc6e0
0xd230: mov      rdx, qword ptr [rip + 0x1e1d1]
0xd237: mov      rax, qword ptr [rdx + 0x38]
0xd23b: mov      dword ptr [rdx + 0x10], 0xe
0xd242: cmp      rax, rbx
0xd245: cmova     rax, rbx
0xd249: mov      qword ptr [rdx + 0x20], rax
0xd24d: pop      rbx
0xd24e: ret
0xd24f: nop
0xd250: endbr64

```

```

0xd254: mov     rax, qword ptr [rip + 0x1e1ad]
0xd25b: push    rbx
0xd25c: mov     edi, 3
0xd261: mov     rdx, qword ptr [rax + 0x18]
0xd265: mov     rbx, qword ptr [rax + 0x20]
0xd269: mov     dword ptr [rax + 0x10], 6
0xd270: mov     qword ptr [rax + 0x20], rdx
0xd274: mov     rdx, qword ptr [rax + 0x38]
0xd278: mov     qword ptr [rax + 0x18], rdx
0xd27c: xor     eax, eax
0xd27e: call    0xc6e0
0xd283: mov     rdx, qword ptr [rip + 0x1e17e]
0xd28a: mov     rax, qword ptr [rdx + 0x38]
0xd28e: mov     dword ptr [rdx + 0x10], 0xf
0xd295: cmp     rax, rbx
0xd298: cmova   rax, rbx
0xd29c: mov     qword ptr [rdx + 0x20], rax
0xd2a0: pop     rbx
0xd2a1: ret
0xd2a2: nop     word ptr cs:[rax + rax]
0xd2ad: nop     dword ptr [rax]
0xd2b0: endbr64
0xd2b4: mov     rax, qword ptr [rip + 0x1e14d]
0xd2bb: mov     rdx, qword ptr [rax + 0x58]
0xd2bf: mov     rax, qword ptr [rax + 0x18]
0xd2c3: cmp     byte ptr [rdx + rax], 0x20
0xd2c7: je      0xd2d0
0xd2c9: ret
0xd2ca: nop     word ptr [rax + rax]
0xd2d0: sub     rsp, 8
0xd2d4: nop     dword ptr [rax]
0xd2d8: xor     eax, eax
0xd2da: call    0xcde0
0xd2df: mov     rax, qword ptr [rip + 0x1e122]
0xd2e6: mov     rdx, qword ptr [rax + 0x58]
0xd2ea: mov     rcx, qword ptr [rax + 0x18]
0xd2ee: cmp     byte ptr [rdx + rcx], 0x20
0xd2f2: je      0xd2d8
0xd2f4: cmp     qword ptr [rax + 0x48], 0
0xd2f9: jne     0xd31d
0xd2fb: jmp     0xd324
0xd324:
0xd2fd: nop     dword ptr [rax]
0xd300: xor     eax, eax
0xd302: call    0xce70
0xd307: mov     rax, qword ptr [rip + 0x1e0fa]
0xd30e: cmp     qword ptr [rax + 0x48], 0
0xd313: je      0xd324

```

```

0xd315: mov     rdx, qword ptr [rax + 0x58]
0xd319: mov     rcx, qword ptr [rax + 0x18]
0xd31d: cmp     byte ptr [rdx + rcx - 1], 0x20
0xd322: je      0xd300
0xd324: mov     dword ptr [rax + 0x10], 0x11
0xd32b: add     rsp, 8
0xd32f: ret
0xd330: endbr64
0xd334: mov     rax, qword ptr [rip + 0x1e0cd]
0xd33b: mov     rdx, qword ptr [rax + 0x58]
0xd33f: mov     rax, qword ptr [rax + 0x18]
0xd343: cmp     byte ptr [rdx + rax], 0x20
0xd347: je      0xd350
0xd349: ret
0xd34a: nop     word ptr [rax + rax]
0xd350: sub     rsp, 8
0xd354: xor     eax, eax
0xd356: call    0xd2b0
0xd35b: mov     edi, 0x20
0xd360: call    0xcd30
0xd365: mov     rax, qword ptr [rip + 0x1e09c]
0xd36c: mov     dword ptr [rax + 0x10], 0x10
0xd373: add     rsp, 8
0xd377: ret
0xd378: nop     dword ptr [rax + rax]
0xd380: endbr64
0xd384: push    rbp
0xd385: push    rbx
0xd386: sub     rsp, 8
0xd38a: mov     rbp, qword ptr [rip + 0x1e077]
0xd391: mov     rbx, qword ptr [rbp + 0x18]
0xd395: cmp     rbx, qword ptr [rbp + 0x38]
0xd399: jb      0xd3a8
0xd39b: add     rsp, 8
0xd39f: pop     rbx
0xd3a0: pop     rbp
0xd3a1: ret
0xd3a2: nop     word ptr [rax + rax]
0xd3a8: xor     eax, eax
0xd3aa: call    0xcc30
0xd3af: cmp     rbx, qword ptr [rbp + 0x18]
0xd3b3: jae     0xd3e4
0xd3b5: call    0x4ef0
0xd3ba: nop     word ptr [rax + rax]
0xd3c0: mov     rdx, qword ptr [rbp + 0x58]
0xd3c4: mov     rcx, qword ptr [rax]
0xd3c7: add     rdx, rbx
0xd3ca: add     rbx, 1

```

```

0xd3ce: movsx    rsi, byte ptr [rdx]
0xd3d2: mov     ecx, dword ptr [rcx + rsi*4]
0xd3d5: mov     byte ptr [rdx], cl
0xd3d7: mov     rbp, qword ptr [rip + 0x1e02a]
0xd3de: cmp     qword ptr [rbp + 0x18], rbx
0xd3e2: ja      0xd3c0
0xd3e4: mov     dword ptr [rbp + 0x10], 0x12
0xd3eb: add     rsp, 8
0xd3ef: pop     rbx
0xd3f0: pop     rbp
0xd3f1: ret
0xd3f2: nop     word ptr cs:[rax + rax]
0xd3fd: nop     dword ptr [rax]
0xd400: endbr64
0xd404: push    rbp
0xd405: push    rbx
0xd406: sub     rsp, 8
0xd40a: mov     rbp, qword ptr [rip + 0x1dff7]
0xd411: mov     rbx, qword ptr [rbp + 0x18]
0xd415: cmp     rbx, qword ptr [rbp + 0x38]
0xd419: jb      0xd428
0xd41b: add     rsp, 8
0xd41f: pop     rbx
0xd420: pop     rbp
0xd421: ret
0xd422: nop     word ptr [rax + rax]
0xd428: xor     eax, eax
0xd42a: call    0xcc30
0xd42f: cmp     rbx, qword ptr [rbp + 0x18]
0xd433: jae     0xd464
0xd435: call    0x47c0
0xd43a: nop     word ptr [rax + rax]
0xd440: mov     rdx, qword ptr [rbp + 0x58]
0xd444: mov     rcx, qword ptr [rax]
0xd447: add     rdx, rbx
0xd44a: add     rbx, 1
0xd44e: movsx    rsi, byte ptr [rdx]
0xd452: mov     ecx, dword ptr [rcx + rsi*4]
0xd455: mov     byte ptr [rdx], cl
0xd457: mov     rbp, qword ptr [rip + 0x1dfaa]
0xd45e: cmp     qword ptr [rbp + 0x18], rbx
0xd462: ja      0xd440
0xd464: mov     dword ptr [rbp + 0x10], 0x13
0xd46b: add     rsp, 8
0xd46f: pop     rbx
0xd470: pop     rbp
0xd471: ret
0xd472: nop     word ptr cs:[rax + rax]

```



```

0xd47d: nop      dword ptr [rax]
0xd480: endbr64
0xd484: push     r14
0xd486: push     r13
0xd488: push     r12
0xd48a: push     rbp
0xd48b: mov      rbp, qword ptr [rip + 0x1df76]
0xd492: push     rbx
0xd493: mov      rbx, qword ptr [rbp + 0x18]
0xd497: cmp      rbx, qword ptr [rbp + 0x38]
0xd49b: jb       0xd4b0
0xd49d: pop      rbx
0xd49e: pop      rbp
0xd49f: pop      r12
0xd4a1: pop      r13
0xd4a3: pop      r14
0xd4a5: ret
0xd4a6: nop      word ptr cs:[rax + rax]
0xd4b0: xor      eax, eax
0xd4b2: call     0xcc30
0xd4b7: cmp      rbx, qword ptr [rbp + 0x18]
0xd4bb: jae      0xd528
0xd4bd: call     0x4f00
0xd4c2: mov      r14d, 1
0xd4c8: mov      r13, rax
0xd4cb: jmp      0xd4f3
0xd4f3:
0xd4cd: nop      dword ptr [rax]
0xd4d0: call     0x47c0
0xd4d5: xor      r14d, r14d
0xd4d8: mov      rax, qword ptr [rax]
0xd4db: mov      eax, dword ptr [rax + r12*4]
0xd4df: mov      byte ptr [rbp], al
0xd4e2: mov      rbp, qword ptr [rip + 0x1df1f]
0xd4e9: add      rbx, 1
0xd4ed: cmp      qword ptr [rbp + 0x18], rbx
0xd4f1: jbe      0xd528
0xd4f3: mov      rax, qword ptr [rbp + 0x58]
0xd4f7: add      rax, rbx
0xd4fa: movsx    r12, byte ptr [rax]
0xd4fe: mov      rbp, rax
0xd501: mov      rax, qword ptr [r13]
0xd505: test     byte ptr [rax + r12*2], 8
0xd50a: je       0xd4e2
0xd50c: test     r14d, r14d
0xd50f: jne      0xd4d0
0xd511: call     0x4ef0
0xd516: mov      rax, qword ptr [rax]

```

```

0xd519: mov     eax, dword ptr [rax + r12*4]
0xd51d: mov     byte ptr [rbp], al
0xd520: jmp     0xd4e2
0xd4e2:
0xd522: nop     word ptr [rax + rax]
0xd528: mov     dword ptr [rbp + 0x10], 0x14
0xd52f: pop     rbx
0xd530: pop     rbp
0xd531: pop     r12
0xd533: pop     r13
0xd535: pop     r14
0xd537: ret
0xd538: nop     dword ptr [rax + rax]
0xd540: endbr64
0xd544: push    r12
0xd546: mov     r12, rdi
0xd549: push    rbp
0xd54a: push    rbx
0xd54b: call    0x49b0
0xd550: mov     rbp, qword ptr [rip + 0x1deb1]
0xd557: mov     rbx, rax
0xd55a: mov     rsi, qword ptr [rbp + 0x40]
0xd55e: sub     rax, rsi
0xd561: add     qword ptr [rbp + 0x18], rax
0xd565: add     qword ptr [rbp + 0x20], rax
0xd569: mov     rax, qword ptr [rbp + 0x48]
0xd56d: lea     rdx, [rax + 1]
0xd571: lea     r8, [rbx + rdx]
0xd575: cmp     r8, qword ptr [rbp + 0x50]
0xd579: ja      0xd5d8
0xd57b: mov     rdi, qword ptr [rbp + 0x58]
0xd57f: add     rsi, rdi
0xd582: add     rdi, rbx
0xd585: call    0x4d40
0xd58a: mov     rax, qword ptr [rip + 0x1de77]
0xd591: mov     rdx, rbx
0xd594: mov     rsi, r12
0xd597: mov     rdi, qword ptr [rax + 0x58]
0xd59b: call    0x4bb0
0xd5a0: mov     rax, qword ptr [rip + 0x1de61]
0xd5a7: mov     rsi, rbx
0xd5aa: mov     rdi, qword ptr [rax + 0x58]
0xd5ae: call    0x184d0
0xd5b3: mov     rax, qword ptr [rip + 0x1de4e]
0xd5ba: mov     qword ptr [rax + 0x40], rbx
0xd5be: add     rbx, qword ptr [rax + 0x48]
0xd5c2: mov     qword ptr [rax + 0x38], rbx
0xd5c6: mov     dword ptr [rax + 0x10], 0x15

```

```

0xd5cd: pop    rbx
0xd5ce: pop    rbp
0xd5cf: pop    r12
0xd5d1: ret
0xd5d2: nop     word ptr [rax + rax]
0xd5d8: mov     qword ptr [rbp + 0x50], r8
0xd5dc: mov     rdi, qword ptr [rbp + 0x58]
0xd5e0: mov     rsi, r8
0xd5e3: call    0xd690
0xd5e8: mov     qword ptr [rbp + 0x58], rax
0xd5ec: mov     rbp, qword ptr [rip + 0x1de15]
0xd5f3: mov     rax, qword ptr [rbp + 0x48]
0xd5f7: mov     rsi, qword ptr [rbp + 0x40]
0xd5fb: lea     rdx, [rax + 1]
0xd5ff: jmp     0xd57b
0xd57b:
0xd604: nop     word ptr cs:[rax + rax]
0xd60f: nop
0xd610: endbr64
0xd614: test    rdi, rdi
0xd617: je      0xd6f0
0xd61d: push    r12
0xd61f: push    rbp
0xd620: mov     rbp, rdi
0xd623: push    rbx
0xd624: call    0x49b0
0xd629: mov     r12, qword ptr [rip + 0x1ddd8]
0xd630: mov     rbx, rax
0xd633: mov     rax, qword ptr [r12 + 0x38]
0xd638: lea     rdx, [rax + 1]
0xd63c: mov     rax, qword ptr [r12 + 0x50]
0xd641: lea     rcx, [rbx + rdx]
0xd645: cmp     rcx, rax
0xd648: ja      0xd6c0
0xd64a: mov     rax, qword ptr [r12 + 0x18]
0xd64f: mov     rdi, qword ptr [r12 + 0x58]
0xd654: sub     rdx, rax
0xd657: lea     rsi, [rdi + rax]
0xd65b: add     rax, rbx
0xd65e: add     rdi, rax
0xd661: call    0x4d40
0xd666: mov     rax, qword ptr [rip + 0x1dd9b]
0xd66d: mov     rdx, rbx
0xd670: mov     rsi, rbp
0xd673: mov     rdi, qword ptr [rax + 0x18]
0xd677: add     rdi, qword ptr [rax + 0x58]
0xd67b: call    0x4bb0
0xd680: mov     rax, qword ptr [rip + 0x1dd81]

```

```

0xd687: mov     rsi, rbx
0xd68a: mov     rdi, qword ptr [rax + 0x18]
0xd68e: add     rdi, qword ptr [rax + 0x58]
0xd692: call    0x184d0
0xd697: mov     rax, qword ptr [rip + 0x1dd6a]
0xd69e: add     qword ptr [rax + 0x18], rbx
0xd6a2: add     qword ptr [rax + 0x38], rbx
0xd6a6: add     qword ptr [rax + 0x48], rbx
0xd6aa: mov     dword ptr [rax + 0x10], 0x16
0xd6b1: pop     rbx
0xd6b2: pop     rbp
0xd6b3: pop     r12
0xd6b5: ret
0xd6b6: nop     word ptr cs:[rax + rax]
0xd6c0: lea     rsi, [rax + rbx + 0x21]
0xd6c5: mov     rdi, qword ptr [r12 + 0x58]
0xd6ca: mov     qword ptr [r12 + 0x50], rsi
0xd6cf: call    0x1d690
0xd6d4: mov     qword ptr [r12 + 0x58], rax
0xd6d9: mov     r12, qword ptr [rip + 0x1dd28]
0xd6e0: mov     rax, qword ptr [r12 + 0x38]
0xd6e5: lea     rdx, [rax + 1]
0xd6e9: jmp     0xd64a
0xd64a:
0xd6ee: nop
0xd6f0: ret
0xd6f1: nop     word ptr cs:[rax + rax]
0xd6fc: nop     dword ptr [rax]
0xd700: endbr64
0xd704: mov     rax, qword ptr [rip + 0x1dcfd]
0xd70b: mov     rdi, qword ptr [rax + 0x60]
0xd70f: test    rdi, rdi
0xd712: je      0xd730
0xd714: sub     rsp, 8
0xd718: call    0xd610
0xd71d: mov     rax, qword ptr [rip + 0x1dce4]
0xd724: mov     dword ptr [rax + 0x10], 0x1a
0xd72b: add     rsp, 8
0xd72f: ret
0xd730: ret
0xd731: nop     word ptr cs:[rax + rax]
0xd73c: nop     dword ptr [rax]
0xd740: endbr64
0xd744: mov     rax, qword ptr [rip + 0x1dcbd]
0xd74b: push    r12
0xd74d: push    rbp
0xd74e: push    rbx
0xd74f: mov     rbx, qword ptr [rax + 0x40]

```

```

0xd753: mov     r12, qword ptr [rax + 0x28]
0xd757: mov     eax, dword ptr [rax + 0xc]
0xd75a: lea     rbp, [rbx + 3]
0xd75e: test    eax, eax
0xd760: je      0xd7f0
0xd766: mov     edx, dword ptr [rip + 0x1caa4]
0xd76c: mov     esi, dword ptr [rip + 0x1ca9a]
0xd772: mov     edi, dword ptr [rip + 0x1ca9c]
0xd778: call    0xa570
0xd77d: mov     rcx, qword ptr [rip + 0x1dc84]
0xd784: cmp     r12, rbp
0xd787: jb      0xd810
0xd78d: mov     rsi, qword ptr [rcx + 0x18]
0xd791: mov     rdx, qword ptr [rcx + 0x40]
0xd795: xor     ebx, ebx
0xd797: mov     rdi, qword ptr [rcx + 0x28]
0xd79b: mov     rax, rdi
0xd79e: sub     rax, rdx
0xd7a1: mov     edx, 1
0xd7a6: mov     r8, rax
0xd7a9: shr     r8, 2
0xd7ad: cmp     rax, 3
0xd7b1: cmova   edx, r8d
0xd7b5: xor     eax, eax
0xd7b7: cmp     rdi, rsi
0xd7ba: ja      0xd7d3
0xd7bc: mov     rax, rsi
0xd7bf: sub     rax, rdi
0xd7c2: movsxd  rdi, edx
0xd7c5: lea     r8d, [rdx + rax]
0xd7c9: xor     edx, edx
0xd7cb: div     rdi
0xd7ce: mov     eax, r8d
0xd7d1: sub     eax, edx
0xd7d3: mov     edx, esi
0xd7d5: mov     rdi, qword ptr [rcx]
0xd7d8: xor     esi, esi
0xd7da: sub     edx, eax
0xd7dc: call    0xc680
0xd7e1: cmp     r12, rbp
0xd7e4: jb      0xd840
0xd7e6: pop     rbx
0xd7e7: pop     rbp
0xd7e8: pop     r12
0xd7ea: ret
0xd7eb: nop     dword ptr [rax + rax]
0xd7f0: mov     edx, dword ptr [rip + 0x1ca0e]
0xd7f6: mov     esi, dword ptr [rip + 0x1ca04]

```

```

0xd7fc: mov     edi, dword ptr [rip + 0x1ca06]
0xd802: jmp     0xd778
0xd778:
0xd807: nop     word ptr [rax + rax]
0xd810: mov     rbx, qword ptr [rcx + 0x40]
0xd814: mov     rsi, qword ptr [rcx + 0x18]
0xd818: mov     qword ptr [rcx + 0x40], 0
0xd820: xor     edx, edx
0xd822: add     qword ptr [rcx + 0x58], rbx
0xd826: sub     rsi, rbx
0xd829: sub     qword ptr [rcx + 0x38], rbx
0xd82d: sub     qword ptr [rcx + 0x20], rbx
0xd831: movsxd  rbx, ebx
0xd834: mov     qword ptr [rcx + 0x18], rsi
0xd838: jmp     0xd797
0xd797:
0xd83d: nop     dword ptr [rax]
0xd840: mov     rax, qword ptr [rip + 0x1dbc1]
0xd847: movsxd  rbx, ebx
0xd84a: mov     qword ptr [rax + 0x40], rbx
0xd84e: sub     qword ptr [rax + 0x58], rbx
0xd852: add     qword ptr [rax + 0x38], rbx
0xd856: add     qword ptr [rax + 0x18], rbx
0xd85a: add     qword ptr [rax + 0x20], rbx
0xd85e: pop     rbx
0xd85f: pop     rbp
0xd860: pop     r12
0xd862: ret
0xd863: nop     word ptr cs:[rax + rax]
0xd86e: nop
0xd870: endbr64
0xd874: push    r15
0xd876: push    r14
0xd878: push    r13
0xd87a: push    r12
0xd87c: push    rbp
0xd87d: push    rbx
0xd87e: sub     rsp, 0x28
0xd882: mov     rax, qword ptr fs:[0x28]
0xd88b: mov     qword ptr [rsp + 0x18], rax
0xd890: mov     rax, qword ptr [rip + 0x1db71]
0xd897: lea     r14, [rsp + 0x17]
0xd89c: mov     rdi, r14
0xd89f: mov     rbx, qword ptr [rax + 0x40]
0xd8a3: mov     r13, qword ptr [rax + 0x28]
0xd8a7: call    0xa6a0
0xd8ac: mov     rax, qword ptr [rip + 0x1db55]
0xd8b3: add     rbx, 3

```

```

0xd8b7: mov     ecx, dword ptr [rax + 0xc]
0xd8ba: test    ecx, ecx
0xd8bc: je      0xda20
0xd8c2: mov     edx, dword ptr [rip + 0x1c948]
0xd8c8: mov     esi, dword ptr [rip + 0x1c93e]
0xd8ce: mov     edi, dword ptr [rip + 0x1c940]
0xd8d4: call    0xa570
0xd8d9: mov     rdx, qword ptr [rip + 0x1db28]
0xd8e0: cmp     r13, rbx
0xd8e3: jb      0xda90
0xd8e9: mov     qword ptr [rsp + 8], 0
0xd8f2: mov     rax, qword ptr [rdx + 0x18]
0xd8f6: mov     rcx, qword ptr [rdx + 0x40]
0xd8fa: mov     rdi, qword ptr [rdx + 0x28]
0xd8fe: mov     r15, rdi
0xd901: sub     r15, rcx
0xd904: cmp     r15, 3
0xd908: jbe     0xda80
0xd90e: shr     r15, 2
0xd912: movsxd  rcx, r15d
0xd915: cmp     rdi, rax
0xd918: ja      0xda70
0xd91e: mov     edx, eax
0xd920: sub     rax, rdi
0xd923: sub     edx, edi
0xd925: add     r15d, edx
0xd928: xor     edx, edx
0xd92a: div     rcx
0xd92d: sub     r15d, edx
0xd930: mov     r12d, r15d
0xd933: call    0x1d670
0xd938: mov     esi, 0x20
0xd93d: mov     rbp, rax
0xd940: mov     rax, qword ptr [rip + 0x1dac1]
0xd947: mov     rdi, rbp
0xd94a: mov     rdx, qword ptr [rax + 0x28]
0xd94e: call    0x4a50
0xd953: mov     rax, qword ptr [rip + 0x1daae]
0xd95a: mov     edx, dword ptr [rax + 8]
0xd95d: test    edx, edx
0xd95f: je      0xda40
0xd965: mov     rdx, qword ptr [rax + 0x38]
0xd969: sub     rdx, qword ptr [rax + 0x40]
0xd96d: mov     rdi, rbp
0xd970: sub     rdx, r12
0xd973: mov     rsi, qword ptr [rax + 0x28]
0xd977: sub     rsi, qword ptr [rax + 0x40]
0xd97b: cmp     rdx, rsi

```

```

0xd97e: cmova    rdx, rsi
0xd982: mov     rsi, qword ptr [rax + 0x40]
0xd986: add     rsi, r12
0xd989: add     rsi, qword ptr [rax + 0x58]
0xd98d: call    0x4bb0
0xd992: mov     rax, qword ptr [rip + 0x1da6f]
0xd999: xor     edx, edx
0xd99b: xor     esi, esi
0xd99d: mov     rdi, qword ptr [rax]
0xd9a0: call    0xc680
0xd9a5: cmp     r13, rbx
0xd9a8: jae     0xdaf0
0xd9ae: mov     rax, qword ptr [rip + 0x1da53]
0xd9b5: mov     rsi, rbp
0xd9b8: mov     rdi, qword ptr [rax]
0xd9bb: mov     rdx, qword ptr [rax + 0x28]
0xd9bf: sub     edx, dword ptr [rax + 0x40]
0xd9c2: call    0xc600
0xd9c7: mov     rax, qword ptr [rip + 0x1da3a]
0xd9ce: xor     esi, esi
0xd9d0: mov     edx, dword ptr [rax + 0x18]
0xd9d3: mov     rdi, qword ptr [rax]
0xd9d6: sub     edx, r15d
0xd9d9: call    0xc680
0xd9de: cmp     r13, rbx
0xd9e1: jb      0xdac8
0xd9e7: mov     rdi, rbp
0xd9ea: call    0x18d10
0xd9ef: mov     rdi, r14
0xd9f2: call    0xa6b0
0xd9f7: mov     rax, qword ptr [rsp + 0x18]
0xd9fc: sub     rax, qword ptr fs:[0x28]
0xda05: jne     0xdb0b
0xda0b: add     rsp, 0x28
0xda0f: pop     rbx
0xda10: pop     rbp
0xda11: pop     r12
0xda13: pop     r13
0xda15: pop     r14
0xda17: pop     r15
0xda19: ret
0xda1a: nop     word ptr [rax + rax]
0xda20: mov     edx, dword ptr [rip + 0x1c7de]
0xda26: mov     esi, dword ptr [rip + 0x1c7d4]
0xda2c: mov     edi, dword ptr [rip + 0x1c7d6]
0xda32: jmp     0xd8d4
0xd8d4
0xda37: nop     word ptr [rax + rax]

```



```

0xda40: mov     rsi, qword ptr [rax + 0x40]
0xda44: mov     rdx, qword ptr [rax + 0x38]
0xda48: mov     rdi, rbp
0xda4b: mov     rax, qword ptr [rax + 0x28]
0xda4f: sub     rdx, rsi
0xda52: sub     rdx, r12
0xda55: sub     rax, rsi
0xda58: mov     esi, 0x2a
0xda5d: cmp     rdx, rax
0xda60: cmova   rdx, rax
0xda64: call    0x4a50
0xda69: jmp     0xd992
0xd992:
0xda6e: nop
0xda70: xor     r12d, r12d
0xda73: xor     r15d, r15d
0xda76: jmp     0xd933
0xd933:
0xda7b: nop     dword ptr [rax + rax]
0xda80: mov     ecx, 1
0xda85: mov     r15d, 1
0xda8b: jmp     0xd915
0xd915:
0xda90: mov     r12, qword ptr [rdx + 0x40]
0xda94: mov     rax, qword ptr [rdx + 0x18]
0xda98: mov     qword ptr [rdx + 0x40], 0
0xdaa0: add     qword ptr [rdx + 0x58], r12
0xdaa4: movsxd rcx, r12d
0xdaa7: sub     rax, r12
0xdaaa: sub     qword ptr [rdx + 0x38], r12
0xdaae: mov     qword ptr [rsp + 8], rcx
0xdab3: xor     ecx, ecx
0xdab5: sub     qword ptr [rdx + 0x20], r12
0xdab9: mov     qword ptr [rdx + 0x18], rax
0xdabd: jmp     0xd8fa
0xd8fa:
0xdac2: nop     word ptr [rax + rax]
0xdac8: mov     rax, qword ptr [rip + 0xd939]
0xdacf: movsxd r12, dword ptr [rsp + 8]
0xdad4: sub     qword ptr [rax + 0x58], r12
0xdad8: add     qword ptr [rax + 0x38], r12
0xdadc: add     qword ptr [rax + 0x18], r12
0xdae0: add     qword ptr [rax + 0x20], r12
0xdae4: mov     qword ptr [rax + 0x40], r12
0xdae8: jmp     0xd9e7
0xd9e7:
0xdaed: nop     dword ptr [rax]
0xdaf0: mov     rax, qword ptr [rip + 0xd911]

```

```

0xdaf7: mov     rsi, qword ptr [rax + 0x58]
0xdafb: mov     edx, dword ptr [rax + 0x40]
0xdafe: mov     rdi, qword ptr [rax]
0xdb01: call    0xc600
0xdb06: jmp     0xd9ae
0xd9ae
0xdb0b: call    0x49d0
0xdb10: endbr64
0xdb14: mov     rax, qword ptr [rip + 0x1d8ed]
0xdb1b: push    rbx
0xdb1c: mov     rbx, rdi
0xdb1f: mov     rdi, qword ptr [rdi]
0xdb22: mov     rsi, qword ptr [rax + 0x48]
0xdb26: add     rsi, 1
0xdb2a: call    0x1d690
0xdb2f: mov     qword ptr [rbx], rax
0xdb32: mov     rdi, rax
0xdb35: mov     rax, qword ptr [rip + 0x1d8cc]
0xdb3c: mov     rcx, qword ptr [rax + 0x48]
0xdb40: mov     rsi, qword ptr [rax + 0x40]
0xdb44: add     rsi, qword ptr [rax + 0x58]
0xdb48: lea     rdx, [rcx + 1]
0xdb4c: call    0x4bb0
0xdb51: mov     rax, qword ptr [rip + 0x1d8b0]
0xdb58: pop     rbx
0xdb59: mov     eax, dword ptr [rax + 0x48]
0xdb5c: ret
0xdb5d: nop     dword ptr [rax]
0xdb60: endbr64
0xdb64: push    rbp
0xdb65: xor     eax, eax
0xdb67: mov     rbp, rdi
0xdb6a: call    0xcff0
0xdb6f: test    rbp, rbp
0xdb72: lea     rax, [rip + 0x133bf]
0xdb79: cmov     rbp, rax
0xdb7d: mov     rdi, rbp
0xdb80: call    0xd540
0xdb85: xor     eax, eax
0xdb87: call    0xd870
0xdb8c: xor     eax, eax
0xdb8e: pop     rbp
0xdb8f: jmp     0xd740
0xd740
0xdb94: nop     word ptr cs:[rax + rax]
0xdb9f: nop
0xdba0: endbr64
0xdba4: mov     rax, qword ptr [rip + 0x1d85d]

```

```

0xdbab: mov     dword ptr [rax + 0xc], edi
0xdbae: ret
0xdbaf: nop
0xdbb0: endbr64
0xdbb4: push     r12
0xdbb6: push     rbp
0xdbb7: push     rbx
0xdbb8: cmp      edi, 1
0xdbbb: je       0xdbd0
0xdbbd: cmp      edi, 2
0xdbc0: je       0xdc90
0xdbc6: test     edi, edi
0xdbc8: je       0xdc40
0xdbca: pop      rbx
0xdbc b: pop      rbp
0xdbcc: pop      r12
0xdbce: ret
0xdbcf: nop
0xdbd0: mov      ecx, dword ptr [rip + 0x1d83e]
0xdbd6: test     ecx, ecx
0xdbd8: je       0xdd38
0xdbde: xor      eax, eax
0xdbe0: call     0x1ab10
0xdbe5: mov      rbp, rax
0xdbe8: test     rax, rax
0xdbeb: je       0xdbca
0xdbed: mov      r12, qword ptr [rbp]
0xdbf1: mov      rbx, qword ptr [rip + 0x1d810]
0xdbf8: mov      rdi, r12
0xdbfb: call     0x49b0
0xdc00: mov      rdi, qword ptr [rbx + 0x40]
0xdc04: mov      qword ptr [rbx + 0x48], rax
0xdc08: lea      rsi, [rax + rdi]
0xdc0c: mov      qword ptr [rbx + 0x38], rsi
0xdc10: mov      qword ptr [rbx + 0x18], rsi
0xdc14: add      rsi, 1
0xdc18: cmp      rsi, qword ptr [rbx + 0x50]
0xdc1c: ja       0xdd70
0xdc22: add      rdi, qword ptr [rbx + 0x58]
0xdc26: mov      rsi, r12
0xdc29: call     0x48b0
0xdc2e: pop      rbx
0xdc2f: xor      eax, eax
0xdc31: pop      rbp
0xdc32: pop      r12
0xdc34: jmp      0xd870
0xd870
0xdc39: nop      dword ptr [rax]

```

```

0xdc40: mov     eax, dword ptr [rip + 0x1d7ce]
0xdc46: test    eax, eax
0xdc48: jne     0xdd18
0xdc4e: xor     eax, eax
0xdc50: call    0x1ab10
0xdc55: test    rax, rax
0xdc58: je      0xdd98
0xdc5e: mov     rdx, qword ptr [rip + 0x1d7a3]
0xdc65: mov     rsi, qword ptr [rax]
0xdc68: mov     rbp, qword ptr [rdx + 0x40]
0xdc6c: add     rbp, qword ptr [rdx + 0x58]
0xdc70: mov     rdi, rbp
0xdc73: call    0x4b30
0xdc78: test    eax, eax
0xdc7a: jne     0xdd58
0xdc80: pop     rbx
0xdc81: xor     eax, eax
0xdc83: pop     rbp
0xdc84: pop     r12
0xdc86: jmp     0x1ab40
0x1ab40
0xdc8b: nop     dword ptr [rax + rax]
0xdc90: mov     edx, dword ptr [rip + 0x1d77e]
0xdc96: test    edx, edx
0xdc98: je      0xdcf8
0xdc9a: xor     eax, eax
0xdc9c: call    0x1ab40
0xdca1: mov     rbp, rax
0xdca4: test    rax, rax
0xdca7: jne     0xdbed
0xdcad: mov     rax, qword ptr [rip + 0x1d754]
0xdcb4: mov     edi, 2
0xdcb9: mov     rdx, qword ptr [rax + 0x40]
0xdcbd: mov     dword ptr [rax + 0x10], 6
0xdcc4: mov     qword ptr [rax + 0x20], rdx
0xdcc8: mov     rdx, qword ptr [rax + 0x38]
0xdccc: mov     qword ptr [rax + 0x18], rdx
0xdcd0: xor     eax, eax
0xdcd2: call    0xc6e0
0xdcd7: mov     rax, qword ptr [rip + 0x1d72a]
0xdcde: mov     rdx, qword ptr [rax + 0x18]
0xdce2: mov     dword ptr [rax + 0x10], 0xd
0xdce9: mov     qword ptr [rax + 0x20], rdx
0xdced: jmp     0xdc2e
0xdc2e
0xdcf2: nop     word ptr [rax + rax]
0xdcf8: mov     dword ptr [rip + 0x1d712], 1
0xdd02: xor     eax, eax

```

```

0xdd04: call    0x1a570
0xdd09: mov     dword ptr [rip + 0x1d701], eax
0xdd0f: jmp     0xdc9a
0xdc9a:
0xdd11: nop     dword ptr [rax]
0xdd18: mov     edi, dword ptr [rip + 0x1d6f2]
0xdd1e: xor     eax, eax
0xdd20: call    0x1ac30
0xdd25: mov     dword ptr [rip + 0x1d6e5], 0
0xdd2f: jmp     0xdc4e
0xdc4e:
0xdd34: nop     dword ptr [rax]
0xdd38: mov     dword ptr [rip + 0x1d6d2], 1
0xdd42: xor     eax, eax
0xdd44: call    0x1a570
0xdd49: mov     dword ptr [rip + 0x1d6c1], eax
0xdd4f: jmp     0xdbde
0xdbde:
0xdd54: nop     dword ptr [rax]
0xdd58: xor     eax, eax
0xdd5a: mov     rdi, rbp
0xdd5d: call    0x1a380
0xdd62: xor     eax, eax
0xdd64: call    0x1ab40
0xdd69: jmp     0xdc80
0xdc80:
0xdd6e: nop
0xdd70: mov     qword ptr [rbx + 0x50], rsi
0xdd74: mov     rdi, qword ptr [rbx + 0x58]
0xdd78: call    0x1d690
0xdd7d: mov     qword ptr [rbx + 0x58], rax
0xdd81: mov     rbx, qword ptr [rip + 0x1d680]
0xdd88: mov     r12, qword ptr [rbp]
0xdd8c: mov     rdi, qword ptr [rbx + 0x40]
0xdd90: jmp     0xdc22
0xdc22:
0xdd95: nop     dword ptr [rax]
0xdd98: mov     rax, qword ptr [rip + 0x1d669]
0xdd9f: mov     rdi, qword ptr [rax + 0x40]
0xdda3: add     rdi, qword ptr [rax + 0x58]
0xdda7: xor     eax, eax
0xdda9: call    0x1a380
0xddae: jmp     0xdc80
0xdc80:
0xddb3: nop     word ptr cs:[rax + rax]
0xddbd: nop     dword ptr [rax]
0dddc0: endbr64
0dddc4: mov     eax, dword ptr [rip + 0x1be96]

```

```

0xddca: push    rbp
0xddcb: mov     rbp, rdi
0xddce: lea     rdi, [rip + 0x1be73]
0xddd5: test    eax, eax
0xddd7: lea     rax, [rip + 0x1be52]
0xddde: cmov    rdi, rax
0dde2: call    0x17c50
0dde7: mov     edx, 9
0dddec: lea     rsi, [rip + 0x1c4ad]
0xddf3: lea     rdi, [rip + 0x1c466]
0xddfa: call    0x17f70
0xddff: mov     rdi, rbp
0xde02: call    0x1d7a0
0xde07: mov     rdi, rax
0xde0a: mov     rbp, rax
0xde0d: mov     qword ptr [rip + 0x1d794], rax
0xde14: call    0x49b0
0xde19: mov     rdi, rbp
0xde1c: mov     rsi, rax
0xde1f: call    0x184d0
0xde24: xor     eax, eax
0xde26: call    0xc590
0xde2b: lea     rdi, [rip + 0x1d5ee]
0xde32: pop     rbp
0xde33: mov     qword ptr [rip + 0x1d78e], rax
0xde3a: jmp     0x4a30
0x4a30:
0xde3f: nop
0xde40: endbr64
0xde44: mov     rdi, qword ptr [rip + 0x1d77d]
0xde4b: jmp     0xc5c0
0xc5c0:
0xde50: endbr64
0xde54: push    rbp
0xde55: mov     rbp, rdi
0xde58: push    rbx
0xde59: mov     rbx, rsi
0xde5c: sub     rsp, 8
0xde60: mov     rdi, qword ptr [rip + 0x1d749]
0xde67: test    rdi, rdi
0xde6a: je      0xde71
0xde6c: call    0x18d10
0xde71: mov     rdi, rbp
0xde74: call    0x1d670
0xde79: mov     rdi, qword ptr [rip + 0x1d748]
0xde80: mov     edx, ebx
0xde82: xor     esi, esi
0xde84: mov     qword ptr [rip + 0x1d725], rax

```

```

0xde8b: add     rsp, 8
0xde8f: mov     r8d, ebp
0xde92: mov     ecx, 1
0xde97: pop     rbx
0xde98: pop     rbp
0xde99: jmp     0xc5e0
0xc5e0
0xde9e: nop
0xdea0: endbr64
0xdea4: push    rbp
0xdea5: mov     rbp, rsp
0xdea8: push    r15
0xdeaa: push    r14
0xdeac: push    r13
0xdeae: push    r12
0xdeb0: push    rbx
0xdeb1: sub     rsp, 0x68
0xdeb5: mov     rax, qword ptr fs:[0x28]
0xdeb6: mov     qword ptr [rbp - 0x38], rax
0xdec2: xor     eax, eax
0xdec4: lea     rax, [rbp - 0x61]
0xdec8: mov     rdi, rax
0xdecb: mov     qword ptr [rbp - 0x80], rax
0xdecf: call    0xa6a0
0xded4: mov     rax, qword ptr [rip + 0x1d6ed]
0xdedb: mov     esi, 0x20
0xdee0: mov     rdi, qword ptr [rip + 0x1d6c9]
0xdee7: movsxd  rdx, dword ptr [rax + 0xc]
0xdeeb: call    0x4a50
0xdef0: mov     rdi, qword ptr [rip + 0x1d6c9]
0xdef7: call    0x49b0
0xdefc: lea     r14, [rax + 1]
0xdf00: mov     rdi, r14
0xdf03: call    0x1d670
0xdf08: mov     rcx, qword ptr [rip + 0x1d6b1]
0xdf0f: mov     r13, rax
0xdf12: movzx   eax, byte ptr [rcx]
0xdf15: test    al, al
0xdf17: je      0xe430
0xdf1d: xor     esi, esi
0xdf1f: xor     ebx, ebx
0xdf21: lea     r15, [rip + 0x13188]
0xdf28: xor     edx, edx
0xdf2a: mov     dword ptr [rbp - 0x74], esi
0xdf2d: jmp     0xdf48
0xdf48
0xdf2f: nop
0xdf30: cmp     al, 9

```

```

0xdf32: je      0xdf80
0xdf34: add     dword ptr [rbp - 0x74], 1
0xdf38: mov     byte ptr [r13 + r12], al
0xdf3d: movsxd  rdx, ebx
0xdf40: movzx   eax, byte ptr [rcx + rdx]
0xdf44: test    al, al
0xdf46: je      0xdfc0
0xdf48: movsxd  r12, dword ptr [rbp - 0x74]
0xdf4c: add     ebx, 1
0xdf4f: cmp     al, 0x5c
0xdf51: jne     0xdf30
0xdf53: lea     rdx, [rcx + rdx + 1]
0xdf58: movzx   eax, byte ptr [rdx]
0xdf5b: test    al, al
0xdf5d: je      0xe320
0xdf63: sub     eax, 0x5c
0xdf66: cmp     al, 0x17
0xdf68: ja      0xe0d8
0xdf6e: movzx   eax, al
0xdf71: movsxd  rax, dword ptr [r15 + rax*4]
0xdf75: add     rax, r15
0xdf78: jmp     rax
rax
                                ->->->                                indirect JMP
0xa570: endbr64
0xa574: shl     edi, 6
0xa577: and     esi, 7
0xa57a: shl     edx, 3
0xa57d: mov     eax, edi
0xa57f: movzx   edi, byte ptr [rip + 0x20dcb]
0xa586: and     edx, 0x38
0xa589: and     eax, 0x40
0xa58c: and     edi, 0xffffffffb8
0xa58f: or      edi, eax
0xa591: or      edi, esi
0xa593: and     edi, 0xffffffffc7
0xa596: or      edi, edx
0xa598: mov     byte ptr [rip + 0x20db2], dil
0xa59f: ret
0xa5a0: endbr64
0xa5a4: push    rbp
0xa5a5: push    rbx
0xa5a6: sub     rsp, 8
0xa5aa: movsxd  rax, dword ptr [rip + 0x20957]
0xa5b1: cmp     eax, 0x400
0xa5b6: je      0xa610
0xa5b8: lea     edx, [rax + 1]
0xa5bb: lea     rbx, [rip + 0x2095e]

```



```

0xa5c2: mov     dword ptr [rip + 0x20940], edx
0xa5c8: mov     byte ptr [rbx + rax], 7
0xa5cc: test    edx, edx
0xa5ce: jle     0xa5fd
0xa5d0: xor     ebp, ebp
0xa5d2: jmp     0xa5e4
0xa5e4:
0xa5d4: nop     dword ptr [rax]
0xa5d8: mov     edx, dword ptr [rip + 0x2092a]
0xa5de: add     ebp, eax
0xa5e0: cmp     edx, ebp
0xa5e2: jle     0xa5fd
0xa5e4: sub     edx, ebp
0xa5e6: movsxd  rsi, ebp
0xa5e9: mov     edi, 1
0xa5ee: add     rsi, rbx
0xa5f1: movsxd  rdx, edx
0xa5f4: call    0x18da0
0xa5f9: test    eax, eax
0xa5fb: jns     0xa5d8
0xa5fd: mov     dword ptr [rip + 0x20901], 0
0xa607: add     rsp, 8
0xa60b: pop     rbx
0xa60c: pop     rbp
0xa60d: ret
0xa60e: nop
0xa610: xor     ebp, ebp
0xa612: lea     rbx, [rip + 0x20907]
0xa619: jmp     0xa62c
0xa62c:
0xa61b: nop     dword ptr [rax + rax]
0xa620: add     ebp, eax
0xa622: mov     eax, dword ptr [rip + 0x208e0]
0xa628: cmp     eax, ebp
0xa62a: jle     0xa645
0xa62c: sub     eax, ebp
0xa62e: movsxd  rsi, ebp
0xa631: mov     edi, 1
0xa636: add     rsi, rbx
0xa639: movsxd  rdx, eax
0xa63c: call    0x18da0
0xa641: test    eax, eax
0xa643: jns     0xa620
0xa645: mov     dword ptr [rip + 0x208b9], 1
0xa64f: mov     edx, 1
0xa654: mov     byte ptr [rip + 0x208c5], 7
0xa65b: jmp     0xa5d0
0xa5d0:

```

```

0xa660: endbr64
0xa664: test    edi, edi
0xa666: je      0xa688
0xa668: mov     rdi, qword ptr [rip + 0x1f759]
0xa66f: test    rdi, rdi
0xa672: je      0xa694
0xa674: lea     rdx, [rip - 0xc0b]
0xa67b: mov     esi, 1
0xa680: jmp     0x4820
0x4820
0xa685: nop     dword ptr [rax]
0xa688: mov     rdi, qword ptr [rip + 0x1f719]
0xa68f: test    rdi, rdi
0xa692: jne     0xa674
0xa694: ret
0xa695: nop     word ptr cs:[rax + rax]
0xa6a0: endbr64
0xa6a4: movzx   eax, byte ptr [rip + 0x20ca6]
0xa6ab: mov     byte ptr [rdi], al
0xa6ad: ret
0xa6ae: nop
0xa6b0: endbr64
0xa6b4: movzx   eax, byte ptr [rdi]
0xa6b7: mov     byte ptr [rip + 0x20c94], al
0xa6bd: ret
0xa6be: nop
0xa6c0: endbr64
0xa6c4: push    rbp
0xa6c5: push    rbx
0xa6c6: sub     rsp, 8
0xa6ca: cmp     dword ptr [rip + 0x1f58f], 1
0xa6d1: je      0xa720
0xa6d3: mov     rdi, qword ptr [rip + 0x1f66e]
0xa6da: test    rdi, rdi
0xa6dd: je      0xa6f0
0xa6df: lea     rdx, [rip - 0xc76]
0xa6e6: mov     esi, 1
0xa6eb: call    0x4820
0xa6f0: mov     dword ptr [rip + 0x20c52], 0
0xa6fa: mov     dword ptr [rip + 0x20c44], 0
0xa704: mov     dword ptr [rip + 0x20c36], 0
0xa70e: mov     dword ptr [rip + 0x20c28], 0
0xa718: add     rsp, 8
0xa71c: pop     rbx
0xa71d: pop     rbp
0xa71e: ret
0xa71f: nop
0xa720: mov     edx, dword ptr [rip + 0x207e2]

```

```

0xa726: cmp     edx, 0x3fb
0xa72c: jg       0xa750
0xa72e: movsxd   rax, edx
0xa731: lea      rcx, [rip + 0x207e8]
0xa738: add      edx, 4
0xa73b: add      rax, rcx
0xa73e: mov      ecx, dword ptr [rip + 0x1f5db]
0xa744: mov      dword ptr [rip + 0x207be], edx
0xa74a: mov      dword ptr [rax], ecx
0xa74c: jmp      0xa6d3
0xa6d3:
0xa74e: nop
0xa750: xor      ebx, ebx
0xa752: lea      rbp, [rip + 0x207c7]
0xa759: jmp      0xa76c
0xa76c:
0xa75b: nop      dword ptr [rax + rax]
0xa760: mov      edx, dword ptr [rip + 0x207a2]
0xa766: add      ebx, eax
0xa768: cmp      edx, ebx
0xa76a: jle      0xa785
0xa76c: sub      edx, ebx
0xa76e: movsxd   rsi, ebx
0xa771: mov      edi, 1
0xa776: add      rsi, rbp
0xa779: movsxd   rdx, edx
0xa77c: call     0x18da0
0xa781: test     eax, eax
0xa783: jns      0xa760
0xa785: mov      edx, 4
0xa78a: lea      rax, [rip + 0x2078f]
0xa791: jmp      0xa73e
0xa73e:
0xa793: nop      word ptr cs:[rax + rax]
0xa79e: nop
0xa7a0: push     r12
0xa7a2: mov      r12d, edi
0xa7a5: push     rbp
0xa7a6: mov      ebp, esi
0xa7a8: sub      rsp, 8
0xa7ac: mov      eax, dword ptr [rip + 0x1f71e]
0xa7b2: test     eax, eax
0xa7b4: je       0xa800
0xa7b6: mov      eax, ebp
0xa7b8: mov      esi, dword ptr [rip + 0x201da]
0xa7be: mov      edx, dword ptr [rip + 0x201d8]
0xa7c4: or       eax, r12d
0xa7c7: mov      rdi, qword ptr [rip + 0x1f63a]

```

```

0xa7ce: js      0xa7d4
0xa7d0: cmp      ebp, esi
0xa7d2: jl       0xa810
0xa7d4: sub      edx, 1
0xa7d7: sub      esi, 1
0xa7da: call     0x4960
0xa7df: add      rsp, 8
0xa7e3: mov      esi, 1
0xa7e8: lea      rdx, [rip - 0xd7f]
0xa7ef: pop      rbp
0xa7f0: mov      rdi, rax
0xa7f3: pop      r12
0xa7f5: jmp      0x4820
0x4820
0xa7fa: nop      word ptr [rax + rax]
0xa800: xor      eax, eax
0xa802: call     0xa6c0
0xa807: jmp      0xa7b6
0xa7b6
0xa809: nop      dword ptr [rax]
0xa810: cmp      r12d, edx
0xa813: jge      0xa7d4
0xa815: mov      edx, r12d
0xa818: mov      esi, ebp
0xa81a: jmp      0xa7da
0xa7da
0xa81c: nop      dword ptr [rax]
0xa820: endbr64
0xa824: push     r15
0xa826: push     r14
0xa828: push     r13
0xa82a: push     r12
0xa82c: push     rbp
0xa82d: push     rbx
0xa82e: sub      rsp, 0x38
0xa832: mov      ebx, dword ptr [rip + 0x20160]
0xa838: imul     ebx, dword ptr [rip + 0x2015d]
0xa83f: mov      rax, qword ptr fs:[0x28]
0xa848: mov      qword ptr [rsp + 0x28], rax
0xa84d: xor      eax, eax
0xa84f: mov      rdi, qword ptr [rip + 0x1f552]
0xa856: cmp      dword ptr [rip + 0x20aff], 1
0xa85d: sbb      ebx, 0
0xa860: test     rdi, rdi
0xa863: je       0xa876
0xa865: lea      rdx, [rip - 0xd7c]
0xa86c: mov      esi, 1
0xa871: call     0x4820

```

```

0xa876: test    ebx, ebx
0xa878: jle     0xab70
0xa87e: movsxd  rax, ebx
0xa881: mov     dword ptr [rsp + 4], 0xffffffff
0xa889: xor     ebp, ebp
0xa88b: mov     r13d, 0xffffffff
0xa891: mov     qword ptr [rsp + 8], rax
0xa896: lea     r12, [rip + 0x20683]
0xa89d: jmp     0xa9ce
0xa9ce:
0xa8a2: nop     word ptr [rax + rax]
0xa8a8: cdq
0xa8a9: idiv    dword ptr [rip + 0x200e9]
0xa8af: mov     r14d, edx
0xa8b2: cmp     edx, r13d
0xa8b5: jne     0xaa20
0xa8bb: cmp     eax, dword ptr [rsp + 4]
0xa8bf: jne     0xaa20
0xa8c5: mov     rax, qword ptr [rip + 0x20a64]
0xa8cc: movzx   ebx, byte ptr [rax + rbp]
0xa8d0: mov     edi, ebx
0xa8d2: mov     r14d, ebx
0xa8d5: sar     edi, 6
0xa8d8: and     r14d, 7
0xa8dc: and     edi, 1
0xa8df: call    0x9870
0xa8e4: mov     eax, ebx
0xa8e6: and     eax, 7
0xa8e9: cmp     dword ptr [rip + 0x20a5c], 1
0xa8f0: je      0xaa40
0xa8f6: cmp     dword ptr [rip + 0x1f363], 1
0xa8fd: je      0xaa5f
0xa903: xor     edi, edi
0xa905: cmp     eax, 7
0xa908: setne   dil
0xa90c: call    0x9790
0xa911: movzx   eax, byte ptr [rip + 0x20a38]
0xa918: mov     dword ptr [rip + 0x20a2a], 1
0xa922: and     eax, 0xffffffff8
0xa925: or      eax, r14d
0xa928: mov     byte ptr [rip + 0x20a22], al
0xa92e: sar     ebx, 3
0xa931: and     ebx, 7
0xa934: cmp     dword ptr [rip + 0x20a0d], 1
0xa93b: je      0xaab0
0xa941: cmp     dword ptr [rip + 0x1f318], 1
0xa948: je      0xaad8
0xa94e: xor     edi, edi

```

```

0xa950: test    ebx, ebx
0xa952: setne   dil
0xa956: call    0x9790
0xa95b: mov     edx, dword ptr [rip + 0x205a7]
0xa961: movzx   eax, byte ptr [rip + 0x209e8]
0xa968: shl     ebx, 3
0xa96b: mov     dword ptr [rip + 0x209d3], 1
0xa975: and     eax, 0xffffffffc7
0xa978: or      ebx, eax
0xa97a: mov     byte ptr [rip + 0x209d0], bl
0xa980: mov     rax, qword ptr [rip + 0x209b1]
0xa987: movzx   ebx, byte ptr [rax + rbp]
0xa98b: cmp     edx, 0x400
0xa991: je      0xab30
0xa997: lea     eax, [rdx + 1]
0xa99a: movsxd  rdx, edx
0xa99d: add     r13d, 1
0xa9a1: cmp     dword ptr [rip + 0x1fff0], r13d
0xa9a8: mov     dword ptr [rip + 0x2055a], eax
0xa9ae: mov     byte ptr [r12 + rdx], bl
0xa9b2: jne     0xa9bf
0xa9b4: mov     dword ptr [rsp + 4], 0xffffffff
0xa9bc: xor     r13d, r13d
0xa9bf: add     rbp, 1
0xa9c3: cmp     rbp, qword ptr [rsp + 8]
0xa9c8: je      0xab70
0xa9ce: mov     rdx, qword ptr [rip + 0x20953]
0xa9d5: mov     rcx, qword ptr [rip + 0x2095c]
0xa9dc: mov     eax, ebp
0xa9de: movzx   ebx, byte ptr [rdx + rbp]
0xa9e2: cmp     byte ptr [rcx + rbp], bl
0xa9e5: jne     0xa8a8
0xa9eb: mov     rdx, qword ptr [rip + 0x2092e]
0xa9f2: mov     rcx, qword ptr [rip + 0x20937]
0xa9f9: movzx   edi, byte ptr [rdx + rbp]
0xa9fd: cmp     byte ptr [rcx + rbp], dil
0xaa01: je      0xa9bf
0xaa03: cdq
0xaa04: idiv    dword ptr [rip + 0x1ff8e]
0xaa0a: mov     r14d, edx
0xaa0d: cmp     edx, r13d
0xaa10: je      0xa8bb
0xaa16: nop     word ptr cs:[rax + rax]
0xaa20: mov     esi, r14d
0xaa23: mov     edi, eax
0xaa25: mov     dword ptr [rsp + 4], eax
0xaa29: mov     r13d, r14d
0xaa2c: call    0xa7a0

```

```

0xaa31: jmp      0xa8c5
0xa8c5:
0xaa36: nop      word ptr cs:[rax + rax]
0xaa40: movzx   edx, byte ptr [rip + 0x20909]
0xaa47: and     edx, 7
0xaa4a: cmp     eax, edx
0xaa4c: je      0xa92e
0xaa52: cmp     dword ptr [rip + 0x1f207], 1
0xaa59: jne     0xa903
0xaa5f: mov     eax, dword ptr [rip + 0x164b4]
0xaa65: mov     edx, dword ptr [rip + 0x2049d]
0xaa6b: mov     dword ptr [rsp + 0x10], eax
0xaa6f: movzx   eax, byte ptr [rip + 0x164a7]
0xaa76: mov     byte ptr [rsp + 0x14], al
0xaa7a: lea     eax, [r14 + 0x30]
0xaa7e: mov     byte ptr [rsp + 0x13], al
0xaa82: cmp     edx, 0x3fa
0xaa88: jg      0xac18
0xaa8e: movsxd  rax, edx
0xaa91: add     edx, 5
0xaa94: add     rax, r12
0xaa97: mov     ecx, dword ptr [rsp + 0x10]
0xaa9b: mov     dword ptr [rip + 0x20467], edx
0xaaa1: mov     dword ptr [rax], ecx
0xaaa3: movzx   ecx, byte ptr [rsp + 0x14]
0xaaa8: mov     byte ptr [rax + 4], cl
0xaaab: jmp     0xa911
0xa911:
0xaab0: movzx   eax, byte ptr [rip + 0x20899]
0xaab7: mov     edx, dword ptr [rip + 0x2044b]
0xaabd: sar     eax, 3
0xaac0: and     eax, 7
0xaac3: cmp     ebx, eax
0xaac5: je      0xa980
0xaacb: cmp     dword ptr [rip + 0x1f18e], 1
0xaad2: jne     0xa94e
0xaad8: mov     eax, dword ptr [rip + 0x16436]
0xaade: mov     edx, dword ptr [rip + 0x20424]
0xaae4: mov     dword ptr [rsp + 0x10], eax
0xaae8: movzx   eax, byte ptr [rip + 0x16429]
0xaaef: mov     byte ptr [rsp + 0x14], al
0xaaf3: lea     eax, [rbx + 0x30]
0xaaf6: mov     byte ptr [rsp + 0x13], al
0xaafa: cmp     edx, 0x3fa
0xab00: jg      0xac58
0xab06: movsxd  rax, edx
0xab09: add     edx, 5
0xab0c: add     rax, r12

```

```

0xab0f: mov     ecx, dword ptr [rsp + 0x10]
0xab13: mov     dword ptr [rip + 0x203ef], edx
0xab19: mov     dword ptr [rax], ecx
0xab1b: movzx   ecx, byte ptr [rsp + 0x14]
0xab20: mov     byte ptr [rax + 4], cl
0xab23: jmp     0xa961
0xa961
0xab28: nop     dword ptr [rax + rax]
0xab30: xor     r14d, r14d
0xab33: jmp     0xab46
0xab46
0xab35: nop     dword ptr [rax]
0xab38: mov     edx, dword ptr [rip + 0x203ca]
0xab3e: add     r14d, eax
0xab41: cmp     edx, r14d
0xab44: jle     0xab60
0xab46: sub     edx, r14d
0xab49: movsxd  rsi, r14d
0xab4c: mov     edi, 1
0xab51: add     rsi, r12
0xab54: movsxd  rdx, edx
0xab57: call    0x18da0
0xab5c: test    eax, eax
0xab5e: jns     0xab38
0xab60: mov     eax, 1
0xab65: xor     edx, edx
0xab67: jmp     0xa99a
0xa99a
0xab6c: nop     dword ptr [rax]
0xab70: mov     edi, dword ptr [rip + 0x207de]
0xab76: mov     esi, dword ptr [rip + 0x207dc]
0xab7c: call    0xa7a0
0xab81: mov     rdi, qword ptr [rip + 0x1f240]
0xab88: test    rdi, rdi
0xab8b: je      0xab9e
0xab8d: lea     rdx, [rip - 0x1124]
0xab94: mov     esi, 1
0xab99: call    0x4820
0xab9e: mov     edx, dword ptr [rip + 0x20364]
0xaba4: test    edx, edx
0xaba6: jne     0xac98
0xabac: mov     edx, dword ptr [rip + 0x1fde6]
0xabb2: imul    edx, dword ptr [rip + 0x1fde3]
0xabb9: mov     rsi, qword ptr [rip + 0x20778]
0xabc0: mov     rdi, qword ptr [rip + 0x20761]
0xabc7: movsxd  rdx, edx
0xabca: call    0x4bb0
0xabcf: mov     edx, dword ptr [rip + 0x1fdc3]

```



```

0xabd5: imul    edx, dword ptr [rip + 0x1fdc0]
0xabdc: mov     rdi, qword ptr [rip + 0x2073d]
0xabe3: mov     rsi, qword ptr [rip + 0x20746]
0xabea: movsxd  rdx, edx
0xabed: mov     rax, qword ptr [rsp + 0x28]
0xabf2: sub     rax, qword ptr fs:[0x28]
0xabfb: jne     0xace4
0xac01: add     rsp, 0x38
0xac05: pop     rbx
0xac06: pop     rbp
0xac07: pop     r12
0xac09: pop     r13
0xac0b: pop     r14
0xac0d: pop     r15
0xac0f: jmp     0x4bb0
0x4bb0
0xac14: nop     dword ptr [rax]
0xac18: xor     r15d, r15d
0xac1b: jmp     0xac2e
0xac2e
0xac1d: nop     dword ptr [rax]
0xac20: mov     edx, dword ptr [rip + 0x202e2]
0xac26: add     r15d, eax
0xac29: cmp     edx, r15d
0xac2c: jle     0xac48
0xac2e: sub     edx, r15d
0xac31: movsxd  rsi, r15d
0xac34: mov     edi, 1
0xac39: add     rsi, r12
0xac3c: movsxd  rdx, edx
0xac3f: call    0x18da0
0xac44: test    eax, eax
0xac46: jns     0xac20
0xac48: mov     edx, 5
0xac4d: mov     rax, r12
0xac50: jmp     0xaa97
0xaa97
0xac55: nop     dword ptr [rax]
0xac58: xor     r14d, r14d
0xac5b: jmp     0xac6e
0xac6e
0xac5d: nop     dword ptr [rax]
0xac60: mov     edx, dword ptr [rip + 0x202a2]
0xac66: add     r14d, eax
0xac69: cmp     edx, r14d
0xac6c: jle     0xac88
0xac6e: sub     edx, r14d
0xac71: movsxd  rsi, r14d

```

```

0xac74: mov     edi, 1
0xac79: add     rsi, r12
0xac7c: movsxd  rdx, edx
0xac7f: call    0x18da0
0xac84: test    eax, eax
0xac86: jns     0xac60
0xac88: mov     edx, 5
0xac8d: mov     rax, r12
0xac90: jmp     0xab0f
0xab0f:
0xac95: nop     dword ptr [rax]
0xac98: mov     ebx, 0
0xac9d: lea     rbp, [rip + 0x2027c]
0xaca4: jg      0xacbc
0xaca6: jmp     0xacd5
0xacd5:
0xaca8: nop     dword ptr [rax + rax]
0xacb0: mov     edx, dword ptr [rip + 0x20252]
0xacb6: add     ebx, eax
0xacb8: cmp     edx, ebx
0xacba: jle     0xacd5
0xacbc: sub     edx, ebx
0xacbe: movsxd  rsi, ebx
0xacc1: mov     edi, 1
0xacc6: add     rsi, rbp
0xacc9: movsxd  rdx, edx
0xaccc: call    0x18da0
0xacd1: test    eax, eax
0xacd3: jns     0xacb0
0xacd5: mov     dword ptr [rip + 0x20229], 0
0xacdf: jmp     0xabac
0xabac:
0xace4: call    0x49d0
0xace9: nop     dword ptr [rax]
0xacf0: endbr64
0xacf4: push    r12
0xacf6: xor     eax, eax
0xacf8: mov     r12, rdi
0xacfb: push    rbx
0xacfc: mov     ebx, esi
0xacfe: sub     rsp, 8
0xad02: call    0xa820
0xad07: mov     rdx, qword ptr [rip + 0x1fc1a]
0xad0e: test    rdx, rdx
0xad11: je      0xad17
0xad13: xor     eax, eax
0xad15: call    rdx
0xad17: movsxd  rdx, ebx

```

```

0xad1a: mov     rsi, r12
0xad1d: xor     edi, edi
0xad1f: call    0x18d30
0xad24: mov     rdx, qword ptr [rip + 0x1fbf5]
0xad2b: mov     r12d, eax
0xad2e: test    rdx, rdx
0xad31: je      0xad37
0xad33: xor     eax, eax
0xad35: call    rdx
0xad37: add     rsp, 8
0xad3b: mov     eax, r12d
0xad3e: pop     rbx
0xad3f: pop     r12
0xad41: ret
0xad42: nop     word ptr cs:[rax + rax]
0xad4d: nop     dword ptr [rax]
0xad50: push    r12
0xad52: mov     edi, 1
0xad57: push    rbp
0xad58: lea     rbp, [rip + 0x1fda1]
0xad5f: push    rbx
0xad60: sub     rsp, 0x50
0xad64: mov     rax, qword ptr fs:[0x28]
0xad6d: mov     qword ptr [rsp + 0x48], rax
0xad72: xor     eax, eax
0xad74: mov     r12, rsp
0xad77: mov     rsi, r12
0xad7a: call    0x4d60
0xad7f: mov     edx, 0xff
0xad84: mov     esi, 1
0xad89: mov     edi, 1
0xad8e: mov     word ptr [rsp + 0x11], dx
0xad93: mov     rdx, r12
0xad96: call    0x4d70
0xad9b: mov     edi, 1
0xada0: mov     dword ptr [rip + 0x1ef7a], 0xffffffff
0xadaa: call    0x18b70
0xadaf: mov     dword ptr [rip + 0x20147], 0
0xadb9: nop     dword ptr [rax]
0xadc0: xor     eax, eax
0xadc2: call    0xa820
0xadc7: mov     rdx, qword ptr [rip + 0x1fb5a]
0xadce: test    rdx, rdx
0xadd1: je      0xadd7
0xadd3: xor     eax, eax
0xadd5: call    rdx
0xadd7: mov     edx, 0x400
0xaddc: mov     rsi, rbp

```

```

0xaddf: xor     edi, edi
0xade1: call    0x18d30
0xade6: mov     rdx, qword ptr [rip + 0x1fb33]
0xaded: mov     ebx, eax
0xadef: test    rdx, rdx
0xadf2: je      0xadf8
0xadf4: xor     eax, eax
0xadf6: call    rdx
0xadf8: mov     dword ptr [rip + 0x20106], ebx
0xadfe: test    ebx, ebx
0xae00: js      0xadc0
0xae02: xor     edi, edi
0xae04: call    0x18b70
0xae09: mov     rsi, r12
0xae0c: mov     edi, 1
0xae11: call    0x4d60
0xae16: mov     eax, 7
0xae1b: mov     rdx, r12
0xae1e: mov     esi, 1
0xae23: mov     edi, 1
0xae28: mov     word ptr [rsp + 0x11], ax
0xae2d: call    0x4d70
0xae32: mov     eax, dword ptr [rip + 0x200cc]
0xae38: mov     dword ptr [rip + 0x1eee2], 7
0xae42: test    eax, eax
0xae44: je      0xae9f
0xae46: movsxd  rsi, dword ptr [rip + 0x200b3]
0xae4d: movsx   r8d, byte ptr [rbp + rsi]
0xae53: mov     rdx, rsi
0xae56: mov     ecx, r8d
0xae59: test    r8b, r8b
0xae5c: js      0xae90
0xae5e: sub     eax, 1
0xae61: add     edx, 1
0xae64: mov     dword ptr [rip + 0x2009a], eax
0xae6a: mov     dword ptr [rip + 0x20090], edx
0xae70: mov     rax, qword ptr [rsp + 0x48]
0xae75: sub     rax, qword ptr fs:[0x28]
0xae7e: jne     0xaea7
0xae80: add     rsp, 0x50
0xae84: mov     eax, r8d
0xae87: pop     rbx
0xae88: pop     rbp
0xae89: pop     r12
0xae8b: ret
0xae8c: nop     dword ptr [rax]
0xae90: and     ecx, 0x7f
0xae93: mov     r8d, 0x1b

```

```

0xae99: mov     byte ptr [rbp + rsi], cl
0xae9d: jmp     0xae70
0xae70:
0xae9f: mov     r8d, 0xffffffff
0xaea5: jmp     0xae70
0xae70:
0xaea7: call    0x49d0
0xaeac: nop     dword ptr [rax]
0xaeb0: endbr64
0xaeb4: xor     eax, eax
0xaeb6: sub     rsp, 8
0xaeba: call    0x18ab0
0xaebf: mov     eax, dword ptr [rip + 0x2003f]
0xaec5: test    eax, eax
0xaec7: jne     0xaed8
0xaec9: xor     eax, eax
0xaecb: add     rsp, 8
0xaecf: jmp     0xad50
0xad50:
0xaed4: nop     dword ptr [rax]
0xaed8: movsxd  rsi, dword ptr [rip + 0x20021]
0xaedf: lea     rdi, [rip + 0x1fc1a]
0xae6: movsx   r8d, byte ptr [rdi + rsi]
0xaeeb: mov     rdx, rsi
0xaeee: mov     ecx, r8d
0xae1: test    r8b, r8b
0xae14: js      0xaf10
0xae16: sub     eax, 1
0xae19: add     edx, 1
0xae1c: mov     dword ptr [rip + 0x20002], eax
0xaf02: mov     dword ptr [rip + 0x1fff8], edx
0xaf08: mov     eax, r8d
0xaf0b: add     rsp, 8
0xaf0f: ret
0xaf10: and     ecx, 0x7f
0xaf13: mov     r8d, 0x1b
0xaf19: mov     byte ptr [rdi + rsi], cl
0xaf1c: jmp     0xaf08
0xaf08:
0xaf1e: nop
0xaf20: endbr64
0xaf24: push    r13
0xaf26: mov     r13, rsi
0xaf29: push    r12
0xaf2b: mov     r12, rdx
0xaf2e: push    rbp
0xaf2f: mov     rbp, rdi
0xaf32: mov     edi, 0x18

```

```

0xaf37: push    rbx
0xaf38: sub      rsp, 8
0xaf3c: call     0x1d670
0xaf41: mov      rdi, r13
0xaf44: mov      rbx, rax
0xaf47: call     0x1d7a0
0xaf4c: mov      qword ptr [rbx + 0x10], r12
0xaf50: mov      qword ptr [rbx], rax
0xaf53: mov      rax, qword ptr [rbp]
0xaf57: mov      qword ptr [rbx + 8], rax
0xaf5b: mov      qword ptr [rbp], rbx
0xaf5f: add      rsp, 8
0xaf63: pop      rbx
0xaf64: pop      rbp
0xaf65: pop      r12
0xaf67: pop      r13
0xaf69: ret
0xaf6a: nop      word ptr [rax + rax]
0xaf70: endbr64
0xaf74: cmp      byte ptr [rdi], 0
0xaf77: je       0xb030
0xaf7d: push     r13
0xaf7f: push     r12
0xaf81: mov      r12, rsi
0xaf84: push     rbp
0xaf85: mov      rbp, rdi
0xaf88: push     rbx
0xaf89: sub      rsp, 8
0xaf8d: mov      r13, qword ptr [rip + 0x1fa0c]
0xaf94: test     r13, r13
0xaf97: je       0xb010
0xaf99: mov      rbx, qword ptr [r13]
0xaf9d: mov      rsi, qword ptr [rbx]
0xafa0: call     0x4b30
0xafa5: test     eax, eax
0xafa7: jle     0xb010
0xafa9: nop      dword ptr [rax]
0xafb0: lea      r13, [rbx + 8]
0xafb4: mov      rbx, qword ptr [rbx + 8]
0xafb8: mov      qword ptr [rip + 0x1f9e1], r13
0xafbf: test     rbx, rbx
0xafc2: je       0xafd3
0xafc4: mov      rsi, qword ptr [rbx]
0xafc7: mov      rdi, rbp
0xafca: call     0x4b30
0xafcf: test     eax, eax
0xafd1: jg       0xafb0
0xafd3: mov      edi, 0x18

```

```

0xafd8: call    0x1d670
0xafdd: mov     rdi, rbp
0xafe0: mov     rbx, rax
0xafe3: call    0x1d7a0
0xafe8: mov     qword ptr [rbx + 0x10], r12
0xafec: mov     qword ptr [rbx], rax
0xafef: mov     rax, qword ptr [r13]
0xaff3: mov     qword ptr [rbx + 8], rax
0xaff7: mov     qword ptr [r13], rbx
0xaffb: add     rsp, 8
0xafff: pop     rbx
0xb000: pop     rbp
0xb001: pop     r12
0xb003: pop     r13
0xb005: ret
0xb006: nop     word ptr cs:[rax + rax]
0xb010: mov     rbx, qword ptr [rip + 0x1f969]
0xb017: lea     r13, [rip + 0x1f962]
0xb01e: mov     qword ptr [rip + 0x1f97b], r13
0xb025: test    rbx, rbx
0xb028: jne     0xafc4
0xb02a: jmp     0xafd3
0xafd3:
0xb02c: nop     dword ptr [rax]
0xb030: ret
0xb031: nop     word ptr cs:[rax + rax]
0xb03c: nop     dword ptr [rax]
0xb040: endbr64
0xb044: mov     rax, qword ptr [rip + 0x1f935]
0xb04b: mov     qword ptr [rip + 0x1f926], rax
0xb052: ret
0xb053: nop     word ptr cs:[rax + rax]
0xb05e: nop
0xb060: endbr64
0xb064: cmp     qword ptr [rip + 0x1f90c], 0
0xb06c: je      0xb073
0xb06e: jmp     0x99e0
0x99e0:
0xb073: xor     eax, eax
0xb075: ret
0xb076: nop     word ptr cs:[rax + rax]
0xb080: endbr64
0xb084: mov     eax, dword ptr [rip + 0x1f912]
0xb08a: push    r12
0xb08c: mov     r12, rdi
0xb08f: push    rbp
0xb090: push    rbx
0xb091: sub     eax, 1

```

```

0xb094: movzx    ebx, byte ptr [rip + 0x202b6]
0xb09b: mov     dword ptr [rip + 0x202b3], eax
0xb0a1: mov     dword ptr [rip + 0x202ad], 0
0xb0ab: mov     eax, ebx
0xb0ad: and     eax, 0xffffffffc0
0xb0b0: or      eax, 0x38
0xb0b3: mov     byte ptr [rip + 0x20298], al
0xb0b9: mov     eax, dword ptr [rip + 0x1f8d9]
0xb0bf: lea     edi, [rax + 1]
0xb0c2: movsxd  rdi, edi
0xb0c5: call    0x1d670
0xb0ca: movsxd  rdx, dword ptr [rip + 0x1f8c7]
0xb0d1: mov     esi, 0x20
0xb0d6: mov     rdi, rax
0xb0d9: mov     rbp, rax
0xb0dc: call    0x4a50
0xb0e1: movsxd  rax, dword ptr [rip + 0x1f8b0]
0xb0e8: mov     rdi, rbp
0xb0eb: mov     byte ptr [rbp + rax], 0
0xb0f0: mov     esi, dword ptr [rip + 0x1f8a2]
0xb0f6: call    0xa1a0
0xb0fb: mov     rdi, rbp
0xb0fe: lea     rbp, [rip + 0x1f8bb]
0xb105: call    0x18d10
0xb10a: mov     eax, dword ptr [rip + 0x1f88c]
0xb110: mov     rdi, r12
0xb113: mov     dword ptr [rip + 0x2023b], 0
0xb11d: sub     eax, 1
0xb120: mov     dword ptr [rip + 0x2022e], eax
0xb126: call    0x9fe0
0xb12b: mov     esi, 0x13
0xb130: lea     rdi, [rip + 0x15ad5]
0xb137: call    0xa1a0
0xb13c: mov     rdi, rbp
0xb13f: call    0x49b0
0xb144: mov     rdi, rbp
0xb147: mov     esi, eax
0xb149: call    0xa1a0
0xb14e: mov     esi, 1
0xb153: lea     rdi, [rip + 0x155eb]
0xb15a: call    0xa1a0
0xb15f: xor     eax, eax
0xb161: call    0xa820
0xb166: mov     byte ptr [rip + 0x201e5], bl
0xb16c: pop     rbx
0xb16d: pop     rbp
0xb16e: pop     r12
0xb170: ret

```



```

0xb171: nop      word ptr cs:[rax + rax]
0xb17c: nop      dword ptr [rax]
0xb180: endbr64
0xb184: push     r15
0xb186: push     r14
0xb188: mov      r14, rdi
0xb18b: push     r13
0xb18d: push     r12
0xb18f: push     rbp
0xb190: lea      rbp, [rip + 0x1f969]
0xb197: push     rbx
0xb198: sub      rsp, 0x28
0xb19c: jmp      0xb1ae
0xb1ae:
0xb19e: nop
0xb1a0: xor      eax, eax
0xb1a2: call     0xad50
0xb1a7: mov      ebx, eax
0xb1a9: cmp      ebx, -1
0xb1ac: jne      0xb1ef
0xb1ae: xor      eax, eax
0xb1b0: call     0x18ab0
0xb1b5: mov      eax, dword ptr [rip + 0x1fd49]
0xb1bb: test     eax, eax
0xb1bd: je       0xb1a0
0xb1bf: movsxd   rsi, dword ptr [rip + 0x1fd3a]
0xb1c6: movsx    ebx, byte ptr [rbp + rsi]
0xb1cb: mov      rdx, rsi
0xb1ce: mov      ecx, ebx
0xb1d0: test     bl, bl
0xb1d2: js       0xb4a0
0xb1d8: sub      eax, 1
0xb1db: add      edx, 1
0xb1de: mov      dword ptr [rip + 0x1fd20], eax
0xb1e4: mov      dword ptr [rip + 0x1fd16], edx
0xb1ea: cmp      ebx, -1
0xb1ed: je       0xb1ae
0xb1ef: test     r14, r14
0xb1f2: je       0xb1fb
0xb1f4: mov      dword ptr [r14], 1
0xb1fb: mov      eax, dword ptr [rip + 0x201fb]
0xb201: test     ebx, ebx
0xb203: jne      0xb445
0xb209: mov      ebx, 0xff
0xb20e: test     eax, eax
0xb210: jne      0xb240
0xb212: xor      r12d, r12d
0xb215: mov      ebp, 0x1fe

```

```

0xb21a: mov     ebx, 0xff
0xb21f: call    0x4f00
0xb224: mov     rax, qword ptr [rax]
0xb227: movzx   eax, word ptr [rax + rbp]
0xb22b: test    ah, 0x40
0xb22e: jne     0xb47d
0xb234: test    r12b, r12b
0xb237: jne     0xb47d
0xb23d: nop     dword ptr [rax]
0xb240: mov     rax, qword ptr [rip + 0x1f739]
0xb247: xor     r12d, r12d
0xb24a: lea     r13, [rip + 0x1f8af]
0xb251: mov     dword ptr [rip + 0x1f751], 0
0xb25b: mov     dword ptr [rip + 0x1f743], 0
0xb265: mov     qword ptr [rip + 0x1f70c], rax
0xb26c: test    ebx, ebx
0xb26e: mov     eax, 0xffffffff
0xb273: movsxd  rbp, r12d
0xb276: cmovne  eax, ebx
0xb279: mov     rbx, qword ptr [rip + 0x200e0]
0xb280: mov     byte ptr [rbx + rbp], al
0xb283: add     rbp, 1
0xb287: mov     byte ptr [rbx + rbp], 0
0xb28b: cmp     qword ptr [rip + 0x1f6e5], 0
0xb293: je      0xb2b3
0xb295: mov     rdi, rbx
0xb298: call    0x99e0
0xb29d: mov     r15, rax
0xb2a0: cmp     rax, -1
0xb2a4: je      0xb33a
0xb2aa: test    rax, rax
0xb2ad: jne     0xb38f
0xb2b3: mov     ecx, dword ptr [rip + 0x1fc4b]
0xb2b9: test    ecx, ecx
0xb2bb: jne     0xb2ee
0xb2bd: mov     edx, dword ptr [rip + 0x1f6e5]
0xb2c3: test    edx, edx
0xb2c5: je      0xb380
0xb2cb: mov     rdi, rbx
0xb2ce: call    0xb080
0xb2d3: mov     dword ptr [rip + 0x1f6cf], 1
0xb2dd: jmp     0xb2ee
0xb2ee
0xb2df: nop
0xb2e0: xor     eax, eax
0xb2e2: call    0xad50
0xb2e7: mov     ebx, eax
0xb2e9: cmp     ebx, -1

```

```

0xb2ec: jne      0xb32c
0xb2ee: xor       eax, eax
0xb2f0: call     0x18ab0
0xb2f5: mov      eax, dword ptr [rip + 0x1fc09]
0xb2fb: test     eax, eax
0xb2fd: je       0xb2e0
0xb2ff: movsxd   rsi, dword ptr [rip + 0x1fbfa]
0xb306: movsx    ebx, byte ptr [r13 + rsi]
0xb30c: mov      rdx, rsi
0xb30f: mov      ecx, ebx
0xb311: test     bl, bl
0xb313: js       0xb360
0xb315: sub      eax, 1
0xb318: add      edx, 1
0xb31b: mov      dword ptr [rip + 0x1fbe3], eax
0xb321: mov      dword ptr [rip + 0x1fbd9], edx
0xb327: cmp      ebx, -1
0xb32a: je       0xb2ee
0xb32c: add      r12d, 1
0xb330: cmp      r12d, 0xf
0xb334: jne      0xb26c
0xb33a: mov      edi, 1
0xb33f: xor      r15d, r15d
0xb342: call     0x4aa0
0xb347: mov      dword ptr [rip + 0x1f65b], 0
0xb351: jmp      0xb433
0xb433:
0xb356: nop      word ptr cs:[rax + rax]
0xb360: and      ecx, 0x7f
0xb363: add      r12d, 1
0xb367: mov      byte ptr [r13 + rsi], cl
0xb36c: cmp      r12d, 0xf
0xb370: je       0xb33a
0xb372: mov      eax, 0x1b
0xb377: jmp      0xb279
0xb279:
0xb37c: nop      dword ptr [rax]
0xb380: mov      edi, 1
0xb385: call     0x4aa0
0xb38a: jmp      0xb2d3
0xb2d3:
0xb38f: test     r14, r14
0xb392: je       0xb41f
0xb398: mov      ecx, dword ptr [rip + 0x1fb66]
0xb39e: cmp      r12d, ecx
0xb3a1: jge      0xb41f
0xb3a3: movsxd   rax, dword ptr [rip + 0x1fb56]
0xb3aa: lea      r9d, [r12 + 1]

```

```

0xb3af: lea      rdx, [rip + 0x1f74a]
0xb3b6: movsxd   rdi, r9d
0xb3b9: mov      qword ptr [rsp + 0x18], rdi
0xb3be: lea      r8d, [r9 + rax]
0xb3c2: lea      r13, [rax + rdx]
0xb3c6: jmp      0xb3f1
0xb3f1:
0xb3c8: nop      dword ptr [rax + rax]
0xb3d0: sub      ecx, r9d
0xb3d3: add      dword ptr [r14], 1
0xb3d7: add      r13, qword ptr [rsp + 0x18]
0xb3dc: mov      dword ptr [rip + 0x1fb1d], r8d
0xb3e3: add      r8d, r9d
0xb3e6: mov      dword ptr [rip + 0x1fb18], ecx
0xb3ec: cmp      r12d, ecx
0xb3ef: jge      0xb41f
0xb3f1: mov      rdx, rbp
0xb3f4: mov      rsi, r13
0xb3f7: mov      rdi, rbx
0xb3fa: mov      dword ptr [rsp + 0x14], r8d
0xb3ff: mov      dword ptr [rsp + 0x10], r9d
0xb404: mov      dword ptr [rsp + 0xc], ecx
0xb408: call     0x4af0
0xb40d: mov      ecx, dword ptr [rsp + 0xc]
0xb411: mov      r9d, dword ptr [rsp + 0x10]
0xb416: test     eax, eax
0xb418: mov      r8d, dword ptr [rsp + 0x14]
0xb41d: je       0xb3d0
0xb41f: mov      edi, 1
0xb424: call     0x4aa0
0xb429: mov      dword ptr [rip + 0x1f579], 0
0xb433: add      rsp, 0x28
0xb437: mov      rax, r15
0xb43a: pop      rbx
0xb43b: pop      rbp
0xb43c: pop      r12
0xb43e: pop      r13
0xb440: pop      r14
0xb442: pop      r15
0xb444: ret
0xb445: test     eax, eax
0xb447: jne      0xb240
0xb44d: cmp      ebx, 0xa
0xb450: sete     bpl
0xb454: cmp      ebx, 0xd
0xb457: sete     al
0xb45a: or       ebp, eax
0xb45c: call     0x4f00

```

```

0xb461: mov     rax, qword ptr [rax]
0xb464: test    bpl, bpl
0xb467: je      0xb4c6
0xb469: mov     ebx, 0xa
0xb46e: test    byte ptr [rax + 0x15], 0x40
0xb472: je      0xb240
0xb478: mov     ebx, 0xa
0xb47d: mov     rax, qword ptr [rip + 0x1f4dc]
0xb484: lea     r15, [rip + 0x1f4d5]
0xb48b: mov     byte ptr [rax], bl
0xb48d: mov     rax, qword ptr [rip + 0x1f4cc]
0xb494: mov     byte ptr [rax + 1], 0
0xb498: jmp     0xb433
0xb433:
0xb49a: nop     word ptr [rax + rax]
0xb4a0: and     ecx, 0x7f
0xb4a3: mov     byte ptr [rbp + rsi], cl
0xb4a7: test    r14, r14
0xb4aa: je      0xb4b3
0xb4ac: mov     dword ptr [r14], 1
0xb4b3: mov     eax, dword ptr [rip + 0x1ff43]
0xb4b9: mov     ebx, 0x1b
0xb4be: test    eax, eax
0xb4c0: jne     0xb240
0xb4c6: movsxd  rbp, ebx
0xb4c9: add     rbp, rbp
0xb4cc: cmp     ebx, 7
0xb4cf: sete    r12b
0xb4d3: jmp     0xb21f
0xb21f:
0xb4d8: nop     dword ptr [rax + rax]
0xb4e0: endbr64
0xb4e4: mov     eax, dword ptr [rip + 0x1f4c2]
0xb4ea: test    eax, eax
0xb4ec: jne     0xb4f0
0xb4ee: ret
0xb4ef: nop
0xb4f0: sub     rsp, 8
0xb4f4: mov     rdi, qword ptr [rip + 0x1fe65]
0xb4fb: call    0xb080
0xb500: mov     dword ptr [rip + 0x1f49e], 1
0xb50a: add     rsp, 8
0xb50e: ret
0xb50f: nop
0xb510: endbr64
0xb514: mov     rax, qword ptr [rip + 0x1fe45]
0xb51b: ret
0xb51c: nop     dword ptr [rax]

```

```

0xb520: endbr64
0xb524: push    r15
0xb526: mov     esi, 0x5413
0xb52b: mov     edi, 1
0xb530: push    r14
0xb532: push    r13
0xb534: push    r12
0xb536: push    rbp
0xb537: push    rbx
0xb538: sub     rsp, 0x58
0xb53c: mov     rax, qword ptr fs:[0x28]
0xb545: mov     qword ptr [rsp + 0x48], rax
0xb54a: xor     eax, eax
0xb54c: lea     rdx, [rsp + 0x18]
0xb551: mov     dword ptr [rsp + 0x10], 0
0xb559: mov     dword ptr [rsp + 0x14], 0
0xb561: call    0x4a70
0xb566: test    eax, eax
0xb568: jne     0xb5a0
0xb56a: mov     ebx, eax
0xb56c: movzx   eax, word ptr [rsp + 0x1a]
0xb571: test    ax, ax
0xb574: je      0xb790
0xb57a: movzx   ebp, word ptr [rsp + 0x18]
0xb57f: test    bp, bp
0xb582: je      0xb790
0xb588: movzx   ebx, ax
0xb58b: lea     r12d, [rbp - 1]
0xb58f: lea     eax, [rbx - 1]
0xb592: mov     dword ptr [rsp + 0xc], eax
0xb596: jmp     0xb5b2
0xb5b2:
0xb598: nop     dword ptr [rax + rax]
0xb5a0: mov     dword ptr [rsp + 0xc], 0xffffffff
0xb5a8: mov     r12d, 0xffffffff
0xb5ae: xor     ebx, ebx
0xb5b0: xor     ebp, ebp
0xb5b2: lea     r13, [rip + 0x15667]
0xb5b9: lea     r15, [rip + 0x15a13]
0xb5c0: mov     rdi, r13
0xb5c3: call    0x47e0
0xb5c8: mov     rdi, rax
0xb5cb: test    rax, rax
0xb5ce: je      0xb5df
0xb5d0: lea     rdx, [rsp + 0x14]
0xb5d5: mov     rsi, r15
0xb5d8: xor     eax, eax
0xb5da: call    0x4c60

```

```

0xb5df: lea    r14, [rip + 0x15642]
0xb5e6: mov     rdi, r14
0xb5e9: call    0x47e0
0xb5ee: mov     rdi, rax
0xb5f1: test    rax, rax
0xb5f4: je      0xb605
0xb5f6: lea     rdx, [rsp + 0x10]
0xb5fb: mov     rsi, r15
0xb5fe: xor     eax, eax
0xb600: call    0x4c60
0xb605: cmp     dword ptr [rsp + 0xc], 0x3ff
0xb60d: mov     eax, dword ptr [rip + 0x1e8fd]
0xb613: jbe     0xb628
0xb615: mov     ebx, dword ptr [rsp + 0x14]
0xb619: lea     edx, [rbx - 1]
0xb61c: cmp     edx, 0x3ff
0xb622: ja      0xb7c0
0xb628: mov     dword ptr [rip + 0x1f36a], ebx
0xb62e: cmp     r12d, 0x3ff
0xb635: jbe     0xb64a
0xb637: mov     ebp, dword ptr [rsp + 0x10]
0xb63b: lea     edx, [rbp - 1]
0xb63e: cmp     edx, 0x3ff
0xb644: ja      0xb7a8
0xb64a: lea     r12, [rsp + 0x20]
0xb64f: mov     r8d, ebp
0xb652: mov     rcx, r15
0xb655: mov     edx, 0x20
0xb65a: mov     esi, 1
0xb65f: mov     rdi, r12
0xb662: xor     eax, eax
0xb664: mov     dword ptr [rip + 0x1f332], ebp
0xb66a: call    0x4f10
0xb66f: mov     rsi, r12
0xb672: mov     rdi, r14
0xb675: call    0x18380
0xb67a: mov     rcx, r15
0xb67d: mov     edx, 0x20
0xb682: mov     rdi, r12
0xb685: mov     r8d, dword ptr [rip + 0x1f30c]
0xb68c: mov     esi, 1
0xb691: xor     eax, eax
0xb693: call    0x4f10
0xb698: mov     rdi, r13
0xb69b: mov     rsi, r12
0xb69e: call    0x18380
0xb6a3: mov     rdi, qword ptr [rip + 0x1fc8e]
0xb6aa: test    rdi, rdi

```

```

0xb6ad: je      0xb6b4
0xb6af: call    0x18d10
0xb6b4: mov     rdi, qword ptr [rip + 0x1fc75]
0xb6bb: test    rdi, rdi
0xb6be: je      0xb6c5
0xb6c0: call    0x18d10
0xb6c5: mov     rdi, qword ptr [rip + 0x1fc5c]
0xb6cc: test    rdi, rdi
0xb6cf: je      0xb6d6
0xb6d1: call    0x18d10
0xb6d6: mov     rdi, qword ptr [rip + 0x1fc43]
0xb6dd: test    rdi, rdi
0xb6e0: je      0xb6e7
0xb6e2: call    0x18d10
0xb6e7: mov     edi, dword ptr [rip + 0x1f2ab]
0xb6ed: imul    edi, dword ptr [rip + 0x1f2a8]
0xb6f4: mov     esi, 1
0xb6f9: movsxd  rdi, edi
0xb6fc: call    0x1d740
0xb701: mov     edi, dword ptr [rip + 0x1f291]
0xb707: mov     esi, 1
0xb70c: imul    edi, dword ptr [rip + 0x1f289]
0xb713: mov     qword ptr [rip + 0x1fc1e], rax
0xb71a: movsxd  rdi, edi
0xb71d: call    0x1d740
0xb722: mov     edi, dword ptr [rip + 0x1f270]
0xb728: mov     esi, 1
0xb72d: imul    edi, dword ptr [rip + 0x1f268]
0xb734: mov     qword ptr [rip + 0x1fbf5], rax
0xb73b: movsxd  rdi, edi
0xb73e: call    0x1d740
0xb743: mov     edi, dword ptr [rip + 0x1f24f]
0xb749: mov     esi, 1
0xb74e: imul    edi, dword ptr [rip + 0x1f247]
0xb755: mov     qword ptr [rip + 0x1fbcc], rax
0xb75c: movsxd  rdi, edi
0xb75f: call    0x1d740
0xb764: mov     qword ptr [rip + 0x1fbb5], rax
0xb76b: mov     rax, qword ptr [rsp + 0x48]
0xb770: sub     rax, qword ptr fs:[0x28]
0xb779: jne     0xb7dc
0xb77b: add     rsp, 0x58
0xb77f: pop     rbx
0xb780: pop     rbp
0xb781: pop     r12
0xb783: pop     r13
0xb785: pop     r14
0xb787: pop     r15

```



```

0xb789: ret
0xb78a: nop      word ptr [rax + rax]
0xb790: mov      dword ptr [rsp + 0xc], 0xffffffff
0xb798: xor      ebp, ebp
0xb79a: mov      r12d, 0xffffffff
0xb7a0: jmp      0xb5b2
0xb5b2
0xb7a5: nop      dword ptr [rax]
0xb7a8: lea      edx, [rax - 1]
0xb7ab: mov      ebp, 0x18
0xb7b0: cmp      edx, 0x400
0xb7b6: cmovb    ebp, eax
0xb7b9: jmp      0xb64a
0xb64a
0xb7be: nop
0xb7c0: mov      ebx, dword ptr [rip + 0x1e72a]
0xb7c6: lea      edx, [rbx - 1]
0xb7c9: cmp      edx, 0x3ff
0xb7cf: mov      edx, 0x50
0xb7d4: cmova    ebx, edx
0xb7d7: jmp      0xb628
0xb628
0xb7dc: call     0x49d0
0xb7e1: nop      word ptr cs:[rax + rax]
0xb7ec: nop      dword ptr [rax]
0xb7f0: endbr64
0xb7f4: push     r15
0xb7f6: push     r14
0xb7f8: push     r13
0xb7fa: push     r12
0xb7fc: mov      r12, rdi
0xb7ff: mov      edi, 0x5858
0xb804: push     rbp
0xb805: push     rbx
0xb806: mov      ebx, esi
0xb808: sub      rsp, 0x38
0xb80c: mov      rax, qword ptr fs:[0x28]
0xb815: mov      qword ptr [rsp + 0x28], rax
0xb81a: movabs   rax, 0x7363762f7665642f
0xb824: mov      word ptr [rsp + 8], di
0xb829: mov      byte ptr [rsp + 0xa], 0
0xb82e: mov      dword ptr [rsp + 0x18], 0x585861
0xb836: mov      qword ptr [rsp], rax
0xb83a: mov      qword ptr [rsp + 0x10], rax
0xb83f: cmp      esi, 1
0xb842: je       0xb978
0xb848: mov      esi, dword ptr [rip + 0x1fbb2]
0xb84e: mov      r13, qword ptr [rip + 0x1d783]

```

```

0xb855: mov     r14d, 1
0xb85b: test    esi, esi
0xb85d: jne     0xb90c
0xb863: mov     rax, qword ptr [rip + 0x1f126]
0xb86a: lea     r15, [rsp + 0x10]
0xb86f: mov     esi, r14d
0xb872: mov     dword ptr [rip + 0x1fb84], 0
0xb87c: mov     rdi, r15
0xb87f: movzx   edx, byte ptr [rax + 8]
0xb883: mov     byte ptr [rsp + 0x19], dl
0xb887: movzx   eax, byte ptr [rax + 9]
0xb88b: mov     byte ptr [rsp + 0x1a], al
0xb88f: call    0x4930
0xb894: mov     ebp, eax
0xb896: cmp     eax, -1
0xb899: je      0xb8f0
0xb89b: mov     eax, dword ptr [rip + 0x1f0fb]
0xb8a1: imul    eax, dword ptr [rip + 0x1f0f0]
0xb8a8: mov     edi, ebp
0xb8aa: mov     rsi, r12
0xb8ad: lea     edx, [rax + rax + 4]
0xb8b1: xor     eax, eax
0xb8b3: call    r13
0xb8b6: mov     edi, ebp
0xb8b8: call    0x4ab0
0xb8bd: cmp     ebx, 2
0xb8c0: je      0xb990
0xb8c6: mov     eax, 1
0xb8cb: mov     rdx, qword ptr [rsp + 0x28]
0xb8d0: sub     rdx, qword ptr fs:[0x28]
0xb8d9: jne     0xbb07
0xb8df: add     rsp, 0x38
0xb8e3: pop     rbx
0xb8e4: pop     rbp
0xb8e5: pop     r12
0xb8e7: pop     r13
0xb8e9: pop     r14
0xb8eb: pop     r15
0xb8ed: ret
0xb8ee: nop
0xb8f0: mov     ecx, 0x30
0xb8f5: mov     esi, r14d
0xb8f8: mov     rdi, r15
0xb8fb: mov     word ptr [rsp + 0x19], cx
0xb900: call    0x4930
0xb905: mov     ebp, eax
0xb907: cmp     eax, -1
0xb90a: jne     0xb89b

```

```

0xb90c: mov     dword ptr [rip + 0x1faea], 1
0xb916: cmp     ebx, 2
0xb919: je      0xba30
0xb91f: mov     rax, qword ptr [rip + 0x1f06a]
0xb926: mov     r15, rsp
0xb929: mov     esi, r14d
0xb92c: mov     rdi, r15
0xb92f: movzx   edx, byte ptr [rax + 8]
0xb933: movzx   eax, byte ptr [rax + 9]
0xb937: mov     byte ptr [rsp + 8], dl
0xb93b: mov     byte ptr [rsp + 9], al
0xb93f: call    0x4930
0xb944: mov     ebp, eax
0xb946: cmp     eax, -1
0xb949: je      0xba00
0xb94f: mov     edx, dword ptr [rip + 0x1f043]
0xb955: mov     edi, ebp
0xb957: mov     rsi, r12
0xb95a: xor     eax, eax
0xb95c: imul    edx, dword ptr [rip + 0x1f039]
0xb963: add     edx, 4
0xb966: call    r13
0xb969: mov     edi, ebp
0xb96b: call    0x4ab0
0xb970: jmp     0xb8c6
0xb8c6:
0xb975: nop     dword ptr [rax]
0xb978: mov     r13, qword ptr [rip + 0x1d661]
0xb97f: xor     r14d, r14d
0xb982: jmp     0xb863
0xb863:
0xb987: nop     word ptr [rax + rax]
0xb990: movsx   esi, byte ptr [r12 + 2]
0xb996: movsx   edi, byte ptr [r12 + 3]
0xb99c: call    0xa7a0
0xb9a1: mov     edx, dword ptr [rip + 0x1f561]
0xb9a7: test    edx, edx
0xb9a9: jle     0xb9e5
0xb9ab: xor     ebp, ebp
0xb9ad: lea     rbx, [rip + 0x1f56c]
0xb9b4: jmp     0xb9cc
0xb9cc:
0xb9b6: nop     word ptr cs:[rax + rax]
0xb9c0: mov     edx, dword ptr [rip + 0x1f542]
0xb9c6: add     ebp, eax
0xb9c8: cmp     edx, ebp
0xb9ca: jle     0xb9e5
0xb9cc: sub     edx, ebp

```

```

0xb9ce: movsxd  rsi, ebp
0xb9d1: mov      edi, 1
0xb9d6: add      rsi, rbx
0xb9d9: movsxd  rdx, edx
0xb9dc: call     0x18da0
0xb9e1: test     eax, eax
0xb9e3: jns      0xb9c0
0xb9e5: mov      dword ptr [rip + 0x1f519], 0
0xb9ef: mov      eax, 1
0xb9f4: jmp      0xb8cb
0xb8cb:
0xb9f9: nop      dword ptr [rax]
0xba00: mov      edx, 0x30
0xba05: mov      esi, r14d
0xba08: mov      rdi, r15
0xba0b: mov      word ptr [rsp + 8], dx
0xba10: call     0x4930
0xba15: mov      ebp, eax
0xba17: cmp      eax, -1
0xba1a: jne      0xb94f
0xba20: xor      eax, eax
0xba22: jmp      0xb8cb
0xb8cb:
0xba27: nop      word ptr [rax + rax]
0xba30: xor      eax, eax
0xba32: mov      r15, rsp
0xba35: call     0xa300
0xba3a: mov      rax, qword ptr [rip + 0x1ef4f]
0xba41: mov      esi, r14d
0xba44: mov      rdi, r15
0xba47: movzx    edx, byte ptr [rax + 8]
0xba4b: movzx    eax, byte ptr [rax + 9]
0xba4f: mov      byte ptr [rsp + 8], dl
0xba53: mov      byte ptr [rsp + 9], al
0xba57: call     0x4930
0xba5c: mov      ebp, eax
0xba5e: cmp      eax, -1
0xba61: je       0xbae0
0xba63: mov      edx, dword ptr [rip + 0x1ef2f]
0xba69: mov      rsi, r12
0xba6c: mov      edi, ebp
0xba6e: xor      eax, eax
0xba70: imul     edx, dword ptr [rip + 0x1ef25]
0xba77: add      edx, 4
0xba7a: call     r13
0xba7d: mov      edi, ebp
0xba7f: call     0x4ab0
0xba84: mov      eax, dword ptr [rip + 0x1ef12]

```

```

0xba8a: xor     esi, esi
0xba8c: lea     edi, [rax - 1]
0xba8f: call    0xa7a0
0xba94: mov     edx, dword ptr [rip + 0x1f46e]
0xba9a: test    edx, edx
0xba9c: jle     0xb9e5
0xbaa2: xor     ebp, ebp
0xbaa4: lea     rbx, [rip + 0x1f475]
0xbaab: jmp     0xbac0
0xbac0:
0xbaad: nop     dword ptr [rax]
0xbab0: mov     edx, dword ptr [rip + 0x1f452]
0xbab6: add     ebp, eax
0xbab8: cmp     edx, ebp
0xbaba: jle     0xb9e5
0xbac0: sub     edx, ebp
0xbac2: movsxd  rsi, ebp
0xbac5: mov     edi, 1
0xbaca: add     rsi, rbx
0xbacd: movsxd  rdx, edx
0xbad0: call    0x18da0
0xbad5: test    eax, eax
0xbad7: jns     0xbab0
0xbad9: jmp     0xb9e5
0xb9e5:
0xbade: nop
0xbae0: mov     eax, 0x30
0xbae5: mov     esi, r14d
0xbae8: mov     rdi, r15
0xbaeb: mov     word ptr [rsp + 8], ax
0xbaf0: call    0x4930
0xbaf5: mov     ebp, eax
0xbaf7: cmp     eax, -1
0xbafa: jne     0xba63
0xbb00: xor     eax, eax
0xbb02: jmp     0xb8cb
0xb8cb:
0xbb07: call    0x49d0
0xbb0c: nop     dword ptr [rax]
0xbb10: endbr64
0xbb14: mov     eax, dword ptr [rip + 0x1ee6e]
0xbb1a: test    eax, eax
0xbb1c: jne     0xbb20
0xbb1e: ret
0xbb1f: nop
0xbb20: sub     rsp, 8
0xbb24: mov     esi, 1
0xbb29: xor     eax, eax

```

```

0xbb2b: call    0xb7f0
0xbb30: mov     dword ptr [rip + 0x1f8ce], eax
0xbb36: add     rsp, 8
0xbb3a: ret
0xbb3b: nop     dword ptr [rax + rax]
0xbb40: endbr64
0xbb44: push    rbp
0xbb45: xor     eax, eax
0xbb47: mov     rbp, rdi
0xbb4a: call    0xa6c0
0xbb4f: mov     edx, dword ptr [rip + 0x1ee33]
0xbb55: test    edx, edx
0xbb57: je      0xbb94
0xbb59: mov     eax, dword ptr [rip + 0x1f8a5]
0xbb5f: test    eax, eax
0xbb61: je      0xbb94
0xbb63: mov     edx, dword ptr [rip + 0x1ee33]
0xbb69: imul    edx, dword ptr [rip + 0x1ee28]
0xbb70: xor     esi, esi
0xbb72: mov     rdi, qword ptr [rip + 0x1f7af]
0xbb79: movsxd  rdx, edx
0xbb7c: call    0x4a50
0xbb81: xor     eax, eax
0xbb83: mov     esi, 2
0xbb88: mov     rdi, rbp
0xbb8b: call    0xb7f0
0xbb90: test    eax, eax
0xbb92: jne     0xbba0
0xbb94: xor     eax, eax
0xbb96: pop     rbp
0xbb97: jmp     0xa300
0xa300:
0xbb9c: nop     dword ptr [rax]
0xbba0: mov     edx, dword ptr [rip + 0x1edf6]
0xbba6: imul    edx, dword ptr [rip + 0x1edeb]
0xbbad: xor     esi, esi
0xbbae: mov     rdi, qword ptr [rip + 0x1f782]
0xbbb6: pop     rbp
0xbbb7: movsxd  rdx, edx
0xbbba: jmp     0x4a50
0x4a50:
0xbbbf: nop
0xbbbc0: endbr64
0xbbbc4: push    rbp
0xbbbc5: mov     rbp, rdi
0xbbbc8: push    rbx
0xbbbc9: sub     rsp, 8
0xbbcd: cmp     dword ptr [rip + 0x1ed84], 1

```

```

0xbbd4: je      0xbcd0
0xbbda: xor      eax, eax
0xbddc: call     0xa6c0
0xbbe1: test     rbp, rbp
0xbbe4: je      0xbbf4
0xbbe6: mov      eax, dword ptr [rip + 0x1ed9c]
0xbbec: test     eax, eax
0xbbee: jne      0xbcc0
0xbbf4: mov      eax, dword ptr [rip + 0x1eda2]
0xbbfa: mov      rdi, qword ptr [rip + 0x1e347]
0xbc01: lea      rdx, [rip - 0x2198]
0xbc08: lea      esi, [rax - 1]
0xbc0b: call     0x4820
0xbc10: mov      edx, dword ptr [rip + 0x1f2f2]
0xbc16: test     edx, edx
0xbc18: jle      0xbc4d
0xbc1a: xor      ebp, ebp
0xbc1c: lea      rbx, [rip + 0x1f2fd]
0xbc23: jmp      0xbc34
0xbc34:
0xbc25: nop      dword ptr [rax]
0xbc28: mov      edx, dword ptr [rip + 0x1f2da]
0xbc2e: add      ebp, eax
0xbc30: cmp      edx, ebp
0xbc32: jle      0xbc4d
0xbc34: sub      edx, ebp
0xbc36: movsxd   rsi, ebp
0xbc39: mov      edi, 1
0xbc3e: add      rsi, rbx
0xbc41: movsxd   rdx, edx
0xbc44: call     0x18da0
0xbc49: test     eax, eax
0xbc4b: jns      0xbc28
0xbc4d: mov      edi, dword ptr [rip + 0x1ed49]
0xbc53: xor      esi, esi
0xbc55: mov      dword ptr [rip + 0x1f2a9], 0
0xbc5f: call     0xa7a0
0xbc64: mov      edx, dword ptr [rip + 0x1f29e]
0xbc6a: test     edx, edx
0xbc6c: jle      0xbca5
0xbc6e: xor      ebp, ebp
0xbc70: lea      rbx, [rip + 0x1f2a9]
0xbc77: jmp      0xbc8c
0xbc8c:
0xbc79: nop      dword ptr [rax]
0xbc80: mov      edx, dword ptr [rip + 0x1f282]
0xbc86: add      ebp, eax
0xbc88: cmp      edx, ebp

```

```

0xbc8a: jle      0xbca5
0xbc8c: sub       edx, ebp
0xbc8e: movsxd   rsi, ebp
0xbc91: mov      edi, 1
0xbc96: add      rsi, rbx
0xbc99: movsxd   rdx, edx
0xbc9c: call     0x18da0
0xbca1: test     eax, eax
0xbca3: jns      0xbc80
0xbca5: mov      dword ptr [rip + 0x1f259], 0
0xbcaf: add      rsp, 8
0xbcb3: mov      edi, 0xa
0xbcb8: pop      rbx
0xbcb9: pop      rbp
0xbcba: jmp      0x4810
0x4810
0xbcbf: nop
0xbcc0: mov      rdi, rbp
0xbcc3: call     0xbb40
0xbcc8: jmp      0xbbf4
0xbbf4
0xbccd: nop      dword ptr [rax]
0xbcd0: lea      rdx, [rip + 0x1f6e9]
0xbcd7: mov      esi, 1
0xbcdc: mov      edi, 1
0xbce1: call     0x4d70
0xbce6: mov      dword ptr [rip + 0x1ec68], 0
0xbcf0: jmp      0xbbda
0xbbda
0xbcf5: nop      word ptr cs:[rax + rax]
0xbd00: endbr64
0xbd04: push     r13
0xbd06: lea      r13, [rip + 0x1cc33]
0xbd0d: push     r12
0xbd0f: push     rbp
0xbd10: mov      rbp, rdi
0xbd13: lea      rdi, [rip + 0x14f14]
0xbd1a: push     rbx
0xbd1b: xor      ebx, ebx
0xbd1d: sub      rsp, 8
0xbd21: jmp      0xbd2d
0xbd2d
0xbd23: nop      dword ptr [rax + rax]
0xbd28: mov      rdi, qword ptr [r13 + rbx*8]
0xbd2d: mov      rsi, rbp
0xbd30: mov      r12d, ebx
0xbd33: call     0x4b30
0xbd38: test     eax, eax

```



```

0xbd3a: je      0xbd60
0xbd3c: add     rbx, 1
0xbd40: cmp     rbx, 0xa
0xbd44: jne     0xbd28
0xbd46: add     rsp, 8
0xbd4a: mov     r12d, 0xffffffff
0xbd50: pop     rbx
0xbd51: mov     eax, r12d
0xbd54: pop     rbp
0xbd55: pop     r12
0xbd57: pop     r13
0xbd59: ret
0xbd5a: nop     word ptr [rax + rax]
0xbd60: lea     eax, [rbx - 8]
0xbd63: cmp     ebx, 7
0xbd66: cmovg   r12d, eax
0xbd6a: add     rsp, 8
0xbd6e: pop     rbx
0xbd6f: pop     rbp
0xbd70: mov     eax, r12d
0xbd73: pop     r12
0xbd75: pop     r13
0xbd77: ret
0xbd78: nop     dword ptr [rax + rax]
0xbd80: endbr64
0xbd84: push    r12
0xbd86: mov     r12, rdi
0xbd89: push    rbp
0xbd8a: lea     rbp, [rip + 0x1e1e7]
0xbd91: push    rbx
0xbd92: mov     ebx, 0x11
0xbd97: jmp     0xbdac
0xbdac:
0xbd99: nop     dword ptr [rax]
0xbda0: add     ebx, 1
0xbda3: add     rbp, 0x20
0xbda7: cmp     ebx, 0x26
0xbdaa: je      0xbdd8
0xbdac: mov     rdi, qword ptr [rbp]
0xbdb0: mov     rsi, r12
0xbdb3: call    0x4b30
0xbdb8: test    eax, eax
0xbdba: jne     0xbda0
0xbdbc: movsxd  rbx, ebx
0xbdbf: lea     rax, [rip + 0x1df7a]
0xbdc6: shl     rbx, 5
0xbdca: mov     rax, qword ptr [rax + rbx + 8]
0xbdcf: pop     rbx

```

```

0xbdd0: pop      rbp
0xbdd1: pop      r12
0xbdd3: ret
0xbdd4: nop      dword ptr [rax]
0xbdd8: pop      rbx
0xbdd9: xor      eax, eax
0xbddb: pop      rbp
0xbddc: pop      r12
0xbdde: ret
0xbddf: nop
0xbde0: endbr64
0xbde4: push     r14
0xbde6: lea      rdi, [rip + 0x1479d]
0xbded: push     r13
0xbdef: push     r12
0xbdf1: push     rbp
0xbdf2: push     rbx
0xbdf3: sub      rsp, 0xb0
0xbdfa: mov      rax, qword ptr fs:[0x28]
0xbe03: mov      qword ptr [rsp + 0xa8], rax
0xbe0b: xor      eax, eax
0xbe0d: call     0x47e0
0xbe12: lea      rsi, [rsp + 0x10]
0xbe17: mov      edi, 1
0xbe1c: mov      rbp, rax
0xbe1f: call     0x4eb0
0xbe24: mov      rax, qword ptr [rsp + 0x38]
0xbe29: shr      rax, 8
0xbe2d: cmp      rax, 4
0xbe31: sete     al
0xbe34: movzx    eax, al
0xbe37: mov      dword ptr [rip + 0x1eb4b], eax
0xbe3d: test     rbp, rbp
0xbe40: je       0xc36d
0xbe46: mov      rdi, rbp
0xbe49: call     0x49b0
0xbe4e: cmp      rax, 0x3f
0xbe52: ja       0xc308
0xbe58: lea      r12, [rip + 0x14ff1]
0xbe5f: lea      rbx, [rip + 0x14d6f]
0xbe66: lea      r14, [rip + 0x14e02]
0xbe6d: lea      r13, [rip + 0x14e07]
0xbe74: nop      dword ptr [rax]
0xbe78: xor      edi, edi
0xbe7a: mov      rsi, rbp
0xbe7d: call     0x4e90
0xbe82: cmp      eax, -1
0xbe85: je       0xbf38

```

```

0xbe8b: test    eax, eax
0xbe8d: jne     0xbff8
0xbe93: mov     rdi, qword ptr [rip + 0x1e9e6]
0xbe9a: mov     esi, 1
0xbe9f: mov     r8, rbp
0xbea2: mov     rdx, r12
0xbea5: mov     rcx, qword ptr [rip + 0x1ea3c]
0xbeac: lea     r9, [rip + 0x1de5d]
0xeb3:  call   0x4e60
0xeb8:  mov     rsi, rbx
0xebb:  mov     rdi, rbp
0xebe:  call   0x4b30
0x3:   test    eax, eax
0x5:   je      0xbf98
0x3b:  mov     rdi, rbp
0x3e:  call   0x9940
0x3d:  test    eax, eax
0x35:  jne     0xbfc8
0x3db:  mov     rsi, r14
0x3de:  mov     rdi, rbp
0x3ee:  call   0x4b30
0x3ee:  test    eax, eax
0x3ee:  je      0xbf01
0x3ee:  lea     rsi, [rip + 0x14d84]
0x3ef:  mov     rdi, rbp
0x3ef:  call   0x4b30
0x3ef:  test    eax, eax
0x3ef:  jne     0xc3b4
0xbf0:  mov     rdi, qword ptr [rip + 0x1e978]
0xbf0:  mov     rcx, qword ptr [rip + 0x1e9d9]
0xbf0:  mov     esi, 1
0xbf1:  xor     eax, eax
0xbf1:  mov     rdx, r13
0xbf1:  lea     rbp, [rip + 0x14cba]
0xbf2:  call   0x4e60
0xbf2:  xor     edi, edi
0xbf2:  mov     rsi, rbp
0xbf2:  call   0x4e90
0xbf2:  cmp     eax, -1
0xbf2:  jne     0xbe8b
0xbf3:  mov     rcx, qword ptr [rip + 0x1e9a9]
0xbf3:  mov     rdi, qword ptr [rip + 0x1e93a]
0xbf4:  xor     eax, eax
0xbf4:  lea     r8, [rip + 0x1ddc1]
0xbf4:  lea     rdx, [rip + 0x14ea2]
0xbf5:  mov     esi, 1
0xbf5:  call   0x4e60
0xbf6:  mov     rdi, qword ptr [rip + 0x1e919]

```

```

0xbf67: xor     eax, eax
0xbf69: mov     rcx, qword ptr [rip + 0x1e978]
0xbf70: lea     r8, [rip + 0x1dd89]
0xbf77: lea     rdx, [rip + 0x14ea2]
0xbf7e: mov     esi, 1
0xbf83: call    0x4e60
0xbf88: mov     edi, 1
0xbf8d: call    0x4e20
0xbf92: nop     word ptr [rax + rax]
0xbf98: mov     rcx, qword ptr [rip + 0x1e949]
0xbf9f: mov     rdi, qword ptr [rip + 0x1e8da]
0xbfa6: lea     rdx, [rip + 0x14c99]
0xbfad: mov     esi, 1
0xbfb2: lea     rbp, [rip + 0x14c21]
0xbfb9: call    0x4e60
0xbfbe: jmp     0xbe78
0xbe78:
0xbfc3: nop     dword ptr [rax + rax]
0xbfc8: mov     rcx, qword ptr [rip + 0x1e919]
0xbfcf: mov     rdi, qword ptr [rip + 0x1e8aa]
0xbfd6: lea     rdx, [rip + 0x14c7d]
0xbfdd: xor     eax, eax
0xbfdf: mov     esi, 1
0xbfe4: lea     rbp, [rip + 0x14bf6]
0xbfeb: call    0x4e60
0xbff0: jmp     0xbe78
0xbe78:
0xbff5: nop     dword ptr [rax]
0xbff8: mov     rdi, rbp
0xbffb: lea     r13, [rsp + 8]
0xc000: call    0x1d7a0
0xc005: mov     edi, 0x800
0xc00a: mov     qword ptr [rip + 0x1e93f], rax
0xc011: call    0x1d670
0xc016: mov     rdi, qword ptr [rip + 0x1de03]
0xc01d: mov     rsi, r13
0xc020: mov     qword ptr [rsp + 8], rax
0xc025: call    0x4ca0
0xc02a: xor     edx, edx
0xc02c: test    rax, rax
0xc02f: je      0xc034
0xc031: movzx   edx, byte ptr [rax]
0xc034: mov     rdi, qword ptr [rip + 0x1de25]
0xc03b: mov     rsi, r13
0xc03e: mov     byte ptr [rip + 0x1e904], dl
0xc044: call    0x4ca0
0xc049: mov     rdi, qword ptr [rip + 0x1ddf0]
0xc050: mov     rsi, r13

```

```

0xc053: mov     qword ptr [rip + 0x1e8e6], rax
0xc05a: call    0x4ca0
0xc05f: cmp     qword ptr [rip + 0x1e8d9], 0
0xc067: mov     qword ptr [rip + 0x1e8ca], rax
0xc06e: je      0xc2c0
0xc074: test    rax, rax
0xc077: je      0xc2c0
0xc07d: mov     rdi, qword ptr [rip + 0x1dcbc]
0xc084: mov     rsi, r13
0xc087: call    0x4ca0
0xc08c: mov     rdi, qword ptr [rip + 0x1dced]
0xc093: mov     rsi, r13
0xc096: mov     qword ptr [rip + 0x1dcab], rax
0xc09d: call    0x4ca0
0xc0a2: mov     rdi, qword ptr [rip + 0x1dcb7]
0xc0a9: mov     rsi, r13
0xc0ac: mov     qword ptr [rip + 0x1dcd5], rax
0xc0b3: call    0x4ca0
0xc0b8: cmp     qword ptr [rip + 0x1dc88], 0
0xc0c0: mov     qword ptr [rip + 0x1dca1], rax
0xc0c7: je      0xc352
0xc0cd: mov     rdi, qword ptr [rip + 0x1ddac]
0xc0d4: mov     rsi, r13
0xc0d7: call    0x4ca0
0xc0dc: xor     edx, edx
0xc0de: mov     qword ptr [rip + 0x1dda3], rax
0xc0e5: test    rax, rax
0xc0e8: je      0xc0ff
0xc0ea: mov     rdi, qword ptr [rip + 0x1ddcf]
0xc0f1: mov     qword ptr [rip + 0x1dc70], rax
0xc0f8: call    0x4c90
0xc0fd: mov     edx, eax
0xc0ff: mov     rdi, qword ptr [rip + 0x1dd9a]
0xc106: mov     dword ptr [rip + 0x1ddc4], edx
0xc10c: call    0x4c50
0xc111: mov     dword ptr [rip + 0x1dd99], eax
0xc117: test    eax, eax
0xc119: js      0xc13c
0xc11b: mov     qword ptr [rip + 0x1dc62], 0
0xc126: mov     qword ptr [rip + 0x1dc37], 0
0xc131: mov     qword ptr [rip + 0x1dc0c], 0
0xc13c: mov     rdi, qword ptr [rip + 0x1dd9d]
0xc143: call    0x4c50
0xc148: mov     rdi, qword ptr [rip + 0x1ddb1]
0xc14f: mov     dword ptr [rip + 0x1dd9b], eax
0xc155: call    0x4c50
0xc15a: mov     rdi, qword ptr [rip + 0x1dc3f]
0xc161: mov     rsi, r13

```

```

0xc164: mov     dword ptr [rip + 0x1dda6], eax
0xc16a: call    0x4ca0
0xc16f: mov     rdi, qword ptr [rip + 0x1dc4a]
0xc176: mov     rsi, r13
0xc179: mov     qword ptr [rip + 0x1dc28], rax
0xc180: call    0x4ca0
0xc185: cmp     qword ptr [rip + 0x1dc1b], 0
0xc18d: mov     qword ptr [rip + 0x1dc34], rax
0xc194: je      0xc2a0
0xc19a: test    rax, rax
0xc19d: je      0xc2a0
0xc1a3: mov     rdi, qword ptr [rip + 0x1dc36]
0xc1aa: mov     rsi, r13
0xc1ad: lea     rbp, [rip + 0x1ddac]
0xc1b4: lea     rbx, [rbp - 0x220]
0xc1bb: call    0x4ca0
0xc1c0: mov     rdi, qword ptr [rip + 0x1dc39]
0xc1c7: mov     rsi, r13
0xc1ca: lea     r12, [rbx + 0x4c0]
0xc1d1: mov     qword ptr [rip + 0x1dc10], rax
0xc1d8: call    0x4ca0
0xc1dd: mov     rdi, qword ptr [rip + 0x1dd3c]
0xc1e4: mov     rsi, r13
0xc1e7: mov     qword ptr [rip + 0x1dc1a], rax
0xc1ee: call    0x4ca0
0xc1f3: mov     rdi, qword ptr [rip + 0x1dd46]
0xc1fa: mov     rsi, r13
0xc1fd: mov     qword ptr [rip + 0x1dd24], rax
0xc204: call    0x4ca0
0xc209: mov     qword ptr [rip + 0x1dd38], rax
0xc210: mov     rdi, qword ptr [rbp]
0xc214: mov     rsi, r13
0xc217: add     rbp, 0x20
0xc21b: call    0x4ca0
0xc220: mov     qword ptr [rbp - 0x18], rax
0xc224: cmp     rbp, r12
0xc227: jne     0xc210
0xc229: xor     ebp, ebp
0xc22b: lea     r13, [rip + 0x161b3]
0xc232: lea     r14, [rip + 0x14c57]
0xc239: jmp     0xc249
0xc249:
0xc23b: nop     dword ptr [rax + rax]
0xc240: add     rbx, 0x20
0xc244: cmp     rbx, r12
0xc247: je      0xc268
0xc249: cmp     qword ptr [rbx + 8], 0
0xc24e: jne     0xc240

```

```

0xc250: mov     eax, dword ptr [rbx + 0x14]
0xc253: test    eax, eax
0xc255: jne     0xc2e0
0xc25b: mov     qword ptr [rbx + 8], r13
0xc25f: add     rbx, 0x20
0xc263: cmp     rbx, r12
0xc266: jne     0xc249
0xc268: test    ebp, ebp
0xc26a: jne     0xc390
0xc270: mov     rax, qword ptr [rsp + 0xa8]
0xc278: sub     rax, qword ptr fs:[0x28]
0xc281: jne     0xc3be
0xc287: add     rsp, 0xb0
0xc28e: pop     rbx
0xc28f: pop     rbp
0xc290: pop     r12
0xc292: pop     r13
0xc294: pop     r14
0xc296: ret
0xc297: nop     word ptr [rax + rax]
0xc2a0: mov     qword ptr [rip + 0x1dafd], 0
0xc2ab: mov     qword ptr [rip + 0x1db12], 0
0xc2b6: jmp     0xc1a3
0xc1a3:
0xc2bb: nop     dword ptr [rax + rax]
0xc2c0: mov     qword ptr [rip + 0x1e66d], 0
0xc2cb: mov     qword ptr [rip + 0x1e66a], 0
0xc2d6: jmp     0xc07d
0xc07d:
0xc2db: nop     dword ptr [rax + rax]
0xc2e0: mov     r8, qword ptr [rbx]
0xc2e3: mov     rcx, qword ptr [rip + 0x1e5fe]
0xc2ea: mov     rdx, r14
0xc2ed: xor     eax, eax
0xc2ef: mov     rdi, qword ptr [rip + 0x1e58a]
0xc2f6: mov     esi, 1
0xc2fb: add     ebp, 1
0xc2fe: call    0x4e60
0xc303: jmp     0xc240
0xc240:
0xc308: mov     rcx, qword ptr [rip + 0x1e5d9]
0xc30f: mov     rdi, qword ptr [rip + 0x1e56a]
0xc316: lea     rdx, [rip + 0x14aab]
0xc31d: xor     eax, eax
0xc31f: mov     esi, 1
0xc324: call    0x4e60
0xc329: mov     rcx, qword ptr [rip + 0x1e550]
0xc330: mov     edx, 0x11

```

```

0xc335: mov     esi, 1
0xc33a: lea     rdi, [rip + 0x148f3]
0xc341: lea     rbp, [rip + 0x1d9d1]
0xc348: call    0x4e40
0xc34d: jmp     0xbe58
0xbe58
0xc352: mov     qword ptr [rip + 0x1da2b], 0
0xc35d: mov     qword ptr [rip + 0x1da00], 0
0xc368: jmp     0xc0cd
0xc0cd
0xc36d: mov     rcx, qword ptr [rip + 0x1e574]
0xc374: mov     rdi, qword ptr [rip + 0x1e505]
0xc37b: lea     rdx, [rip + 0x14a16]
0xc382: xor     eax, eax
0xc384: mov     esi, 1
0xc389: call    0x4e60
0xc38e: jmp     0xc329
0xc329
0xc390: mov     rcx, qword ptr [rip + 0x1e551]
0xc397: mov     rdi, qword ptr [rip + 0x1e4e2]
0xc39e: mov     r8d, ebp
0xc3a1: xor     eax, eax
0xc3a3: lea     rdx, [rip + 0x14b16]
0xc3aa: mov     esi, 1
0xc3af: call    0x4e60
0xc3b4: mov     edi, 1
0xc3b9: call    0x4e20
0xc3be: call    0x49d0
0xc3c3: nop     word ptr cs:[rax + rax]
0xc3ce: nop
0xc3d0: endbr64
0xc3d4: push    rbx
0xc3d5: mov     ebx, edi
0xc3d7: xor     edi, edi
0xc3d9: call    0x48f0
0xc3de: test    eax, eax
0xc3e0: je      0xc453
0xc3e2: mov     edi, 1
0xc3e7: call    0x48f0
0xc3ec: test    eax, eax
0xc3ee: je      0xc453
0xc3f0: mov     edi, 1
0xc3f5: call    0x4cd0
0xc3fa: mov     qword ptr [rip + 0x1e58f], rax
0xc401: test    rax, rax
0xc404: je      0xc47e
0xc406: lea     rsi, [rip + 0x1efb3]
0xc40d: mov     edi, 1

```



```

0xc412: call    0x4d60
0xc417: mov     edi, 0x40
0xc41c: call    0x1d670
0xc421: mov     dword ptr [rip + 0x1efd5], ebx
0xc427: pop     rbx
0xc428: mov     qword ptr [rip + 0x1ef31], rax
0xc42f: mov     qword ptr [rip + 0x1e52a], rax
0xc436: xor     eax, eax
0xc438: mov     qword ptr [rip + 0x1e52d], 0
0xc443: mov     qword ptr [rip + 0x1e51a], 0
0xc44e: jmp     0xbde0
0xbde0
0xc453: mov     rdi, qword ptr [rip + 0x1e426]
0xc45a: mov     rcx, qword ptr [rip + 0x1e487]
0xc461: lea     rdx, [rip + 0x14a88]
0xc468: xor     eax, eax
0xc46a: mov     esi, 1
0xc46f: call    0x4e60
0xc474: mov     edi, 1
0xc479: call    0x4e20
0xc47e: mov     rdi, qword ptr [rip + 0x1e3fb]
0xc485: mov     rcx, qword ptr [rip + 0x1e45c]
0xc48c: mov     esi, 1
0xc491: lea     rdx, [rip + 0x147f8]
0xc498: call    0x4e60
0xc49d: mov     edi, 1
0xc4a2: call    0x4e20
0xc4a7: nop     word ptr [rax + rax]
0xc4b0: endbr64
0xc4b4: push    r12
0xc4b6: push    rbp
0xc4b7: mov     rbp, rdi
0xc4ba: mov     rdi, qword ptr [rip + 0x1e48f]
0xc4c1: push    rbx
0xc4c2: call    0x9940
0xc4c7: test    eax, eax
0xc4c9: jne     0xc4d0
0xc4cb: pop     rbx
0xc4cc: pop     rbp
0xc4cd: pop     r12
0xc4cf: ret
0xc4d0: mov     rdi, rbp
0xc4d3: call    0x49b0
0xc4d8: lea     rdi, [rax + 0x81]
0xc4df: mov     rbx, rax
0xc4e2: call    0x1d670
0xc4e7: mov     rdi, rbp
0xc4ea: mov     r12, rax

```

```

0xc4ed: call    0x1d7a0
0xc4f2: mov     rsi, rbx
0xc4f5: add     ebx, 0x7f
0xc4f8: mov     rbp, rax
0xc4fb: mov     rdi, rax
0xc4fe: call    0x184d0
0xc503: push    7
0xc505: lea     rcx, [rip + 0x147a2]
0xc50c: xor     eax, eax
0xc50e: push    rbp
0xc50f: mov     rdx, -1
0xc516: mov     esi, 1
0xc51b: mov     rdi, r12
0xc51e: lea     r9, [rip + 0x14040]
0xc525: mov     r8d, 0x1b
0xc52b: call    0x4f10
0xc530: mov     eax, dword ptr [rip + 0x1e462]
0xc536: pop     rdx
0xc537: pop     rcx
0xc538: add     eax, eax
0xc53a: cmp     ebx, eax
0xc53c: jl      0xc555
0xc53e: cdqe
0xc540: mov     byte ptr [r12 + rax], 7
0xc545: mov     eax, dword ptr [rip + 0x1e44d]
0xc54b: add     eax, eax
0xc54d: cdqe
0xc54f: mov     byte ptr [r12 + rax + 1], 0
0xc555: mov     rdi, r12
0xc558: call    0x49b0
0xc55d: mov     rsi, r12
0xc560: mov     edi, 1
0xc565: mov     rdx, rax
0xc568: call    0x18da0
0xc56d: mov     rdi, rbp
0xc570: call    0x18d10
0xc575: mov     rdi, r12
0xc578: call    0x18d10
0xc57d: mov     rdi, qword ptr [rip + 0x1e2bc]
0xc584: pop     rbx
0xc585: pop     rbp
0xc586: pop     r12
0xc588: jmp     0x4c40
0x4c40:
0xc58d: nop     dword ptr [rax]
0xc590: endbr64
0xc594: sub     rsp, 8
0xc598: mov     edi, 0x18

```

```

0xc59d: call    0x1d670
0xc5a2: mov     qword ptr [rax], 0
0xc5a9: mov     qword ptr [rax + 8], 0
0xc5b1: add     rsp, 8
0xc5b5: ret
0xc5b6: nop     word ptr cs:[rax + rax]
0xc5c0: endbr64
0xc5c4: test    rdi, rdi
0xc5c7: je      0xc5d0
0xc5c9: jmp     0x18d10
0x18d10
0xc5ce: nop
0xc5d0: ret
0xc5d1: nop     word ptr cs:[rax + rax]
0xc5dc: nop     dword ptr [rax]
0xc5e0: endbr64
0xc5e4: mov     dword ptr [rdi], esi
0xc5e6: mov     dword ptr [rdi + 4], edx
0xc5e9: mov     dword ptr [rdi + 8], ecx
0xc5ec: mov     dword ptr [rdi + 0xc], r8d
0xc5f0: ret
0xc5f1: nop     word ptr cs:[rax + rax]
0xc5fc: nop     dword ptr [rax]
0xc600: endbr64
0xc604: mov     rax, rdi
0xc607: mov     rdi, rsi
0xc60a: mov     esi, edx
0xc60c: mov     ecx, dword ptr [rax + 0x10]
0xc60f: lea     r8d, [rcx + rdx]
0xc613: mov     edx, dword ptr [rax + 0xc]
0xc616: mov     dword ptr [rax + 0x10], r8d
0xc61a: cmp     edx, ecx
0xc61c: jle     0xc640
0xc61e: mov     r9d, dword ptr [rax + 8]
0xc622: cmp     dword ptr [rax + 0x14], r9d
0xc626: jge     0xc640
0xc628: cmp     r8d, edx
0xc62b: jle     0xc631
0xc62d: sub     edx, ecx
0xc62f: mov     esi, edx
0xc631: jmp     0xa1a0
0xa1a0
0xc636: nop     word ptr cs:[rax + rax]
0xc640: xor     eax, eax
0xc642: ret
0xc643: nop     word ptr cs:[rax + rax]
0xc64e: nop
0xc650: endbr64

```

```

0xc654: mov     ecx, dword ptr [rdi + 0x10]
0xc657: lea     edx, [rcx + 1]
0xc65a: mov     dword ptr [rdi + 0x10], edx
0xc65d: cmp     edx, dword ptr [rdi + 0xc]
0xc660: jg      0xc66a
0xc662: mov     ecx, dword ptr [rdi + 8]
0xc665: cmp     dword ptr [rdi + 0x14], ecx
0xc668: jl      0xc670
0xc66a: xor     eax, eax
0xc66c: ret
0xc66d: nop     dword ptr [rax]
0xc670: mov     edi, esi
0xc672: jmp     0xa230
0xa230
0xc677: nop     word ptr [rax + rax]
0xc680: endbr64
0xc684: mov     rax, rdi
0xc687: mov     edi, esi
0xc689: mov     dword ptr [rax + 0x10], edx
0xc68c: add     edx, dword ptr [rax]
0xc68e: mov     dword ptr [rax + 0x14], esi
0xc691: add     edi, dword ptr [rax + 4]
0xc694: mov     esi, edx
0xc696: jmp     0xa4b0
0xa4b0
0xc69b: nop     dword ptr [rax + rax]
0xc6a0: endbr64
0xc6a4: mov     eax, dword ptr [rdi]
0xc6a6: ret
0xc6a7: nop     word ptr [rax + rax]
0xc6b0: endbr64
0xc6b4: mov     eax, dword ptr [rdi + 4]
0xc6b7: ret
0xc6b8: nop     dword ptr [rax + rax]
0xc6c0: endbr64
0xc6c4: mov     eax, dword ptr [rdi + 8]
0xc6c7: ret
0xc6c8: nop     dword ptr [rax + rax]
0xc6d0: endbr64
0xc6d4: mov     eax, dword ptr [rdi + 0xc]
0xc6d7: ret
0xc6d8: nop     dword ptr [rax + rax]
0xc6e0: push    r14
0xc6e2: push    r13
0xc6e4: push    r12
0xc6e6: mov     r12, qword ptr [rip + 0x1ed1b]
0xc6ed: push    rbp
0xc6ee: push    rbx

```

```

0xc6ef: mov     r13, qword ptr [r12 + 0x20]
0xc6f4: mov     rbx, qword ptr [r12 + 0x18]
0xc6f9: cmp     r13, rbx
0xc6fc: je      0xc761
0xc6fe: mov     ebp, edi
0xc700: jbe     0xc770
0xc702: test    bpl, 1
0xc706: je      0xc75c
0xc708: mov     rdi, qword ptr [r12 + 0x60]
0xc70d: test    rdi, rdi
0xc710: je      0xc71e
0xc712: call    0x18d10
0xc717: mov     r12, qword ptr [rip + 0x1ecea]
0xc71e: mov     r14, r13
0xc721: sub     r14, rbx
0xc724: lea     rdi, [r14 + 1]
0xc728: call    0x1d670
0xc72d: mov     rdx, r14
0xc730: mov     qword ptr [r12 + 0x60], rax
0xc735: mov     rax, qword ptr [rip + 0x1eccc]
0xc73c: mov     rsi, qword ptr [rax + 0x58]
0xc740: mov     rdi, qword ptr [rax + 0x60]
0xc744: add     rsi, rbx
0xc747: call    0x4bb0
0xc74c: mov     rax, qword ptr [rip + 0x1ecb5]
0xc753: mov     rax, qword ptr [rax + 0x60]
0xc757: mov     byte ptr [rax + r14], 0
0xc75c: and     ebp, 2
0xc75f: jne     0xc788
0xc761: pop     rbx
0xc762: pop     rbp
0xc763: pop     r12
0xc765: pop     r13
0xc767: pop     r14
0xc769: ret
0xc76a: nop     word ptr [rax + rax]
0xc770: mov     rax, rbx
0xc773: mov     rbx, r13
0xc776: mov     r13, rax
0xc779: test    bpl, 1
0xc77d: je      0xc75c
0xc77f: jmp     0xc708
0xc708:
0xc781: nop     dword ptr [rax]
0xc788: mov     rax, qword ptr [rip + 0x1ec79]
0xc78f: mov     rdx, rbx
0xc792: sub     rdx, r13
0xc795: add     qword ptr [rax + 0x48], rdx

```

```

0xc799: mov     rdi, qword ptr [rax + 0x58]
0xc79d: add     qword ptr [rax + 0x38], rdx
0xc7a1: mov     qword ptr [rax + 0x18], rbx
0xc7a5: lea     rsi, [rdi + r13]
0xc7a9: add     rdi, rbx
0xc7ac: mov     qword ptr [rax + 0x20], rbx
0xc7b0: call    0x48b0
0xc7b5: mov     rbx, qword ptr [rip + 0x1ec4c]
0xc7bc: mov     rax, qword ptr [rbx + 0x38]
0xc7c0: mov     rdi, qword ptr [rbx + 0x58]
0xc7c4: lea     rsi, [rax + 1]
0xc7c8: mov     qword ptr [rbx + 0x50], rsi
0xc7cc: call    0x1d690
0xc7d1: mov     qword ptr [rbx + 0x58], rax
0xc7d5: pop     rbx
0xc7d6: pop     rbp
0xc7d7: pop     r12
0xc7d9: pop     r13
0xc7db: pop     r14
0xc7dd: ret
0xc7de: nop
0xc7e0: endbr64
0xc7e4: test    rdi, rdi
0xc7e7: je      0xc818
0xc7e9: push    rbp
0xc7ea: mov     rbp, rdi
0xc7ed: mov     rdi, qword ptr [rdi + 0x58]
0xc7f1: test    rdi, rdi
0xc7f4: je      0xc7fb
0xc7f6: call    0x18d10
0xc7fb: mov     rdi, qword ptr [rbp + 0x60]
0xc7ff: test    rdi, rdi
0xc802: je      0xc809
0xc804: call    0x18d10
0xc809: mov     rdi, rbp
0xc80c: pop     rbp
0xc80d: jmp     0x18d10
0x18d10:
0xc812: nop     word ptr [rax + rax]
0xc818: ret
0xc819: nop     dword ptr [rax]
0xc820: endbr64
0xc824: mov     rax, qword ptr [rip + 0x1ebdd]
0xc82b: push    rbp
0xc82c: mov     rdi, qword ptr [rax + 0x68]
0xc830: xor     eax, eax
0xc832: call    0x1aac0
0xc837: mov     rax, qword ptr [rip + 0x1ebca]

```

```

0xc83e: mov     rdi, qword ptr [rax]
0xc841: call    0xc5c0
0xc846: mov     rbp, qword ptr [rip + 0x1ebbb]
0xc84d: test    rbp, rbp
0xc850: je      0xc876
0xc852: mov     rdi, qword ptr [rbp + 0x58]
0xc856: test    rdi, rdi
0xc859: je      0xc860
0xc85b: call    0x18d10
0xc860: mov     rdi, qword ptr [rbp + 0x60]
0xc864: test    rdi, rdi
0xc867: je      0xc86e
0xc869: call    0x18d10
0xc86e: mov     rdi, rbp
0xc871: call    0x18d10
0xc876: mov     qword ptr [rip + 0x1eb87], 0
0xc881: pop     rbp
0xc882: ret
0xc883: nop     word ptr cs:[rax + rax]
0xc88e: nop
0xc890: endbr64
0xc894: mov     rax, qword ptr [rip + 0x1eb6d]
0xc89b: movsxd  rcx, edi
0xc89e: mov     edx, esi
0xc8a0: mov     r8, rcx
0xc8a3: mov     qword ptr [rax + 0x28], rcx
0xc8a7: movsxd  rcx, esi
0xc8aa: mov     rdi, qword ptr [rax]
0xc8ad: xor     esi, esi
0xc8af: mov     qword ptr [rax + 0x30], rcx
0xc8b3: mov     ecx, 1
0xc8b8: jmp     0xc5e0
0xc5e0
0xc8bd: nop     dword ptr [rax]
0xc8c0: endbr64
0xc8c4: push    r12
0xc8c6: mov     edi, 0x70
0xc8cb: push    rbx
0xc8cc: sub     rsp, 8
0xc8d0: call    0x1d670
0xc8d5: mov     rbx, qword ptr [rip + 0x1eb2c]
0xc8dc: mov     r12, rax
0xc8df: movdqu  xmm0, xmmword ptr [rbx]
0xc8e3: movups  xmmword ptr [rax], xmm0
0xc8e6: movdqu  xmm1, xmmword ptr [rbx + 0x10]
0xc8eb: movups  xmmword ptr [rax + 0x10], xmm1
0xc8ef: movdqu  xmm2, xmmword ptr [rbx + 0x20]
0xc8f4: movups  xmmword ptr [rax + 0x20], xmm2

```

```

0xc8f8: movdqu xmm3, xmmword ptr [rbx + 0x30]
0xc8fd: movups xmmword ptr [rax + 0x30], xmm3
0xc901: movdqu xmm4, xmmword ptr [rbx + 0x40]
0xc906: movups xmmword ptr [rax + 0x40], xmm4
0xc90a: movdqu xmm5, xmmword ptr [rbx + 0x50]
0xc90f: movups xmmword ptr [rax + 0x50], xmm5
0xc913: movdqu xmm6, xmmword ptr [rbx + 0x60]
0xc918: cmp qword ptr [rax + 0x58], 0
0xc91d: movups xmmword ptr [rax + 0x60], xmm6
0xc921: je 0xc94a
0xc923: mov rdi, qword ptr [rax + 0x50]
0xc927: call 0x1d670
0xc92c: mov rdx, qword ptr [r12 + 0x50]
0xc931: mov qword ptr [rbx + 0x58], rax
0xc935: mov rax, qword ptr [rip + 0x1eacc]
0xc93c: mov rsi, qword ptr [r12 + 0x58]
0xc941: mov rdi, qword ptr [rax + 0x58]
0xc945: call 0x4bb0
0xc94a: mov rdi, qword ptr [r12 + 0x60]
0xc94f: test rdi, rdi
0xc952: je 0xc964
0xc954: mov rbx, qword ptr [rip + 0x1eaad]
0xc95b: call 0x1d7a0
0xc960: mov qword ptr [rbx + 0x60], rax
0xc964: add rsp, 8
0xc968: mov rax, r12
0xc96b: pop rbx
0xc96c: pop r12
0xc96e: ret
0xc96f: nop
0xc970: endbr64
0xc974: test rdi, rdi
0xc977: je 0xc9f0
0xc979: push r13
0xc97b: push r12
0xc97d: push rbp
0xc97e: push rbx
0xc97f: mov rbx, rdi
0xc982: sub rsp, 8
0xc986: mov rbp, qword ptr [rip + 0x1ea7b]
0xc98d: mov rdi, qword ptr [rbp + 0x58]
0xc991: mov r13, qword ptr [rbp + 0x28]
0xc995: mov r12, qword ptr [rbp + 0x30]
0xc999: test rdi, rdi
0xc99c: je 0xc9a3
0xc99e: call 0x18d10
0xc9a3: mov rdi, qword ptr [rbp + 0x60]
0xc9a7: test rdi, rdi

```



```

0xc9aa: je      0xc9b1
0xc9ac: call    0x18d10
0xc9b1: mov     rdi, rbp
0xc9b4: call    0x18d10
0xc9b9: movsxd  rax, r13d
0xc9bc: mov     edx, r12d
0xc9bf: movsxd  r12, r12d
0xc9c2: mov     qword ptr [rbx + 0x28], rax
0xc9c6: mov     rdi, qword ptr [rbx]
0xc9c9: mov     r8d, r13d
0xc9cc: mov     ecx, 1
0xc9d1: mov     qword ptr [rbx + 0x30], r12
0xc9d5: xor     esi, esi
0xc9d7: mov     qword ptr [rip + 0x1ea2a], rbx
0xc9de: add     rsp, 8
0xc9e2: pop     rbx
0xc9e3: pop     rbp
0xc9e4: pop     r12
0xc9e6: pop     r13
0xc9e8: jmp     0xc5e0
0xc5e0
0xc9ed: nop     dword ptr [rax]
0xc9f0: ret
0xc9f1: nop     word ptr cs:[rax + rax]
0xc9fc: nop     dword ptr [rax]
0xca00: endbr64
0xca04: mov     rdx, qword ptr [rip + 0x1e9fd]
0xca0b: mov     rax, qword ptr [rdx + 0x18]
0xca0f: sub     rax, qword ptr [rdx + 0x40]
0xca13: ret
0xca14: nop     word ptr cs:[rax + rax]
0xca1f: nop
0xca20: endbr64
0xca24: mov     rax, qword ptr [rip + 0x1e9dd]
0xca2b: mov     r8d, dword ptr [rax + 8]
0xca2f: mov     dword ptr [rax + 8], edi
0xca32: mov     eax, r8d
0xca35: ret
0xca36: nop     word ptr cs:[rax + rax]
0xca40: endbr64
0xca44: mov     rax, qword ptr [rip + 0x1e9bd]
0xca4b: cmp     qword ptr [rax + 0x48], 0
0xca50: sete    al
0xca53: movzx   eax, al
0xca56: ret
0xca57: nop     word ptr [rax + rax]
0xca60: endbr64
0xca64: mov     rax, qword ptr [rip + 0x1e99d]

```

```

0xca6b: mov     rdx, qword ptr [rax + 0x18]
0xca6f: mov     qword ptr [rax + 0x20], rdx
0xca73: ret
0xca74: nop     word ptr cs:[rax + rax]
0xca7f: nop
0xca80: endbr64
0xca84: sub     rsp, 8
0xca88: xor     eax, eax
0xca8a: mov     edi, 3
0xca8f: call    0xc6e0
0xca94: mov     rax, qword ptr [rip + 0x1e96d]
0xca9b: mov     dword ptr [rax + 0x10], 0x18
0xcaa2: add     rsp, 8
0xcaa6: ret
0xcaa7: nop     word ptr [rax + rax]
0xcab0: endbr64
0xcab4: sub     rsp, 8
0xcab8: xor     eax, eax
0xcaba: mov     edi, 1
0xcabf: call    0xc6e0
0xcac4: mov     rax, qword ptr [rip + 0x1e93d]
0xcacb: mov     dword ptr [rax + 0x10], 0x19
0xcad2: add     rsp, 8
0xcad6: ret
0xcad7: nop     word ptr [rax + rax]
0xcae0: endbr64
0xcae4: mov     rax, qword ptr [rip + 0x1e91d]
0xcaeb: mov     rdx, qword ptr [rax + 0x18]
0xcaef: mov     rcx, qword ptr [rax + 0x20]
0xcaf3: movsxd  rdx, edx
0xcaf6: mov     qword ptr [rax + 0x18], rcx
0xcafa: mov     qword ptr [rax + 0x20], rdx
0xcafe: ret
0xcaff: nop
0xcb00: endbr64
0xcb04: mov     rax, qword ptr [rip + 0x1e8fd]
0xcb0b: mov     rdx, qword ptr [rax + 0x18]
0xcb0f: cmp     rdx, qword ptr [rax + 0x40]
0xcb13: jbe     0xcb24
0xcb15: sub     rdx, 1
0xcb19: mov     dword ptr [rax + 0x10], 1
0xcb20: mov     qword ptr [rax + 0x18], rdx
0xcb24: ret
0xcb25: nop     word ptr cs:[rax + rax]
0xcb30: endbr64
0xcb34: mov     rax, qword ptr [rip + 0x1e8cd]
0xcb3b: mov     rdx, qword ptr [rax + 0x18]
0xcb3f: cmp     rdx, qword ptr [rax + 0x38]

```

```

0xcb43: jae      0xcb54
0xcb45: add      rdx, 1
0xcb49: mov      dword ptr [rax + 0x10], 2
0xcb50: mov      qword ptr [rax + 0x18], rdx
0xcb54: ret
0xcb55: nop      word ptr cs:[rax + rax]
0xcb60: endbr64
0xcb64: mov      rdx, qword ptr [rip + 0x1e89d]
0xcb6b: mov      rax, qword ptr [rdx + 0x18]
0xcb6f: mov      rdi, qword ptr [rdx + 0x40]
0xcb73: cmp      rax, rdi
0xcb76: jbe      0xcc20
0xcb7c: mov      r8, qword ptr [rdx + 0x58]
0xcb80: jmp      0xcba9
0xcba9:
0xcb82: nop      word ptr [rax + rax]
0xcb88: lea      ecx, [rsi - 0x30]
0xcb8b: cmp      cl, 9
0xcb8e: jbe      0xcbbf
0xcb90: sub      esi, 0x24
0xcb93: cmp      sil, 1
0xcb97: jbe      0xcbbf
0xcb99: mov      qword ptr [rdx + 0x18], rax
0xcb9d: mov      dword ptr [rdx + 0x10], 1
0xcba4: cmp      rdi, rax
0xcba7: jae      0xcbef
0xcba9: sub      rax, 1
0xcbad: movzx    esi, byte ptr [r8 + rax]
0xcbb2: mov      ecx, esi
0xcbb4: and      ecx, 0xffffffff
0xcbb7: sub      ecx, 0x41
0xcbb8: cmp      cl, 0x19
0xcbbd: ja      0xcb88
0xcbbf: mov      rax, qword ptr [rdx + 0x18]
0xcbc3: cmp      rdi, rax
0xcbc6: jae      0xcbef
0xcbc8: sub      rax, 1
0xcbcc: movzx    esi, byte ptr [r8 + rax]
0xcbd1: mov      ecx, esi
0xcbd3: and      ecx, 0xffffffff
0xcbd6: sub      ecx, 0x41
0xcbd9: cmp      cl, 0x19
0xcbd9: jbe      0xcc00
0xcbd9: jbe      0xcc00
0xcbde: lea      ecx, [rsi - 0x30]
0xcbe1: cmp      cl, 9
0xcbe4: jbe      0xcc00
0xcbe6: sub      esi, 0x24
0xcbe9: cmp      sil, 1

```

```

0xcbed: jbe      0xcc00
0xcbef: mov      dword ptr [rdx + 0x10], 3
0xcbf6: ret
0xcbf7: nop      word ptr [rax + rax]
0xcc00: mov      qword ptr [rdx + 0x18], rax
0xcc04: mov      dword ptr [rdx + 0x10], 1
0xcc0b: cmp      rdi, rax
0xcc0e: jb       0xcbc8
0xcc10: mov      dword ptr [rdx + 0x10], 3
0xcc17: ret
0xcc18: nop      dword ptr [rax + rax]
0xcc20: ret
0xcc21: nop      word ptr cs:[rax + rax]
0xcc2c: nop      dword ptr [rax]
0xcc30: endbr64
0xcc34: mov      rdx, qword ptr [rip + 0x1e7cd]
0xcc3b: mov      rax, qword ptr [rdx + 0x18]
0xcc3f: mov      rdi, qword ptr [rdx + 0x38]
0xcc43: cmp      rax, rdi
0xcc46: jae      0xcce8
0xcc4c: mov      r8, qword ptr [rdx + 0x58]
0xcc50: jmp      0xcc7d
0xcc7d:
0xcc52: nop      word ptr [rax + rax]
0xcc58: lea      ecx, [rsi - 0x30]
0xcc5b: cmp      cl, 9
0xcc5e: jbe      0xcc8f
0xcc60: sub      esi, 0x24
0xcc63: cmp      sil, 1
0xcc67: jbe      0xcc8f
0xcc69: add      rax, 1
0xcc6d: mov      dword ptr [rdx + 0x10], 2
0xcc74: mov      qword ptr [rdx + 0x18], rax
0xcc78: cmp      rdi, rax
0xcc7b: je       0xccbb
0xcc7d: movzx    esi, byte ptr [r8 + rax]
0xcc82: mov      ecx, esi
0xcc84: and      ecx, 0xffffffff
0xcc87: sub      ecx, 0x41
0xcc8a: cmp      cl, 0x19
0xcc8d: ja       0xcc58
0xcc8f: mov      rax, qword ptr [rdx + 0x18]
0xcc93: cmp      rdi, rax
0xcc96: jbe      0xccbb
0xcc98: movzx    esi, byte ptr [r8 + rax]
0xcc9d: mov      ecx, esi
0xcc9f: and      ecx, 0xffffffff
0xccea: sub      ecx, 0x41

```

```

0xcca5: cmp     cl, 0x19
0xcca8: jbe     0xccc8
0xccaa: lea     ecx, [rsi - 0x30]
0xccad: cmp     cl, 9
0xccb0: jbe     0xccc8
0xccb2: sub     esi, 0x24
0xccb5: cmp     sil, 1
0xccb9: jbe     0xccc8
0xccbb: mov     dword ptr [rdx + 0x10], 4
0xccc2: ret
0xccc3: nop     dword ptr [rax + rax]
0xccc8: add     rax, 1
0xcccc: mov     dword ptr [rdx + 0x10], 2
0xcd3: mov     qword ptr [rdx + 0x18], rax
0xcd7: cmp     rdi, rax
0xcda: jne     0xcc98
0xcdc: mov     dword ptr [rdx + 0x10], 4
0xcce3: ret
0xcce4: nop     dword ptr [rax]
0xcce8: ret
0xcce9: nop     dword ptr [rax]
0xccf0: endbr64
0xccf4: mov     rax, qword ptr [rip + 0x1e70d]
0xccfb: mov     rdx, qword ptr [rax + 0x40]
0xccff: mov     dword ptr [rax + 0x10], 5
0xcd06: mov     qword ptr [rax + 0x18], rdx
0xcd0a: ret
0xcd0b: nop     dword ptr [rax + rax]
0xcd10: endbr64
0xcd14: mov     rax, qword ptr [rip + 0x1e6ed]
0xcd1b: mov     rdx, qword ptr [rax + 0x38]
0xcd1f: mov     dword ptr [rax + 0x10], 6
0xcd26: mov     qword ptr [rax + 0x18], rdx
0xcd2a: ret
0xcd2b: nop     dword ptr [rax + rax]
0xcd30: endbr64
0xcd34: push    rbp
0xcd35: push    rbx
0xcd36: mov     ebx, edi
0xcd38: sub     rsp, 8
0xcd3c: call    0x4f00
0xcd41: movsxd  rdx, ebx
0xcd44: mov     rax, qword ptr [rax]
0xcd47: test    byte ptr [rax + rdx*2 + 1], 0x40
0xcd4c: je      0xcdab
0xcd4e: mov     rbp, qword ptr [rip + 0x1e6b3]
0xcd55: mov     rsi, qword ptr [rbp + 0x38]
0xcd59: lea     rdx, [rsi + 1]

```

```

0xcd5d: cmp     rdx, qword ptr [rbp + 0x50]
0xcd61: jae     0xcdb8
0xcd63: mov     rax, qword ptr [rbp + 0x18]
0xcd67: mov     rcx, qword ptr [rbp + 0x58]
0xcd6b: sub     rdx, rax
0xcd6e: lea     rsi, [rcx + rax]
0xcd72: lea     rdi, [rcx + rax + 1]
0xcd77: call    0x4d40
0xcd7c: mov     rax, qword ptr [rip + 0x1e685]
0xcd83: mov     rdx, qword ptr [rax + 0x58]
0xcd87: mov     rax, qword ptr [rax + 0x18]
0xcd8b: mov     byte ptr [rdx + rax], bl
0xcd8e: mov     rax, qword ptr [rip + 0x1e673]
0xcd95: add     qword ptr [rax + 0x18], 1
0xcd9a: add     qword ptr [rax + 0x38], 1
0xcd9f: add     qword ptr [rax + 0x48], 1
0xcda4: mov     dword ptr [rax + 0x10], 7
0xcdab: add     rsp, 8
0xcdaf: pop     rbx
0xcdb0: pop     rbp
0xcdb1: ret
0xcdb2: nop     word ptr [rax + rax]
0xcdb8: add     rsi, 0x21
0xcdbc: mov     rdi, qword ptr [rbp + 0x58]
0xcdc0: mov     qword ptr [rbp + 0x50], rsi
0xcdc4: call    0x1d690
0xcdc9: mov     qword ptr [rbp + 0x58], rax
0xcdcd: mov     rbp, qword ptr [rip + 0x1e634]
0xcdd4: mov     rax, qword ptr [rbp + 0x38]
0xcdd8: lea     rdx, [rax + 1]
0xcddc: jmp     0xcd63
0xcd63:
0xcdde: nop
0xcde0: endbr64
0xcde4: mov     rax, qword ptr [rip + 0x1e61d]
0xcdeb: mov     rcx, qword ptr [rax + 0x18]
0xcdef: mov     rdx, qword ptr [rax + 0x38]
0xcdf3: cmp     rcx, rdx
0xcdf6: jb     0xce00
0xcdf8: ret
0xcdf9: nop     dword ptr [rax]
0xce00: push    rbx
0xce01: mov     rax, qword ptr [rax + 0x58]
0xce05: add     rdx, 1
0xce09: sub     rdx, rcx
0xce0c: lea     rsi, [rax + rcx + 1]
0xce11: lea     rdi, [rax + rcx]
0xce15: call    0x4bb0

```

```

0xce1a: mov     rbx, qword ptr [rip + 0x1e5e7]
0xce21: mov     rsi, qword ptr [rbx + 0x38]
0xce25: sub     qword ptr [rbx + 0x48], 1
0xce2a: lea     rax, [rsi - 1]
0xce2e: mov     qword ptr [rbx + 0x38], rax
0xce32: test    al, 0xf
0xce34: je      0xce40
0xce36: mov     dword ptr [rbx + 0x10], 8
0xce3d: pop     rbx
0xce3e: ret
0xce3f: nop
0xce40: mov     qword ptr [rbx + 0x50], rsi
0xce44: mov     rdi, qword ptr [rbx + 0x58]
0xce48: call    0x1d690
0xce4d: mov     qword ptr [rbx + 0x58], rax
0xce51: mov     rbx, qword ptr [rip + 0x1e5b0]
0xce58: mov     dword ptr [rbx + 0x10], 8
0xce5f: pop     rbx
0xce60: ret
0xce61: nop     word ptr cs:[rax + rax]
0xce6c: nop     dword ptr [rax]
0xce70: endbr64
0xce74: mov     rax, qword ptr [rip + 0x1e58d]
0xce7b: mov     rcx, qword ptr [rax + 0x18]
0xce7f: cmp     rcx, qword ptr [rax + 0x40]
0xce83: ja      0xce90
0xce85: ret
0xce86: nop     word ptr cs:[rax + rax]
0xce90: push    rbx
0xce91: mov     rdx, qword ptr [rax + 0x38]
0xce95: mov     rsi, qword ptr [rax + 0x58]
0xce99: add     rdx, 1
0xce9d: lea     rdi, [rsi + rcx - 1]
0xcea2: sub     rdx, rcx
0xcea5: add     rsi, rcx
0xcea8: call    0x4bb0
0xcead: mov     rbx, qword ptr [rip + 0x1e554]
0xceb4: mov     rsi, qword ptr [rbx + 0x38]
0xceb8: sub     qword ptr [rbx + 0x18], 1
0xcebd: sub     qword ptr [rbx + 0x48], 1
0xcec2: lea     rax, [rsi - 1]
0xcec6: mov     qword ptr [rbx + 0x38], rax
0xceca: test    al, 0xf
0xcecc: je      0xcee0
0xcece: mov     dword ptr [rbx + 0x10], 9
0xcd5: pop     rbx
0xcd6: ret
0xcd7: nop     word ptr [rax + rax]

```

```

0xcee0: mov     qword ptr [rbx + 0x50], rsi
0xcee4: mov     rdi, qword ptr [rbx + 0x58]
0xcee8: call    0x1d690
0xceed: mov     qword ptr [rbx + 0x58], rax
0xcef1: mov     rbx, qword ptr [rip + 0x1e510]
0xcef8: mov     dword ptr [rbx + 0x10], 9
0xceff: pop     rbx
0xcf00: ret
0xcf01: nop     word ptr cs:[rax + rax]
0xcf0c: nop     dword ptr [rax]
0xcf10: endbr64
0xcf14: push    rbp
0xcf15: push    rbx
0xcf16: sub     rsp, 8
0xcf1a: mov     r9, qword ptr [rip + 0x1e4e7]
0xcf21: mov     rax, qword ptr [r9 + 0x18]
0xcf25: mov     rbx, qword ptr [r9 + 0x20]
0xcf29: mov     qword ptr [r9 + 0x20], rax
0xcf2d: xor     eax, eax
0xcf2f: call    0xcc30
0xcf34: xor     eax, eax
0xcf36: mov     edi, 3
0xcf3b: mov     rbp, qword ptr [r9 + 0x18]
0xcf3f: call    0xc6e0
0xcf44: mov     rax, qword ptr [rip + 0x1e4bd]
0xcf4b: mov     rdx, qword ptr [rax + 0x18]
0xcf4f: cmp     rdx, rbx
0xcf52: jae     0xcf65
0xcf54: lea     rcx, [rdx + rbx]
0xcf58: sub     rcx, rbp
0xcf5b: cmp     rbx, rbp
0xcf5e: cmovae rdx, rcx
0xcf62: mov     rbx, rdx
0xcf65: mov     qword ptr [rax + 0x20], rbx
0xcf69: mov     dword ptr [rax + 0x10], 0xa
0xcf70: add     rsp, 8
0xcf74: pop     rbx
0xcf75: pop     rbp
0xcf76: ret
0xcf77: nop     word ptr [rax + rax]
0xcf80: endbr64
0xcf84: push    rbp
0xcf85: push    rbx
0xcf86: sub     rsp, 8
0xcf8a: mov     rax, qword ptr [rip + 0x1e477]
0xcf91: mov     rbp, qword ptr [rax + 0x18]
0xcf95: mov     rbx, qword ptr [rax + 0x20]
0xcf99: mov     qword ptr [rax + 0x20], rbp

```



```

0xcfd9d: xor     eax, eax
0xcfd9f: call    0xcb60
0xcfa4: xor     eax, eax
0xcfa6: mov     edi, 3
0xcfab: call    0xc6e0
0xcfb0: mov     rax, qword ptr [rip + 0x1e451]
0xcfb7: mov     rdx, qword ptr [rax + 0x18]
0xcfb9: cmp     rdx, rbx
0xcfbe: jae     0xcfd1
0xcfc0: lea     rcx, [rdx + rbx]
0xcfc4: sub     rcx, rbp
0xcfc7: cmp     rbx, rbp
0xcfca: cmovae rdx, rcx
0xcfce: mov     rbx, rdx
0xcfd1: mov     qword ptr [rax + 0x20], rbx
0xcfd5: mov     dword ptr [rax + 0x10], 0xb
0xcfdc: add     rsp, 8
0xcfe0: pop     rbx
0xcfe1: pop     rbp
0xcfe2: ret
0xcfe3: nop     word ptr cs:[rax + rax]
0xcfee: nop
0xcff0: endbr64
0xcff4: push    rbx
0xcff5: mov     rbx, qword ptr [rip + 0x1e40c]
0xcffc: mov     esi, 1
0xd001: mov     qword ptr [rbx + 0x18], 0
0xd009: mov     rdi, qword ptr [rbx + 0x58]
0xd00d: mov     qword ptr [rbx + 0x20], 0
0xd015: mov     qword ptr [rbx + 0x38], 0
0xd01d: mov     qword ptr [rbx + 0x40], 0
0xd025: mov     qword ptr [rbx + 0x48], 0
0xd02d: mov     qword ptr [rbx + 0x50], 1
0xd035: call    0x1d690
0xd03a: mov     qword ptr [rbx + 0x58], rax
0xd03e: mov     rax, qword ptr [rip + 0x1e3c3]
0xd045: mov     rax, qword ptr [rax + 0x58]
0xd049: mov     byte ptr [rax], 0
0xd04c: mov     rax, qword ptr [rip + 0x1e3b5]
0xd053: mov     dword ptr [rax + 0x10], 0xc
0xd05a: pop     rbx
0xd05b: ret
0xd05c: nop     dword ptr [rax]
0xd060: endbr64
0xd064: push    rbx
0xd065: mov     edi, 0x70
0xd06a: sub     rsp, 0x10
0xd06e: mov     rax, qword ptr fs:[0x28]

```

```

0xd077: mov     qword ptr [rsp + 8], rax
0xd07c: xor     eax, eax
0xd07e: call    0x1d670
0xd083: mov     qword ptr [rax + 8], 1
0xd08b: mov     qword ptr [rax + 0x58], 0
0xd093: mov     qword ptr [rax + 0x60], 0
0xd09b: mov     qword ptr [rax + 0x28], 0
0xd0a3: mov     qword ptr [rax + 0x30], 0
0xd0ab: mov     qword ptr [rip + 0x1e356], rax
0xd0b2: xor     eax, eax
0xd0b4: call    0xcff0
0xd0b9: mov     rbx, qword ptr [rip + 0x1e348]
0xd0c0: xor     eax, eax
0xd0c2: call    0xc590
0xd0c7: lea     rdi, [rip + 0x135e4]
0xd0ce: mov     qword ptr [rbx], rax
0xd0d1: call    0x17c50
0xd0d6: mov     ecx, 1
0xd0db: mov     rsi, rsp
0xd0de: mov     edx, 1
0xd0e3: lea     rdi, [rip + 0x13e42]
0xd0ea: call    0x17320
0xd0ef: mov     rdi, qword ptr [rsp]
0xd0f3: lea     rax, [rip + 0x13e24]
0xd0fa: mov     rbx, qword ptr [rip + 0x1e307]
0xd101: test    rdi, rdi
0xd104: cmove   rdi, rax
0xd108: xor     eax, eax
0xd10a: call    0x19940
0xd10f: lea     rdi, [rip + 0x1cb32]
0xd116: mov     qword ptr [rbx + 0x68], rax
0xd11a: mov     eax, dword ptr [rip + 0x1cb40]
0xd120: test    eax, eax
0xd122: lea     rax, [rip + 0x1cb07]
0xd129: cmove   rdi, rax
0xd12d: call    0x17c50
0xd132: lea     rdi, [rip + 0x1d0c7]
0xd139: mov     edx, 6
0xd13e: lea     rsi, [rip + 0x1d0db]
0xd145: call    0x17f70
0xd14a: xor     eax, eax
0xd14c: call    0x1a310
0xd151: mov     rax, qword ptr [rip + 0x1e2b0]
0xd158: mov     rdi, qword ptr [rax + 0x68]
0xd15c: xor     eax, eax
0xd15e: call    0x1a900
0xd163: nop     dword ptr [rax + rax]
0xd168: xor     eax, eax

```

```

0xd16a: call    0x1ab40
0xd16f: test    rax, rax
0xd172: jne     0xd168
0xd174: mov     edi, 0x200
0xd179: call    0x1a620
0xd17e: mov     rax, qword ptr [rsp + 8]
0xd183: sub     rax, qword ptr fs:[0x28]
0xd18c: jne     0xd194
0xd18e: add     rsp, 0x10
0xd192: pop     rbx
0xd193: ret
0xd194: call    0x49d0
0xd199: nop     dword ptr [rax]
0xd1a0: endbr64
0xd1a4: sub     rsp, 8
0xd1a8: mov     rax, qword ptr [rip + 0x1e259]
0xd1af: or      edi, 2
0xd1b2: mov     rdx, qword ptr [rax + 0x40]
0xd1b6: mov     dword ptr [rax + 0x10], 6
0xd1bd: mov     qword ptr [rax + 0x20], rdx
0xd1c1: mov     rdx, qword ptr [rax + 0x38]
0xd1c5: mov     qword ptr [rax + 0x18], rdx
0xd1c9: xor     eax, eax
0xd1cb: call    0xc6e0
0xd1d0: mov     rax, qword ptr [rip + 0x1e231]
0xd1d7: mov     rdx, qword ptr [rax + 0x18]
0xd1db: mov     dword ptr [rax + 0x10], 0xd
0xd1e2: mov     qword ptr [rax + 0x20], rdx
0xd1e6: add     rsp, 8
0xd1ea: ret
0xd1eb: nop     dword ptr [rax + rax]
0xd1f0: endbr64
0xd1f4: mov     rax, qword ptr [rip + 0x1e20d]
0xd1fb: push    rbx
0xd1fc: mov     edi, 3
0xd201: mov     rdx, qword ptr [rax + 0x18]
0xd205: mov     rsi, qword ptr [rax + 0x20]
0xd209: mov     dword ptr [rax + 0x10], 5
0xd210: mov     rcx, qword ptr [rax + 0x40]
0xd214: mov     qword ptr [rax + 0x20], rdx
0xd218: mov     rbx, rsi
0xd21b: mov     qword ptr [rax + 0x18], rcx
0xd21f: sub     rbx, rdx
0xd222: cmp     rsi, rdx
0xd225: cmovbe  rbx, rcx
0xd229: xor     eax, eax
0xd22b: call    0xc6e0
0xd230: mov     rdx, qword ptr [rip + 0x1e1d1]

```

```

0xd237: mov     rax, qword ptr [rdx + 0x38]
0xd23b: mov     dword ptr [rdx + 0x10], 0xe
0xd242: cmp     rax, rbx
0xd245: cmova   rax, rbx
0xd249: mov     qword ptr [rdx + 0x20], rax
0xd24d: pop     rbx
0xd24e: ret
0xd24f: nop
0xd250: endbr64
0xd254: mov     rax, qword ptr [rip + 0x1e1ad]
0xd25b: push    rbx
0xd25c: mov     edi, 3
0xd261: mov     rdx, qword ptr [rax + 0x18]
0xd265: mov     rbx, qword ptr [rax + 0x20]
0xd269: mov     dword ptr [rax + 0x10], 6
0xd270: mov     qword ptr [rax + 0x20], rdx
0xd274: mov     rdx, qword ptr [rax + 0x38]
0xd278: mov     qword ptr [rax + 0x18], rdx
0xd27c: xor     eax, eax
0xd27e: call    0xc6e0
0xd283: mov     rdx, qword ptr [rip + 0x1e17e]
0xd28a: mov     rax, qword ptr [rdx + 0x38]
0xd28e: mov     dword ptr [rdx + 0x10], 0xf
0xd295: cmp     rax, rbx
0xd298: cmova   rax, rbx
0xd29c: mov     qword ptr [rdx + 0x20], rax
0xd2a0: pop     rbx
0xd2a1: ret
0xd2a2: nop     word ptr cs:[rax + rax]
0xd2ad: nop     dword ptr [rax]
0xd2b0: endbr64
0xd2b4: mov     rax, qword ptr [rip + 0x1e14d]
0xd2bb: mov     rdx, qword ptr [rax + 0x58]
0xd2bf: mov     rax, qword ptr [rax + 0x18]
0xd2c3: cmp     byte ptr [rdx + rax], 0x20
0xd2c7: je      0xd2d0
0xd2c9: ret
0xd2ca: nop     word ptr [rax + rax]
0xd2d0: sub     rsp, 8
0xd2d4: nop     dword ptr [rax]
0xd2d8: xor     eax, eax
0xd2da: call    0xcde0
0xd2df: mov     rax, qword ptr [rip + 0x1e122]
0xd2e6: mov     rdx, qword ptr [rax + 0x58]
0xd2ea: mov     rcx, qword ptr [rax + 0x18]
0xd2ee: cmp     byte ptr [rdx + rcx], 0x20
0xd2f2: je      0xd2d8
0xd2f4: cmp     qword ptr [rax + 0x48], 0

```

```

0xd2f9: jne      0xd31d
0xd2fb: jmp      0xd324
0xd324:
0xd2fd: nop      dword ptr [rax]
0xd300: xor      eax, eax
0xd302: call     0xce70
0xd307: mov      rax, qword ptr [rip + 0x1e0fa]
0xd30e: cmp      qword ptr [rax + 0x48], 0
0xd313: je       0xd324
0xd315: mov      rdx, qword ptr [rax + 0x58]
0xd319: mov      rcx, qword ptr [rax + 0x18]
0xd31d: cmp      byte ptr [rdx + rcx - 1], 0x20
0xd322: je       0xd300
0xd324: mov      dword ptr [rax + 0x10], 0x11
0xd32b: add      rsp, 8
0xd32f: ret
0xd330: endbr64
0xd334: mov      rax, qword ptr [rip + 0x1e0cd]
0xd33b: mov      rdx, qword ptr [rax + 0x58]
0xd33f: mov      rax, qword ptr [rax + 0x18]
0xd343: cmp      byte ptr [rdx + rax], 0x20
0xd347: je       0xd350
0xd349: ret
0xd34a: nop      word ptr [rax + rax]
0xd350: sub      rsp, 8
0xd354: xor      eax, eax
0xd356: call     0xd2b0
0xd35b: mov      edi, 0x20
0xd360: call     0xcd30
0xd365: mov      rax, qword ptr [rip + 0x1e09c]
0xd36c: mov      dword ptr [rax + 0x10], 0x10
0xd373: add      rsp, 8
0xd377: ret
0xd378: nop      dword ptr [rax + rax]
0xd380: endbr64
0xd384: push     rbp
0xd385: push     rbx
0xd386: sub      rsp, 8
0xd38a: mov      rbp, qword ptr [rip + 0x1e077]
0xd391: mov      rbx, qword ptr [rbp + 0x18]
0xd395: cmp      rbx, qword ptr [rbp + 0x38]
0xd399: jb       0xd3a8
0xd39b: add      rsp, 8
0xd39f: pop      rbx
0xd3a0: pop      rbp
0xd3a1: ret
0xd3a2: nop      word ptr [rax + rax]
0xd3a8: xor      eax, eax

```

```

0xd3aa: call    0xcc30
0xd3af: cmp     rbx, qword ptr [rbp + 0x18]
0xd3b3: jae     0xd3e4
0xd3b5: call    0x4ef0
0xd3ba: nop     word ptr [rax + rax]
0xd3c0: mov     rdx, qword ptr [rbp + 0x58]
0xd3c4: mov     rcx, qword ptr [rax]
0xd3c7: add     rdx, rbx
0xd3ca: add     rbx, 1
0xd3ce: movsx   rsi, byte ptr [rdx]
0xd3d2: mov     ecx, dword ptr [rcx + rsi*4]
0xd3d5: mov     byte ptr [rdx], cl
0xd3d7: mov     rbp, qword ptr [rip + 0x1e02a]
0xd3de: cmp     qword ptr [rbp + 0x18], rbx
0xd3e2: ja      0xd3c0
0xd3e4: mov     dword ptr [rbp + 0x10], 0x12
0xd3eb: add     rsp, 8
0xd3ef: pop     rbx
0xd3f0: pop     rbp
0xd3f1: ret
0xd3f2: nop     word ptr cs:[rax + rax]
0xd3fd: nop     dword ptr [rax]
0xd400: endbr64
0xd404: push    rbp
0xd405: push    rbx
0xd406: sub     rsp, 8
0xd40a: mov     rbp, qword ptr [rip + 0x1dff7]
0xd411: mov     rbx, qword ptr [rbp + 0x18]
0xd415: cmp     rbx, qword ptr [rbp + 0x38]
0xd419: jnb     0xd428
0xd41b: add     rsp, 8
0xd41f: pop     rbx
0xd420: pop     rbp
0xd421: ret
0xd422: nop     word ptr [rax + rax]
0xd428: xor     eax, eax
0xd42a: call    0xcc30
0xd42f: cmp     rbx, qword ptr [rbp + 0x18]
0xd433: jae     0xd464
0xd435: call    0x47c0
0xd43a: nop     word ptr [rax + rax]
0xd440: mov     rdx, qword ptr [rbp + 0x58]
0xd444: mov     rcx, qword ptr [rax]
0xd447: add     rdx, rbx
0xd44a: add     rbx, 1
0xd44e: movsx   rsi, byte ptr [rdx]
0xd452: mov     ecx, dword ptr [rcx + rsi*4]
0xd455: mov     byte ptr [rdx], cl

```

```

0xd457: mov     rbp, qword ptr [rip + 0x1dfaa]
0xd45e: cmp     qword ptr [rbp + 0x18], rbx
0xd462: ja      0xd440
0xd464: mov     dword ptr [rbp + 0x10], 0x13
0xd46b: add     rsp, 8
0xd46f: pop     rbx
0xd470: pop     rbp
0xd471: ret
0xd472: nop     word ptr cs:[rax + rax]
0xd47d: nop     dword ptr [rax]
0xd480: endbr64
0xd484: push    r14
0xd486: push    r13
0xd488: push    r12
0xd48a: push    rbp
0xd48b: mov     rbp, qword ptr [rip + 0x1df76]
0xd492: push    rbx
0xd493: mov     rbx, qword ptr [rbp + 0x18]
0xd497: cmp     rbx, qword ptr [rbp + 0x38]
0xd49b: jb      0xd4b0
0xd49d: pop     rbx
0xd49e: pop     rbp
0xd49f: pop     r12
0xd4a1: pop     r13
0xd4a3: pop     r14
0xd4a5: ret
0xd4a6: nop     word ptr cs:[rax + rax]
0xd4b0: xor     eax, eax
0xd4b2: call    0xcc30
0xd4b7: cmp     rbx, qword ptr [rbp + 0x18]
0xd4bb: jae     0xd528
0xd4bd: call    0x4f00
0xd4c2: mov     r14d, 1
0xd4c8: mov     r13, rax
0xd4cb: jmp     0xd4f3
0xd4f3:
0xd4cd: nop     dword ptr [rax]
0xd4d0: call    0x47c0
0xd4d5: xor     r14d, r14d
0xd4d8: mov     rax, qword ptr [rax]
0xd4db: mov     eax, dword ptr [rax + r12*4]
0xd4df: mov     byte ptr [rbp], al
0xd4e2: mov     rbp, qword ptr [rip + 0x1df1f]
0xd4e9: add     rbx, 1
0xd4ed: cmp     qword ptr [rbp + 0x18], rbx
0xd4f1: jbe     0xd528
0xd4f3: mov     rax, qword ptr [rbp + 0x58]
0xd4f7: add     rax, rbx

```

```

0xd4fa: movsx    r12, byte ptr [rax]
0xd4fe: mov      rbp, rax
0xd501: mov      rax, qword ptr [r13]
0xd505: test     byte ptr [rax + r12*2], 8
0xd50a: je       0xd4e2
0xd50c: test     r14d, r14d
0xd50f: jne      0xd4d0
0xd511: call     0x4ef0
0xd516: mov      rax, qword ptr [rax]
0xd519: mov      eax, dword ptr [rax + r12*4]
0xd51d: mov      byte ptr [rbp], al
0xd520: jmp      0xd4e2
0xd4e2:
0xd522: nop      word ptr [rax + rax]
0xd528: mov      dword ptr [rbp + 0x10], 0x14
0xd52f: pop      rbx
0xd530: pop      rbp
0xd531: pop      r12
0xd533: pop      r13
0xd535: pop      r14
0xd537: ret
0xd538: nop      dword ptr [rax + rax]
0xd540: endbr64
0xd544: push     r12
0xd546: mov      r12, rdi
0xd549: push     rbp
0xd54a: push     rbx
0xd54b: call     0x49b0
0xd550: mov      rbp, qword ptr [rip + 0x1deb1]
0xd557: mov      rbx, rax
0xd55a: mov      rsi, qword ptr [rbp + 0x40]
0xd55e: sub      rax, rsi
0xd561: add      qword ptr [rbp + 0x18], rax
0xd565: add      qword ptr [rbp + 0x20], rax
0xd569: mov      rax, qword ptr [rbp + 0x48]
0xd56d: lea      rdx, [rax + 1]
0xd571: lea      r8, [rbx + rdx]
0xd575: cmp      r8, qword ptr [rbp + 0x50]
0xd579: ja       0xd5d8
0xd57b: mov      rdi, qword ptr [rbp + 0x58]
0xd57f: add      rsi, rdi
0xd582: add      rdi, rbx
0xd585: call     0x4d40
0xd58a: mov      rax, qword ptr [rip + 0x1de77]
0xd591: mov      rdx, rbx
0xd594: mov      rsi, r12
0xd597: mov      rdi, qword ptr [rax + 0x58]
0xd59b: call     0x4bb0

```



```

0xd5a0: mov     rax, qword ptr [rip + 0x1de61]
0xd5a7: mov     rsi, rbx
0xd5aa: mov     rdi, qword ptr [rax + 0x58]
0xd5ae: call    0x184d0
0xd5b3: mov     rax, qword ptr [rip + 0x1de4e]
0xd5ba: mov     qword ptr [rax + 0x40], rbx
0xd5be: add     rbx, qword ptr [rax + 0x48]
0xd5c2: mov     qword ptr [rax + 0x38], rbx
0xd5c6: mov     dword ptr [rax + 0x10], 0x15
0xd5cd: pop     rbx
0xd5ce: pop     rbp
0xd5cf: pop     r12
0xd5d1: ret
0xd5d2: nop     word ptr [rax + rax]
0xd5d8: mov     qword ptr [rbp + 0x50], r8
0xd5dc: mov     rdi, qword ptr [rbp + 0x58]
0xd5e0: mov     rsi, r8
0xd5e3: call    0x1d690
0xd5e8: mov     qword ptr [rbp + 0x58], rax
0xd5ec: mov     rbp, qword ptr [rip + 0x1de15]
0xd5f3: mov     rax, qword ptr [rbp + 0x48]
0xd5f7: mov     rsi, qword ptr [rbp + 0x40]
0xd5fb: lea     rdx, [rax + 1]
0xd5ff: jmp     0xd57b
0xd57b:
0xd604: nop     word ptr cs:[rax + rax]
0xd60f: nop
0xd610: endbr64
0xd614: test    rdi, rdi
0xd617: je      0xd6f0
0xd61d: push    r12
0xd61f: push    rbp
0xd620: mov     rbp, rdi
0xd623: push    rbx
0xd624: call    0x49b0
0xd629: mov     r12, qword ptr [rip + 0x1ddd8]
0xd630: mov     rbx, rax
0xd633: mov     rax, qword ptr [r12 + 0x38]
0xd638: lea     rdx, [rax + 1]
0xd63c: mov     rax, qword ptr [r12 + 0x50]
0xd641: lea     rcx, [rbx + rdx]
0xd645: cmp     rcx, rax
0xd648: ja      0xd6c0
0xd64a: mov     rax, qword ptr [r12 + 0x18]
0xd64f: mov     rdi, qword ptr [r12 + 0x58]
0xd654: sub     rdx, rax
0xd657: lea     rsi, [rdi + rax]
0xd65b: add     rax, rbx

```

```

0xd65e: add    rdi, rax
0xd661: call   0x4d40
0xd666: mov    rax, qword ptr [rip + 0x1dd9b]
0xd66d: mov    rdx, rbx
0xd670: mov    rsi, rbp
0xd673: mov    rdi, qword ptr [rax + 0x18]
0xd677: add    rdi, qword ptr [rax + 0x58]
0xd67b: call   0x4bb0
0xd680: mov    rax, qword ptr [rip + 0x1dd81]
0xd687: mov    rsi, rbx
0xd68a: mov    rdi, qword ptr [rax + 0x18]
0xd68e: add    rdi, qword ptr [rax + 0x58]
0xd692: call   0x184d0
0xd697: mov    rax, qword ptr [rip + 0x1dd6a]
0xd69e: add    qword ptr [rax + 0x18], rbx
0xd6a2: add    qword ptr [rax + 0x38], rbx
0xd6a6: add    qword ptr [rax + 0x48], rbx
0xd6aa: mov    dword ptr [rax + 0x10], 0x16
0xd6b1: pop    rbx
0xd6b2: pop    rbp
0xd6b3: pop    r12
0xd6b5: ret
0xd6b6: nop    word ptr cs:[rax + rax]
0xd6c0: lea    rsi, [rax + rbx + 0x21]
0xd6c5: mov    rdi, qword ptr [r12 + 0x58]
0xd6ca: mov    qword ptr [r12 + 0x50], rsi
0xd6cf: call   0x1d690
0xd6d4: mov    qword ptr [r12 + 0x58], rax
0xd6d9: mov    r12, qword ptr [rip + 0x1dd28]
0xd6e0: mov    rax, qword ptr [r12 + 0x38]
0xd6e5: lea    rdx, [rax + 1]
0xd6e9: jmp    0xd64a
0xd64a:
0xd6ee: nop
0xd6f0: ret
0xd6f1: nop    word ptr cs:[rax + rax]
0xd6fc: nop    dword ptr [rax]
0xd700: endbr64
0xd704: mov    rax, qword ptr [rip + 0x1dcfd]
0xd70b: mov    rdi, qword ptr [rax + 0x60]
0xd70f: test   rdi, rdi
0xd712: je     0xd730
0xd714: sub    rsp, 8
0xd718: call   0xd610
0xd71d: mov    rax, qword ptr [rip + 0x1dce4]
0xd724: mov    dword ptr [rax + 0x10], 0x1a
0xd72b: add    rsp, 8
0xd72f: ret

```

```

0xd730: ret
0xd731: nop      word ptr cs:[rax + rax]
0xd73c: nop      dword ptr [rax]
0xd740: endbr64
0xd744: mov      rax, qword ptr [rip + 0x1dcbd]
0xd74b: push     r12
0xd74d: push     rbp
0xd74e: push     rbx
0xd74f: mov      rbx, qword ptr [rax + 0x40]
0xd753: mov      r12, qword ptr [rax + 0x28]
0xd757: mov      eax, dword ptr [rax + 0xc]
0xd75a: lea      rbp, [rbx + 3]
0xd75e: test     eax, eax
0xd760: je       0xd7f0
0xd766: mov      edx, dword ptr [rip + 0x1caa4]
0xd76c: mov      esi, dword ptr [rip + 0x1ca9a]
0xd772: mov      edi, dword ptr [rip + 0x1ca9c]
0xd778: call     0xa570
0xd77d: mov      rcx, qword ptr [rip + 0x1dc84]
0xd784: cmp      r12, rbp
0xd787: jb       0xd810
0xd78d: mov      rsi, qword ptr [rcx + 0x18]
0xd791: mov      rdx, qword ptr [rcx + 0x40]
0xd795: xor      ebx, ebx
0xd797: mov      rdi, qword ptr [rcx + 0x28]
0xd79b: mov      rax, rdi
0xd79e: sub      rax, rdx
0xd7a1: mov      edx, 1
0xd7a6: mov      r8, rax
0xd7a9: shr      r8, 2
0xd7ad: cmp      rax, 3
0xd7b1: cmova    edx, r8d
0xd7b5: xor      eax, eax
0xd7b7: cmp      rdi, rsi
0xd7ba: ja       0xd7d3
0xd7bc: mov      rax, rsi
0xd7bf: sub      rax, rdi
0xd7c2: movsxd   rdi, edx
0xd7c5: lea      r8d, [rdx + rax]
0xd7c9: xor      edx, edx
0xd7cb: div      rdi
0xd7ce: mov      eax, r8d
0xd7d1: sub      eax, edx
0xd7d3: mov      edx, esi
0xd7d5: mov      rdi, qword ptr [rcx]
0xd7d8: xor      esi, esi
0xd7da: sub      edx, eax
0xd7dc: call     0xc680

```

```

0xd7e1: cmp     r12, rbp
0xd7e4: jb      0xd840
0xd7e6: pop     rbx
0xd7e7: pop     rbp
0xd7e8: pop     r12
0xd7ea: ret
0xd7eb: nop     dword ptr [rax + rax]
0xd7f0: mov     edx, dword ptr [rip + 0x1ca0e]
0xd7f6: mov     esi, dword ptr [rip + 0x1ca04]
0xd7fc: mov     edi, dword ptr [rip + 0x1ca06]
0xd802: jmp     0xd778
0xd778:
0xd807: nop     word ptr [rax + rax]
0xd810: mov     rbx, qword ptr [rcx + 0x40]
0xd814: mov     rsi, qword ptr [rcx + 0x18]
0xd818: mov     qword ptr [rcx + 0x40], 0
0xd820: xor     edx, edx
0xd822: add     qword ptr [rcx + 0x58], rbx
0xd826: sub     rsi, rbx
0xd829: sub     qword ptr [rcx + 0x38], rbx
0xd82d: sub     qword ptr [rcx + 0x20], rbx
0xd831: movsxd  rbx, ebx
0xd834: mov     qword ptr [rcx + 0x18], rsi
0xd838: jmp     0xd797
0xd797:
0xd83d: nop     dword ptr [rax]
0xd840: mov     rax, qword ptr [rip + 0x1dbc1]
0xd847: movsxd  rbx, ebx
0xd84a: mov     qword ptr [rax + 0x40], rbx
0xd84e: sub     qword ptr [rax + 0x58], rbx
0xd852: add     qword ptr [rax + 0x38], rbx
0xd856: add     qword ptr [rax + 0x18], rbx
0xd85a: add     qword ptr [rax + 0x20], rbx
0xd85e: pop     rbx
0xd85f: pop     rbp
0xd860: pop     r12
0xd862: ret
0xd863: nop     word ptr cs:[rax + rax]
0xd86e: nop
0xd870: endbr64
0xd874: push    r15
0xd876: push    r14
0xd878: push    r13
0xd87a: push    r12
0xd87c: push    rbp
0xd87d: push    rbx
0xd87e: sub     rsp, 0x28
0xd882: mov     rax, qword ptr fs:[0x28]

```

```

0xd88b: mov     qword ptr [rsp + 0x18], rax
0xd890: mov     rax, qword ptr [rip + 0x1db71]
0xd897: lea     r14, [rsp + 0x17]
0xd89c: mov     rdi, r14
0xd89f: mov     rbx, qword ptr [rax + 0x40]
0xd8a3: mov     r13, qword ptr [rax + 0x28]
0xd8a7: call    0xa6a0
0xd8ac: mov     rax, qword ptr [rip + 0x1db55]
0xd8b3: add     rbx, 3
0xd8b7: mov     ecx, dword ptr [rax + 0xc]
0xd8ba: test    ecx, ecx
0xd8bc: je      0xda20
0xd8c2: mov     edx, dword ptr [rip + 0x1c948]
0xd8c8: mov     esi, dword ptr [rip + 0x1c93e]
0xd8ce: mov     edi, dword ptr [rip + 0x1c940]
0xd8d4: call    0xa570
0xd8d9: mov     rdx, qword ptr [rip + 0x1db28]
0xd8e0: cmp     r13, rbx
0xd8e3: jb      0xda90
0xd8e9: mov     qword ptr [rsp + 8], 0
0xd8f2: mov     rax, qword ptr [rdx + 0x18]
0xd8f6: mov     rcx, qword ptr [rdx + 0x40]
0xd8fa: mov     rdi, qword ptr [rdx + 0x28]
0xd8fe: mov     r15, rdi
0xd901: sub     r15, rcx
0xd904: cmp     r15, 3
0xd908: jbe     0xda80
0xd90e: shr     r15, 2
0xd912: movsxd  rcx, r15d
0xd915: cmp     rdi, rax
0xd918: ja      0xda70
0xd91e: mov     edx, eax
0xd920: sub     rax, rdi
0xd923: sub     edx, edi
0xd925: add     r15d, edx
0xd928: xor     edx, edx
0xd92a: div     rcx
0xd92d: sub     r15d, edx
0xd930: mov     r12d, r15d
0xd933: call    0x1d670
0xd938: mov     esi, 0x20
0xd93d: mov     rbp, rax
0xd940: mov     rax, qword ptr [rip + 0x1dac1]
0xd947: mov     rdi, rbp
0xd94a: mov     rdx, qword ptr [rax + 0x28]
0xd94e: call    0x4a50
0xd953: mov     rax, qword ptr [rip + 0x1daae]
0xd95a: mov     edx, dword ptr [rax + 8]

```

```

0xd95d: test    edx, edx
0xd95f: je      0xda40
0xd965: mov     rdx, qword ptr [rax + 0x38]
0xd969: sub     rdx, qword ptr [rax + 0x40]
0xd96d: mov     rdi, rbp
0xd970: sub     rdx, r12
0xd973: mov     rsi, qword ptr [rax + 0x28]
0xd977: sub     rsi, qword ptr [rax + 0x40]
0xd97b: cmp     rdx, rsi
0xd97e: cmova   rdx, rsi
0xd982: mov     rsi, qword ptr [rax + 0x40]
0xd986: add     rsi, r12
0xd989: add     rsi, qword ptr [rax + 0x58]
0xd98d: call    0x4bb0
0xd992: mov     rax, qword ptr [rip + 0x1da6f]
0xd999: xor     edx, edx
0xd99b: xor     esi, esi
0xd99d: mov     rdi, qword ptr [rax]
0xd9a0: call    0xc680
0xd9a5: cmp     r13, rbx
0xd9a8: jae     0xdaf0
0xd9ae: mov     rax, qword ptr [rip + 0x1da53]
0xd9b5: mov     rsi, rbp
0xd9b8: mov     rdi, qword ptr [rax]
0xd9bb: mov     rdx, qword ptr [rax + 0x28]
0xd9bf: sub     edx, dword ptr [rax + 0x40]
0xd9c2: call    0xc600
0xd9c7: mov     rax, qword ptr [rip + 0x1da3a]
0xd9ce: xor     esi, esi
0xd9d0: mov     edx, dword ptr [rax + 0x18]
0xd9d3: mov     rdi, qword ptr [rax]
0xd9d6: sub     edx, r15d
0xd9d9: call    0xc680
0xd9de: cmp     r13, rbx
0xd9e1: jb      0xdac8
0xd9e7: mov     rdi, rbp
0xd9ea: call    0x18d10
0xd9ef: mov     rdi, r14
0xd9f2: call    0xa6b0
0xd9f7: mov     rax, qword ptr [rsp + 0x18]
0xd9fc: sub     rax, qword ptr fs:[0x28]
0xda05: jne     0xdb0b
0xda0b: add     rsp, 0x28
0xda0f: pop     rbx
0xda10: pop     rbp
0xda11: pop     r12
0xda13: pop     r13
0xda15: pop     r14

```

```

0xda17: pop      r15
0xda19: ret
0xda1a: nop      word ptr [rax + rax]
0xda20: mov      edx, dword ptr [rip + 0x1c7de]
0xda26: mov      esi, dword ptr [rip + 0x1c7d4]
0xda2c: mov      edi, dword ptr [rip + 0x1c7d6]
0xda32: jmp      0xd8d4
0xd8d4
0xda37: nop      word ptr [rax + rax]
0xda40: mov      rsi, qword ptr [rax + 0x40]
0xda44: mov      rdx, qword ptr [rax + 0x38]
0xda48: mov      rdi, rbp
0xda4b: mov      rax, qword ptr [rax + 0x28]
0xda4f: sub      rdx, rsi
0xda52: sub      rdx, r12
0xda55: sub      rax, rsi
0xda58: mov      esi, 0x2a
0xda5d: cmp      rdx, rax
0xda60: cmova    rdx, rax
0xda64: call     0x4a50
0xda69: jmp      0xd992
0xd992
0xda6e: nop
0xda70: xor      r12d, r12d
0xda73: xor      r15d, r15d
0xda76: jmp      0xd933
0xd933
0xda7b: nop      dword ptr [rax + rax]
0xda80: mov      ecx, 1
0xda85: mov      r15d, 1
0xda8b: jmp      0xd915
0xd915
0xda90: mov      r12, qword ptr [rdx + 0x40]
0xda94: mov      rax, qword ptr [rdx + 0x18]
0xda98: mov      qword ptr [rdx + 0x40], 0
0xdaa0: add      qword ptr [rdx + 0x58], r12
0xdaa4: movsxd   rcx, r12d
0xdaa7: sub      rax, r12
0xdaaa: sub      qword ptr [rdx + 0x38], r12
0xdaae: mov      qword ptr [rsp + 8], rcx
0xdab3: xor      ecx, ecx
0xdab5: sub      qword ptr [rdx + 0x20], r12
0xdab9: mov      qword ptr [rdx + 0x18], rax
0xdabd: jmp      0xd8fa
0xd8fa
0xdac2: nop      word ptr [rax + rax]
0xdac8: mov      rax, qword ptr [rip + 0x1d939]
0xdacf: movsxd   r12, dword ptr [rsp + 8]

```

```

0xdad4: sub     qword ptr [rax + 0x58], r12
0xdad8: add     qword ptr [rax + 0x38], r12
0xdadc: add     qword ptr [rax + 0x18], r12
0xdae0: add     qword ptr [rax + 0x20], r12
0xdae4: mov     qword ptr [rax + 0x40], r12
0xdae8: jmp     0xd9e7
0xd9e7:
0xdaed: nop     dword ptr [rax]
0xdaf0: mov     rax, qword ptr [rip + 0xd911]
0xdaf7: mov     rsi, qword ptr [rax + 0x58]
0xdafb: mov     edx, dword ptr [rax + 0x40]
0xdafe: mov     rdi, qword ptr [rax]
0xdb01: call    0xc600
0xdb06: jmp     0xd9ae
0xd9ae:
0xdb0b: call    0x49d0
0xdb10: endbr64
0xdb14: mov     rax, qword ptr [rip + 0xd8ed]
0xdb1b: push    rbx
0xdb1c: mov     rbx, rdi
0xdb1f: mov     rdi, qword ptr [rdi]
0xdb22: mov     rsi, qword ptr [rax + 0x48]
0xdb26: add     rsi, 1
0xdb2a: call    0xd690
0xdb2f: mov     qword ptr [rbx], rax
0xdb32: mov     rdi, rax
0xdb35: mov     rax, qword ptr [rip + 0xd8cc]
0xdb3c: mov     rcx, qword ptr [rax + 0x48]
0xdb40: mov     rsi, qword ptr [rax + 0x40]
0xdb44: add     rsi, qword ptr [rax + 0x58]
0xdb48: lea     rdx, [rcx + 1]
0xdb4c: call    0x4bb0
0xdb51: mov     rax, qword ptr [rip + 0xd8b0]
0xdb58: pop     rbx
0xdb59: mov     eax, dword ptr [rax + 0x48]
0xdb5c: ret
0xdb5d: nop     dword ptr [rax]
0xdb60: endbr64
0xdb64: push    rbp
0xdb65: xor     eax, eax
0xdb67: mov     rbp, rdi
0xdb6a: call    0xcff0
0xdb6f: test    rbp, rbp
0xdb72: lea     rax, [rip + 0x133bf]
0xdb79: cmov     rbp, rax
0xdb7d: mov     rdi, rbp
0xdb80: call    0xd540
0xdb85: xor     eax, eax

```



```

0xdb87: call    0xd870
0xdb8c: xor      eax, eax
0xdb8e: pop      rbp
0xdb8f: jmp      0xd740
0xd740:
0xdb94: nop      word ptr cs:[rax + rax]
0xdb9f: nop
0xdba0: endbr64
0xdba4: mov      rax, qword ptr [rip + 0xd85d]
0xdbab: mov      dword ptr [rax + 0xc], edi
0xdbae: ret
0xdbaf: nop
0xdbb0: endbr64
0xdbb4: push     r12
0xdbb6: push     rbp
0xdbb7: push     rbx
0xdbb8: cmp      edi, 1
0xdbbb: je       0xdbd0
0xdbbd: cmp      edi, 2
0xdbc0: je       0xdc90
0xdbc6: test     edi, edi
0xdbc8: je       0xdc40
0xdbca: pop      rbx
0xdbcb: pop      rbp
0xdbcc: pop      r12
0xdbce: ret
0xdbcf: nop
0xdbd0: mov      ecx, dword ptr [rip + 0xd83e]
0xdbd6: test     ecx, ecx
0xdbd8: je       0xdd38
0xdbde: xor      eax, eax
0xdbe0: call     0x1ab10
0xdbe5: mov      rbp, rax
0xdbe8: test     rax, rax
0xdbeb: je       0xdbca
0xdbed: mov      r12, qword ptr [rbp]
0xdbf1: mov      rbx, qword ptr [rip + 0xd810]
0xdbf8: mov      rdi, r12
0xdbfb: call     0x49b0
0xdc00: mov      rdi, qword ptr [rbx + 0x40]
0xdc04: mov      qword ptr [rbx + 0x48], rax
0xdc08: lea      rsi, [rax + rdi]
0xdc0c: mov      qword ptr [rbx + 0x38], rsi
0xdc10: mov      qword ptr [rbx + 0x18], rsi
0xdc14: add      rsi, 1
0xdc18: cmp      rsi, qword ptr [rbx + 0x50]
0xdc1c: ja       0xdd70
0xdc22: add      rdi, qword ptr [rbx + 0x58]

```

```

0xdc26: mov     rsi, r12
0xdc29: call    0x48b0
0xdc2e: pop     rbx
0xdc2f: xor     eax, eax
0xdc31: pop     rbp
0xdc32: pop     r12
0xdc34: jmp     0xd870
0xd870:
0xdc39: nop     dword ptr [rax]
0xdc40: mov     eax, dword ptr [rip + 0x1d7ce]
0xdc46: test    eax, eax
0xdc48: jne     0xdd18
0xdc4e: xor     eax, eax
0xdc50: call    0x1ab10
0xdc55: test    rax, rax
0xdc58: je      0xdd98
0xdc5e: mov     rdx, qword ptr [rip + 0x1d7a3]
0xdc65: mov     rsi, qword ptr [rax]
0xdc68: mov     rbp, qword ptr [rdx + 0x40]
0xdc6c: add     rbp, qword ptr [rdx + 0x58]
0xdc70: mov     rdi, rbp
0xdc73: call    0x4b30
0xdc78: test    eax, eax
0xdc7a: jne     0xdd58
0xdc80: pop     rbx
0xdc81: xor     eax, eax
0xdc83: pop     rbp
0xdc84: pop     r12
0xdc86: jmp     0x1ab40
0x1ab40:
0xdc8b: nop     dword ptr [rax + rax]
0xdc90: mov     edx, dword ptr [rip + 0x1d77e]
0xdc96: test    edx, edx
0xdc98: je      0xdcf8
0xdc9a: xor     eax, eax
0xdc9c: call    0x1ab40
0xdca1: mov     rbp, rax
0xdca4: test    rax, rax
0xdca7: jne     0xdbed
0xdcad: mov     rax, qword ptr [rip + 0x1d754]
0xdcbb: mov     edi, 2
0xdcbb: mov     rdx, qword ptr [rax + 0x40]
0xdcbb: mov     dword ptr [rax + 0x10], 6
0xdcc4: mov     qword ptr [rax + 0x20], rdx
0xdcc8: mov     rdx, qword ptr [rax + 0x38]
0xdccc: mov     qword ptr [rax + 0x18], rdx
0xdcd0: xor     eax, eax
0xdcd2: call    0xc6e0

```

```

0xcd7: mov     rax, qword ptr [rip + 0x1d72a]
0xcd8: mov     rdx, qword ptr [rax + 0x18]
0xce2: mov     dword ptr [rax + 0x10], 0xd
0xce9: mov     qword ptr [rax + 0x20], rdx
0xcded: jmp     0xdc2e
0xdc2e:
0xdcf2: nop     word ptr [rax + rax]
0xdcf8: mov     dword ptr [rip + 0x1d712], 1
0xdd02: xor     eax, eax
0xdd04: call    0x1a570
0xdd09: mov     dword ptr [rip + 0x1d701], eax
0xdd0f: jmp     0xdc9a
0xdc9a:
0xdd11: nop     dword ptr [rax]
0xdd18: mov     edi, dword ptr [rip + 0x1d6f2]
0xdd1e: xor     eax, eax
0xdd20: call    0x1ac30
0xdd25: mov     dword ptr [rip + 0x1d6e5], 0
0xdd2f: jmp     0xdc4e
0xdc4e:
0xdd34: nop     dword ptr [rax]
0xdd38: mov     dword ptr [rip + 0x1d6d2], 1
0xdd42: xor     eax, eax
0xdd44: call    0x1a570
0xdd49: mov     dword ptr [rip + 0x1d6c1], eax
0xdd4f: jmp     0xdbde
0xdbde:
0xdd54: nop     dword ptr [rax]
0xdd58: xor     eax, eax
0xdd5a: mov     rdi, rbp
0xdd5d: call    0x1a380
0xdd62: xor     eax, eax
0xdd64: call    0x1ab40
0xdd69: jmp     0xdc80
0xdc80:
0xdd6e: nop
0xdd70: mov     qword ptr [rbx + 0x50], rsi
0xdd74: mov     rdi, qword ptr [rbx + 0x58]
0xdd78: call    0x1d690
0xdd7d: mov     qword ptr [rbx + 0x58], rax
0xdd81: mov     rbx, qword ptr [rip + 0x1d680]
0xdd88: mov     r12, qword ptr [rbp]
0xdd8c: mov     rdi, qword ptr [rbx + 0x40]
0xdd90: jmp     0xdc22
0xdc22:
0xdd95: nop     dword ptr [rax]
0xdd98: mov     rax, qword ptr [rip + 0x1d669]
0xdd9f: mov     rdi, qword ptr [rax + 0x40]

```

```

0xdda3: add     rdi, qword ptr [rax + 0x58]
0xdda7: xor     eax, eax
0xdda9: call    0x1a380
0xddae: jmp     0xdc80
0xdc80:
0xddb3: nop     word ptr cs:[rax + rax]
0xddbd: nop     dword ptr [rax]
0xddc0: endbr64
0xddc4: mov     eax, dword ptr [rip + 0x1be96]
0xddca: push    rbp
0xddcb: mov     rbp, rdi
0xddce: lea     rdi, [rip + 0x1be73]
0xddd5: test    eax, eax
0xddd7: lea     rax, [rip + 0x1be52]
0xddde: cmov    rdi, rax
0xdde2: call    0x17c50
0xdde7: mov     edx, 9
0xddec: lea     rsi, [rip + 0x1c4ad]
0xddf3: lea     rdi, [rip + 0x1c466]
0xddfa: call    0x17f70
0xddff: mov     rdi, rbp
0xde02: call    0x1d7a0
0xde07: mov     rdi, rax
0xde0a: mov     rbp, rax
0xde0d: mov     qword ptr [rip + 0x1d794], rax
0xde14: call    0x49b0
0xde19: mov     rdi, rbp
0xde1c: mov     rsi, rax
0xde1f: call    0x184d0
0xde24: xor     eax, eax
0xde26: call    0xc590
0xde2b: lea     rdi, [rip + 0x1d5ee]
0xde32: pop     rbp
0xde33: mov     qword ptr [rip + 0x1d78e], rax
0xde3a: jmp     0x4a30
0x4a30:
0xde3f: nop
0xde40: endbr64
0xde44: mov     rdi, qword ptr [rip + 0x1d77d]
0xde4b: jmp     0xc5c0
0xc5c0:
0xde50: endbr64
0xde54: push    rbp
0xde55: mov     rbp, rdi
0xde58: push    rbx
0xde59: mov     rbx, rsi
0xde5c: sub     rsp, 8
0xde60: mov     rdi, qword ptr [rip + 0x1d749]

```

```

0xde67: test    rdi, rdi
0xde6a: je        0xde71
0xde6c: call     0x18d10
0xde71: mov     rdi, rbp
0xde74: call     0x1d670
0xde79: mov     rdi, qword ptr [rip + 0x1d748]
0xde80: mov     edx, ebx
0xde82: xor     esi, esi
0xde84: mov     qword ptr [rip + 0x1d725], rax
0xde8b: add     rsp, 8
0xde8f: mov     r8d, ebp
0xde92: mov     ecx, 1
0xde97: pop     rbx
0xde98: pop     rbp
0xde99: jmp     0xc5e0
0xc5e0
0xde9e: nop
0xdea0: endbr64
0xdea4: push    rbp
0xdea5: mov     rbp, rsp
0xdea8: push    r15
0xdeaa: push    r14
0xdeac: push    r13
0xdeae: push    r12
0xdeb0: push    rbx
0xdeb1: sub     rsp, 0x68
0xdeb5: mov     rax, qword ptr fs:[0x28]
0xdeb6: mov     qword ptr [rbp - 0x38], rax
0xdec2: xor     eax, eax
0xdec4: lea     rax, [rbp - 0x61]
0xdec8: mov     rdi, rax
0xdecb: mov     qword ptr [rbp - 0x80], rax
0xdecf: call    0xa6a0
0xded4: mov     rax, qword ptr [rip + 0x1d6ed]
0xdedb: mov     esi, 0x20
0xdee0: mov     rdi, qword ptr [rip + 0x1d6c9]
0xdee7: movsxd  rdx, dword ptr [rax + 0xc]
0xdeeb: call    0x4a50
0xdef0: mov     rdi, qword ptr [rip + 0x1d6c9]
0xdef7: call    0x49b0
0xdefc: lea     r14, [rax + 1]
0xdf00: mov     rdi, r14
0xdf03: call    0x1d670
0xdf08: mov     rcx, qword ptr [rip + 0x1d6b1]
0xdf0f: mov     r13, rax
0xdf12: movzx   eax, byte ptr [rcx]
0xdf15: test    al, al
0xdf17: je      0xe430

```

```

0xdf1d: xor     esi, esi
0xdf1f: xor     ebx, ebx
0xdf21: lea     r15, [rip + 0x13188]
0xdf28: xor     edx, edx
0xdf2a: mov     dword ptr [rbp - 0x74], esi
0xdf2d: jmp     0xdf48
0xdf48:
0xdf2f: nop
0xdf30: cmp     al, 9
0xdf32: je      0xdf80
0xdf34: add     dword ptr [rbp - 0x74], 1
0xdf38: mov     byte ptr [r13 + r12], al
0xdf3d: movsxd  rdx, ebx
0xdf40: movzx   eax, byte ptr [rcx + rdx]
0xdf44: test    al, al
0xdf46: je      0xdfc0
0xdf48: movsxd  r12, dword ptr [rbp - 0x74]
0xdf4c: add     ebx, 1
0xdf4f: cmp     al, 0x5c
0xdf51: jne     0xdf30
0xdf53: lea     rdx, [rcx + rdx + 1]
0xdf58: movzx   eax, byte ptr [rdx]
0xdf5b: test    al, al
0xdf5d: je      0xe320
0xdf63: sub     eax, 0x5c
0xdf66: cmp     al, 0x17
0xdf68: ja      0xe0d8
0xdf6e: movzx   eax, al
0xdf71: movsxd  rax, dword ptr [r15 + rax*4]
0xdf75: add     rax, r15
0xdf78: jmp     rax
rax
->->->                                indirect JMP
0x107c6: cmp     byte ptr [rax + 0x48], 4
0x107ca: ja      0x4f20
0x107d0: movzx   edx, byte ptr [rax + 0x48]
0x107d4: lea     rcx, [rip + 0x11101]
0x107db: movsxd  rdx, dword ptr [rcx + rdx*4]
0x107df: add     rdx, rcx
0x107e2: jmp     rdx
rdx
->->->                                indirect JMP
0x10760: push    r15
0x10762: push    r14
0x10764: mov     r14d, 4
0x1076a: push    r13
0x1076c: mov     r13, rdi
0x1076f: push    r12

```

```

0x10771:    push    rbp
0x10772:    push    rbx
0x10773:    mov     ebx, esi
0x10775:    mov     esi, 0x20
0x1077a:    sub     rsp, 0x28
0x1077e:    movsxd  rdx, dword ptr [rdi + 0x64]
0x10782:    mov     rdi, qword ptr [rdi + 0x20]
0x10786:    call    0x4a50
0x1078b:    mov     ebp, dword ptr [r13 + 0x64]
0x1078f:    cmp     ebp, 0x27
0x10792:    jle     0x107b0
0x10794:    movsxd  rax, dword ptr [r13 + 0x80]
0x1079b:    cmp     ebp, 0x49
0x1079e:    jg      0x107a5
0x107a0:    cmp     eax, 5
0x107a3:    je      0x107b0
0x107a5:    lea     rdx, [rip + 0x11234]
0x107ac:    movsxd  r14, dword ptr [rdx + rax*4]
0x107b0:    movsxd  rax, ebx
0x107b3:    lea     r12, [rax + rax*4]
0x107b7:    mov     rax, qword ptr [r13 + 0x38]
0x107bb:    shl     r12, 4
0x107bf:    add     rax, r12
0x107c2:    test    ebx, ebx
0x107c4:    jle     0x107e8
0x107c6:    cmp     byte ptr [rax + 0x48], 4
0x107ca:    ja      0x4f20
0x107d0:    movzx   edx, byte ptr [rax + 0x48]
0x107d4:    lea     rcx, [rip + 0x11101]
0x107db:    movsxd  rdx, dword ptr [rcx + rdx*4]
0x107df:    add     rdx, rcx
0x107e2:    jmp     rdx
rdx
->->->                                indirect JMP
0x12e40:    mov     rax, qword ptr [rsp + 0x28]
0x12e45:    sub     rax, qword ptr fs:[0x28]
0x12e4e:    jne     0x12fbc
0x12e54:    add     rsp, 0x38
0x12e58:    mov     edx, 0x1c
0x12e5d:    xor     esi, esi
0x12e5f:    pop     rbx
0x12e60:    lea     rdi, [rip + 0xec11]
0x12e67:    pop     rbp
0x12e68:    pop     r12
0x12e6a:    pop     r13
0x12e6c:    pop     r14
0x12e6e:    pop     r15
0x12e70:    jmp     0x8070

```

```

0x8070
0x12e75:    nop        dword ptr [rax]
0x12e78:    mov        rax, qword ptr [rbx + 0x38]
0x12e7c:    mov        rsi, qword ptr [rax + r12]
0x12e80:    mov        ecx, 0x1e
0x12e85:    xor        edx, edx
0x12e87:    lea        rdi, [rip + 0xec4a]
0x12e8e:    xor        eax, eax
0x12e90:    call       0xf760
0x12e95:    jmp        0x12d6f
0x12d6f
0x12e9a:    nop        word ptr [rax + rax]
0x12ea0:    mov        rax, qword ptr [rbp + 0x38]
0x12ea4:    mov        rsi, qword ptr [rax + r10]
0x12ea8:    jmp        0x12e80
0x12e80
0x12eaa:    nop        word ptr [rax + rax]
0x12eb0:    mov        rax, qword ptr [rbx + 0x38]
0x12eb4:    mov        rsi, qword ptr [rax + r12]
0x12eb8:    mov        ecx, 0x1e
0x12ebd:    xor        edx, edx
0x12ebf:    lea        rdi, [rip + 0xe423]
0x12ec6:    xor        eax, eax
0x12ec8:    call       0xf760
0x12ecd:    jmp        0x12d6f
0x12d6f
0x12ed2:    nop        word ptr [rax + rax]
0x12ed8:    mov        rax, qword ptr [rbp + 0x38]
0x12edc:    mov        rsi, qword ptr [rax + r10]
0x12ee0:    jmp        0x12eb8
0x12eb8
0x12ee2:    nop        word ptr [rax + rax]
0x12ee8:    mov        edx, 0xc
0x12eed:    lea        rsi, [rip + 0xd63e]
0x12ef4:    mov        qword ptr [rsp + 8], r10
0x12ef9:    lea        rdi, [rip + 0xebb0]
0x12f00:    call       0x8070
0x12f05:    mov        r10, qword ptr [rsp + 8]
0x12f0a:    cmp        al, 0x79
0x12f0c:    jne        0x12d6f
0x12f12:    jmp        0x12df2
0x12df2
0x12f17:    mov        rax, qword ptr [rsp + 0x18]
0x12f1c:    mov        rdx, qword ptr [rsp + 0x20]
0x12f21:    cmp        rax, r13
0x12f24:    je         0x12f87
0x12f26:    cmp        rax, rdx
0x12f29:    cmovg     rax, rdx

```



```

0x12f2d:      cmp     rax, r13
0x12f30:      je      0x12fa4
0x12f32:      mov     edi, 0x80
0x12f37:      call    0x1d670
0x12f3c:      mov     r9, r13
0x12f3f:      mov     r8, r13
0x12f42:      mov     esi, 1
0x12f47:      mov     rbp, rax
0x12f4a:      mov     rdi, rax
0x12f4d:      xor     eax, eax
0x12f4f:      mov     rdx, -1
0x12f56:      lea     rcx, [rip + 0xe3af]
0x12f5d:      call    0x4f10
0x12f62:      mov     ecx, 0x14
0x12f67:      xor     edx, edx
0x12f69:      mov     rsi, rbp
0x12f6c:      lea     rdi, [rip + 0xe3a5]
0x12f73:      xor     eax, eax
0x12f75:      call    0xf760
0x12f7a:      mov     rdi, rbp
0x12f7d:      call    0x18d10
0x12f82:      jmp     0x12d6f
0x12d6f
0x12f87:      cmp     rax, rdx
0x12f8a:      jne     0x12f26
0x12f8c:      mov     edx, 0x14
0x12f91:      xor     esi, esi
0x12f93:      lea     rdi, [rip + 0xe365]
0x12f9a:      call    0x8070
0x12f9f:      jmp     0x12d6f
0x12d6f
0x12fa4:      mov     edx, 0x14
0x12fa9:      xor     esi, esi
0x12fab:      lea     rdi, [rip + 0xeb4e]
0x12fb2:      call    0x8070
0x12fb7:      jmp     0x12d6f
0x12d6f
0x12fbc:      call    0x49d0
0x12fc1:      nop     word ptr cs:[rax + rax]
0x12fcc:      nop     dword ptr [rax]
0x12fd0:      endbr64
0x12fd4:      push    r15
0x12fd6:      push    r14
0x12fd8:      push    r13
0x12fda:      push    r12
0x12fdc:      mov     r12, rdi
0x12fdf:      push    rbp
0x12fe0:      mov     rbp, rsi

```

```

0x12fe3:    push    rbx
0x12fe4:    sub     rsp, 0x48
0x12fe8:    mov     rdi, qword ptr [rip + 0x174d9]
0x12fef:    mov     dword ptr [rsp + 0x14], edx
0x12ff3:    mov     rax, qword ptr fs:[0x28]
0x12ffc:    mov     qword ptr [rsp + 0x38], rax
0x13001:    xor     eax, eax
0x13003:    call    0xdb60
0x13008:    xor     eax, eax
0x1300a:    call    0xa820
0x1300f:    mov     rsi, qword ptr [rbp + 0x18]
0x13013:    mov     rdi, qword ptr [r12 + 0x18]
0x13018:    call    0x4b30
0x1301d:    test    eax, eax
0x1301f:    je      0x132d0
0x13025:    mov     ecx, dword ptr [r12 + 0x68]
0x1302a:    test    ecx, ecx
0x1302c:    jle     0x1306d
0x1302e:    lea     edx, [rcx - 1]
0x13031:    mov     rsi, qword ptr [r12 + 0x38]
0x13036:    lea     rdx, [rdx + rdx*4]
0x1303a:    shl     rdx, 4
0x1303e:    lea     rax, [rsi + 0x49]
0x13042:    lea     rdx, [rsi + rdx + 0x99]
0x1304a:    nop     word ptr [rax + rax]
0x13050:    cmp     byte ptr [rax], 0
0x13053:    jne     0x13064
0x13055:    cmp     byte ptr [rax - 1], 0
0x13059:    je      0x13064
0x1305b:    mov     byte ptr [rax], 1
0x1305e:    add     dword ptr [r12 + 0x70], 1
0x13064:    add     rax, 0x50
0x13068:    cmp     rax, rdx
0x1306b:    jne     0x13050
0x1306d:    mov     edx, dword ptr [rbp + 0x68]
0x13070:    test    edx, edx
0x13072:    jle     0x130ab
0x13074:    sub     edx, 1
0x13077:    mov     rsi, qword ptr [rbp + 0x38]
0x1307b:    lea     rdx, [rdx + rdx*4]
0x1307f:    shl     rdx, 4
0x13083:    lea     rax, [rsi + 0x49]
0x13087:    lea     rdx, [rsi + rdx + 0x99]
0x1308f:    nop
0x13090:    cmp     byte ptr [rax], 0
0x13093:    jne     0x130a2
0x13095:    cmp     byte ptr [rax - 1], 0
0x13099:    je      0x130a2

```

```

0x1309b:    mov     byte ptr [rax], 1
0x1309e:    add     dword ptr [rbp + 0x70], 1
0x130a2:    add     rax, 0x50
0x130a6:    cmp     rax, rdx
0x130a9:    jne     0x13090
0x130ab:    lea     rax, [rsp + 0x30]
0x130b0:    mov     dword ptr [rsp + 0x10], 1
0x130b8:    mov     ebx, 0x50
0x130bd:    mov     qword ptr [rsp + 0x18], rax
0x130c2:    cmp     ecx, 1
0x130c5:    jg      0x130e8
0x130c7:    jmp     0x13188
0x13188:
0x130cc:    nop     dword ptr [rax]
0x130d0:    add     dword ptr [rsp + 0x10], 1
0x130d5:    add     rbx, 0x50
0x130d9:    mov     eax, dword ptr [rsp + 0x10]
0x130dd:    cmp     dword ptr [r12 + 0x68], eax
0x130e2:    jle     0x13188
0x130e8:    mov     rax, qword ptr [r12 + 0x38]
0x130ed:    cmp     byte ptr [rax + rbx + 0x48], 1
0x130f2:    jne     0x130d0
0x130f4:    cmp     dword ptr [rbp + 0x68], 1
0x130f8:    jle     0x130d0
0x130fa:    mov     r14d, 0x50
0x13100:    mov     r13d, 1
0x13106:    jmp     0x1311e
0x1311e:
0x13108:    nop     dword ptr [rax + rax]
0x13110:    add     r13d, 1
0x13114:    add     r14, 0x50
0x13118:    cmp     dword ptr [rbp + 0x68], r13d
0x1311c:    jle     0x130d0
0x1311e:    xor     eax, eax
0x13120:    call    0x18ab0
0x13125:    mov     r15, qword ptr [rbp + 0x38]
0x13129:    add     r15, r14
0x1312c:    cmp     byte ptr [r15 + 0x48], 1
0x13131:    jne     0x13110
0x13133:    mov     rdx, qword ptr [r12 + 0x38]
0x13138:    mov     rsi, qword ptr [r15]
0x1313b:    add     rdx, rbx
0x1313e:    mov     rdi, qword ptr [rdx]
0x13141:    mov     qword ptr [rsp + 8], rdx
0x13146:    call    0x4b30
0x1314b:    mov     rdx, qword ptr [rsp + 8]
0x13150:    test    eax, eax
0x13152:    jne     0x13110

```

```

0x13154:    mov     rax, qword ptr [r15 + 8]
0x13158:    cmp     qword ptr [rdx + 8], rax
0x1315c:    je      0x13440
0x13162:    mov     rsi, qword ptr [rdx + 0x20]
0x13166:    mov     rax, qword ptr [r15 + 0x20]
0x1316a:    cmp     rsi, rax
0x1316d:    jl      0x133f3
0x13173:    mov     byte ptr [r15 + 0x49], 0
0x13178:    sub     dword ptr [rbp + 0x70], 1
0x1317c:    jmp     0x130d0
0x130d0:
0x13181:    nop     dword ptr [rax]
0x13188:    xor     eax, eax
0x1318a:    call    0xe4e0
0x1318f:    xor     eax, eax
0x13191:    mov     rdi, r12
0x13194:    call    0x10ec0
0x13199:    xor     eax, eax
0x1319b:    mov     rdi, r12
0x1319e:    call    0x10160
0x131a3:    cmp     byte ptr [r12 + 0x6d], 0
0x131a9:    jne     0x13290
0x131af:    xor     eax, eax
0x131b1:    mov     rdi, r12
0x131b4:    call    0x10240
0x131b9:    cmp     byte ptr [r12 + 0x6d], 0
0x131bf:    jne     0x13280
0x131c5:    xor     eax, eax
0x131c7:    mov     rdi, rbp
0x131ca:    call    0x10ec0
0x131cf:    xor     eax, eax
0x131d1:    mov     rdi, rbp
0x131d4:    call    0x10160
0x131d9:    cmp     byte ptr [rbp + 0x6d], 0
0x131dd:    jne     0x13258
0x131df:    xor     eax, eax
0x131e1:    mov     rdi, rbp
0x131e4:    call    0x10240
0x131e9:    cmp     byte ptr [rbp + 0x6d], 0
0x131ed:    jne     0x13238
0x131ef:    xor     eax, eax
0x131f1:    call    0xa820
0x131f6:    mov     edx, dword ptr [r12 + 0x70]
0x131fb:    test    edx, edx
0x131fd:    jne     0x132a0
0x13203:    mov     eax, dword ptr [rbp + 0x70]
0x13206:    test    eax, eax
0x13208:    jne     0x132a0

```

```

0x1320e:    mov     rax, qword ptr [rsp + 0x38]
0x13213:    sub     rax, qword ptr fs:[0x28]
0x1321c:    jne     0x134c3
0x13222:    add     rsp, 0x48
0x13226:    pop     rbx
0x13227:    pop     rbp
0x13228:    pop     r12
0x1322a:    pop     r13
0x1322c:    pop     r14
0x1322e:    pop     r15
0x13230:    ret
0x13231:    nop     dword ptr [rax]
0x13238:    mov     rdi, rbp
0x1323b:    xor     eax, eax
0x1323d:    call    0x10d70
0x13242:    xor     eax, eax
0x13244:    call    0xa820
0x13249:    mov     edx, dword ptr [r12 + 0x70]
0x1324e:    test    edx, edx
0x13250:    je      0x13203
0x13252:    jmp     0x132a0
0x132a0:
0x13254:    nop     dword ptr [rax]
0x13258:    xor     eax, eax
0x1325a:    mov     rdi, rbp
0x1325d:    call    0xec50
0x13262:    xor     eax, eax
0x13264:    mov     rdi, rbp
0x13267:    call    0x10240
0x1326c:    cmp     byte ptr [rbp + 0x6d], 0
0x13270:    je      0x131ef
0x13276:    jmp     0x13238
0x13238:
0x13278:    nop     dword ptr [rax + rax]
0x13280:    mov     rdi, r12
0x13283:    xor     eax, eax
0x13285:    call    0x10d70
0x1328a:    jmp     0x131c5
0x131c5:
0x1328f:    nop
0x13290:    mov     rdi, r12
0x13293:    xor     eax, eax
0x13295:    call    0xec50
0x1329a:    jmp     0x131af
0x131af:
0x1329f:    nop
0x132a0:    mov     rax, qword ptr [rsp + 0x38]
0x132a5:    sub     rax, qword ptr fs:[0x28]

```

```

0x132ae:    jne     0x134c3
0x132b4:    add     rsp, 0x48
0x132b8:    xor     eax, eax
0x132ba:    pop     rbx
0x132bb:    pop     rbp
0x132bc:    pop     r12
0x132be:    pop     r13
0x132c0:    pop     r14
0x132c2:    pop     r15
0x132c4:    jmp     0xa5a0
0xa5a0
0x132c9:    nop     dword ptr [rax]
0x132d0:    mov     edx, 0x1c
0x132d5:    xor     esi, esi
0x132d7:    lea     rdi, [rip + 0xe89a]
0x132de:    call    0x8070
0x132e3:    mov     edx, dword ptr [r12 + 0x68]
0x132e8:    test    edx, edx
0x132ea:    jle     0x1331c
0x132ec:    sub     edx, 1
0x132ef:    mov     rcx, qword ptr [r12 + 0x38]
0x132f4:    lea     rdx, [rdx + rdx*4]
0x132f8:    shl     rdx, 4
0x132fc:    lea     rax, [rcx + 0x49]
0x13300:    lea     rdx, [rcx + rdx + 0x99]
0x13308:    nop     dword ptr [rax + rax]
0x13310:    mov     byte ptr [rax], 0
0x13313:    add     rax, 0x50
0x13317:    cmp     rdx, rax
0x1331a:    jne     0x13310
0x1331c:    mov     dword ptr [r12 + 0x70], 0
0x13325:    mov     edx, dword ptr [rbp + 0x68]
0x13328:    test    edx, edx
0x1332a:    jle     0x1335c
0x1332c:    sub     edx, 1
0x1332f:    mov     rcx, qword ptr [rbp + 0x38]
0x13333:    lea     rdx, [rdx + rdx*4]
0x13337:    shl     rdx, 4
0x1333b:    lea     rax, [rcx + 0x49]
0x1333f:    lea     rdx, [rcx + rdx + 0x99]
0x13347:    nop     word ptr [rax + rax]
0x13350:    mov     byte ptr [rax], 0
0x13353:    add     rax, 0x50
0x13357:    cmp     rdx, rax
0x1335a:    jne     0x13350
0x1335c:    mov     dword ptr [rbp + 0x70], 0
0x13363:    xor     eax, eax
0x13365:    mov     rdi, r12

```

```

0x13368:      call    0x10ec0
0x1336d:      xor     eax, eax
0x1336f:      mov     rdi, r12
0x13372:      call    0x10160
0x13377:      cmp     byte ptr [r12 + 0x6d], 0
0x1337d:      jne     0x13431
0x13383:      xor     eax, eax
0x13385:      mov     rdi, r12
0x13388:      call    0x10240
0x1338d:      cmp     byte ptr [r12 + 0x6d], 0
0x13393:      jne     0x13422
0x13399:      xor     eax, eax
0x1339b:      mov     rdi, rbp
0x1339e:      call    0x10ec0
0x133a3:      xor     eax, eax
0x133a5:      mov     rdi, rbp
0x133a8:      call    0x10160
0x133ad:      cmp     byte ptr [rbp + 0x6d], 0
0x133b1:      jne     0x13402
0x133b3:      xor     eax, eax
0x133b5:      mov     rdi, rbp
0x133b8:      call    0x10240
0x133bd:      cmp     byte ptr [rbp + 0x6d], 0
0x133c1:      je      0x1320e
0x133c7:      mov     rax, qword ptr [rsp + 0x38]
0x133cc:      sub     rax, qword ptr fs:[0x28]
0x133d5:      jne     0x134c3
0x133db:      add     rsp, 0x48
0x133df:      mov     rdi, rbp
0x133e2:      xor     eax, eax
0x133e4:      pop     rbx
0x133e5:      pop     rbp
0x133e6:      pop     r12
0x133e8:      pop     r13
0x133ea:      pop     r14
0x133ec:      pop     r15
0x133ee:      jmp     0x10d70
0x10d70
0x133f3:      mov     byte ptr [rdx + 0x49], 0
0x133f7:      sub     dword ptr [r12 + 0x70], 1
0x133fd:      jmp     0x130d0
0x130d0
0x13402:      xor     eax, eax
0x13404:      mov     rdi, rbp
0x13407:      call    0xec50
0x1340c:      xor     eax, eax
0x1340e:      mov     rdi, rbp
0x13411:      call    0x10240

```

```

0x13416:      cmp     byte ptr [rbp + 0x6d], 0
0x1341a:      je      0x1320e
0x13420:      jmp     0x133c7
0x133c7:
0x13422:      mov     rdi, r12
0x13425:      xor     eax, eax
0x13427:      call    0x10d70
0x1342c:      jmp     0x13399
0x13399:
0x13431:      mov     rdi, r12
0x13434:      xor     eax, eax
0x13436:      call    0xec50
0x1343b:      jmp     0x13383
0x13383:
0x13440:      mov     ecx, dword ptr [rsp + 0x14]
0x13444:      test    ecx, ecx
0x13446:      je      0x13468
0x13448:      mov     rsi, qword ptr [rdx + 0x20]
0x1344c:      mov     rax, qword ptr [r15 + 0x20]
0x13450:      cmp     rsi, rax
0x13453:      jne     0x1316a
0x13459:      mov     byte ptr [rdx + 0x49], 0
0x1345d:      sub     dword ptr [r12 + 0x70], 1
0x13463:      jmp     0x13173
0x13173:
0x13468:      mov     r9, qword ptr [rsp + 0x18]
0x1346d:      mov     esi, dword ptr [rsp + 0x10]
0x13471:      xor     eax, eax
0x13473:      lea     rdx, [rsp + 0x28]
0x13478:      mov     r8d, r13d
0x1347b:      mov     rcx, rbp
0x1347e:      mov     rdi, r12
0x13481:      call    0x125c0
0x13486:      cmp     rax, -1
0x1348a:      je      0x13188
0x13490:      mov     rdx, qword ptr [r12 + 0x38]
0x13495:      add     rdx, rbx
0x13498:      cmp     qword ptr [rsp + 0x28], rax
0x1349d:      je      0x134b3
0x1349f:      add     r14, qword ptr [rbp + 0x38]
0x134a3:      mov     rsi, qword ptr [rdx + 0x20]
0x134a7:      mov     rax, qword ptr [r14 + 0x20]
0x134ab:      mov     r15, r14
0x134ae:      jmp     0x1316a
0x1316a:
0x134b3:      cmp     qword ptr [rsp + 0x30], rax
0x134b8:      jne     0x1349f
0x134ba:      add     r14, qword ptr [rbp + 0x38]

```



```

0x134be:    mov     r15, r14
0x134c1:    jmp     0x13459
0x13459:
0x134c3:    call    0x49d0
0x134c8:    nop     dword ptr [rax + rax]
0x134d0:    endbr64
0x134d4:    push    r15
0x134d6:    movsxd  rsi, esi
0x134d9:    mov     r15d, edx
0x134dc:    push    r14
0x134de:    push    r13
0x134e0:    lea     r13, [rsi + rsi*4]
0x134e4:    push    r12
0x134e6:    shl     r13, 4
0x134ea:    mov     r12, rdi
0x134ed:    push    rbp
0x134ee:    push    rbx
0x134ef:    sub     rsp, 8
0x134f3:    mov     rax, qword ptr [rdi + 0x38]
0x134f7:    mov     r14, qword ptr [rax + r13]
0x134fb:    mov     rdi, r14
0x134fe:    call    0x1d7a0
0x13503:    mov     rdi, r14
0x13506:    mov     rbp, rax
0x13509:    call    0x49b0
0x1350e:    mov     rbx, rax
0x13511:    test    r15d, r15d
0x13514:    jne     0x13580
0x13516:    test    eax, eax
0x13518:    jle     0x13543
0x1351a:    call    0x4ef0
0x1351f:    lea     edi, [rbx - 1]
0x13522:    xor     edx, edx
0x13524:    nop     dword ptr [rax]
0x13528:    movsx   rsi, byte ptr [r14 + rdx]
0x1352d:    mov     rcx, qword ptr [rax]
0x13530:    mov     ecx, dword ptr [rcx + rsi*4]
0x13533:    mov     byte ptr [rbp + rdx], cl
0x13537:    mov     rcx, rdx
0x1353a:    add     rdx, 1
0x1353e:    cmp     rdi, rcx
0x13541:    jne     0x13528
0x13543:    mov     rsi, rbp
0x13546:    mov     rdi, r14
0x13549:    call    0x4b30
0x1354e:    test    eax, eax
0x13550:    jne     0x135ba
0x13552:    mov     rax, qword ptr [r12 + 0x38]

```

```

0x13557:    mov     byte ptr [rax + r13 + 0x49], 0
0x1355d:    mov     eax, 1
0x13562:    sub     dword ptr [r12 + 0x70], 1
0x13568:    add     rsp, 8
0x1356c:    pop     rbx
0x1356d:    pop     rbp
0x1356e:    pop     r12
0x13570:    pop     r13
0x13572:    pop     r14
0x13574:    pop     r15
0x13576:    ret
0x13577:    nop     word ptr [rax + rax]
0x13580:    test    eax, eax
0x13582:    jle     0x13543
0x13584:    call    0x47c0
0x13589:    lea     edi, [rbx - 1]
0x1358c:    xor     edx, edx
0x1358e:    nop
0x13590:    movsx   rsi, byte ptr [r14 + rdx]
0x13595:    mov     rcx, qword ptr [rax]
0x13598:    mov     ecx, dword ptr [rcx + rsi*4]
0x1359b:    mov     byte ptr [rbp + rdx], cl
0x1359f:    mov     rcx, rdx
0x135a2:    add     rdx, 1
0x135a6:    cmp     rcx, rdi
0x135a9:    jne     0x13590
0x135ab:    mov     rsi, rbp
0x135ae:    mov     rdi, r14
0x135b1:    call    0x4b30
0x135b6:    test    eax, eax
0x135b8:    je      0x13552
0x135ba:    lea     edi, [rbx + 0x21]
0x135bd:    mov     ebx, 1
0x135c2:    movsxd  rdi, edi
0x135c5:    call    0x1d670
0x135ca:    mov     r8, r14
0x135cd:    mov     esi, 1
0x135d2:    lea     rcx, [rip + 0xdd5b]
0x135d9:    mov     r15, rax
0x135dc:    mov     rdi, rax
0x135df:    mov     rdx, -1
0x135e6:    xor     eax, eax
0x135e8:    call    0x4f10
0x135ed:    mov     edx, 1
0x135f2:    mov     esi, 1
0x135f7:    mov     rdi, r15
0x135fa:    call    0xe480
0x135ff:    xor     eax, eax

```

```

0x13601:      call    0xa820
0x13606:      mov     rdi, r15
0x13609:      call    0x18d10
0x1360e:      mov     rax, qword ptr [r12 + 0x38]
0x13613:      cmp     byte ptr [rax + r13 + 0x48], 0
0x13619:      je      0x13698
0x1361b:      mov     rsi, rbp
0x1361e:      mov     rdi, r14
0x13621:      call    0x4e10
0x13626:      test    eax, eax
0x13628:      je      0x137a5
0x1362e:      call    0x4860
0x13633:      mov     r15d, dword ptr [rax]
0x13636:      xor     eax, eax
0x13638:      call    0xe4e0
0x1363d:      xor     eax, eax
0x1363f:      call    0xa820
0x13644:      cmp     r15d, 1
0x13648:      sete    al
0x1364b:      cmp     r15d, 0xd
0x1364f:      sete    dl
0x13652:      or      al, dl
0x13654:      jne     0x13745
0x1365a:      and     ebx, 1
0x1365d:      je      0x13745
0x13663:      cmp     dword ptr [r12 + 0x90], 1
0x1366c:      je      0x13760
0x13672:      mov     rdi, rbp
0x13675:      call    0x4870
0x1367a:      mov     ebx, eax
0x1367c:      test    eax, eax
0x1367e:      je      0x1361b
0x13680:      add     rsp, 8
0x13684:      mov     eax, 5
0x13689:      pop     rbx
0x1368a:      pop     rbp
0x1368b:      pop     r12
0x1368d:      pop     r13
0x1368f:      pop     r14
0x13691:      pop     r15
0x13693:      ret
0x13694:      nop     dword ptr [rax]
0x13698:      xor     esi, esi
0x1369a:      mov     rdi, rbp
0x1369d:      call    0x4d90
0x136a2:      test    eax, eax
0x136a4:      jne     0x13732
0x136aa:      cmp     dword ptr [r12 + 0x90], 1

```

```

0x136b3:    je      0x1378b
0x136b9:    mov     rdi, rbp
0x136bc:    call    0x49b0
0x136c1:    lea     rdi, [rax + 0x21]
0x136c5:    call    0x1d670
0x136ca:    mov     rdx, -1
0x136d1:    mov     r8, rbp
0x136d4:    lea     rcx, [rip + 0xdc6c]
0x136db:    mov     r15, rax
0x136de:    mov     rdi, rax
0x136e1:    mov     esi, 1
0x136e6:    xor     eax, eax
0x136e8:    call    0x4f10
0x136ed:    mov     esi, 1
0x136f2:    mov     rdi, r15
0x136f5:    call    0x193f0
0x136fa:    mov     rdi, r15
0x136fd:    mov     ebx, eax
0x136ff:    call    0x18d10
0x13704:    mov     rdi, qword ptr [r12 + 0x18]
0x13709:    call    0xc4b0
0x1370e:    mov     edx, ebx
0x13710:    mov     eax, 7
0x13715:    and     edx, 0x7f
0x13718:    add     edx, 1
0x1371b:    cmp     dl, 1
0x1371e:    jg      0x13568
0x13724:    test    bl, 0x7f
0x13727:    jne     0x13680
0x1372d:    and     bh, 0xff
0x13730:    jne     0x1377a
0x13732:    mov     rsi, rbp
0x13735:    mov     rdi, r14
0x13738:    call    0x4dc0
0x1373d:    test    eax, eax
0x1373f:    je      0x13552
0x13745:    add     rsp, 8
0x13749:    mov     eax, 6
0x1374e:    pop     rbx
0x1374f:    pop     rbp
0x13750:    pop     r12
0x13752:    pop     r13
0x13754:    pop     r14
0x13756:    pop     r15
0x13758:    ret
0x13759:    nop     dword ptr [rax]
0x13760:    xor     eax, eax
0x13762:    mov     rsi, rbp

```

```

0x13765:      mov     rdi, r12
0x13768:      call    0x11220
0x1376d:      test    eax, eax
0x1376f:      je       0x13672
0x13775:      jmp      0x13568
0x13568
0x1377a:      lea      rdi, [rip + 0xd469]
0x13781:      call    0x195e0
0x13786:      jmp      0x13680
0x13680
0x1378b:      xor      eax, eax
0x1378d:      mov      rsi, rbp
0x13790:      mov      rdi, r12
0x13793:      call    0x11220
0x13798:      test    eax, eax
0x1379a:      je       0x136b9
0x137a0:      jmp      0x13568
0x13568
0x137a5:      mov      rdi, r14
0x137a8:      call    0x4870
0x137ad:      test    eax, eax
0x137af:      je       0x13552
0x137b5:      mov      eax, 4
0x137ba:      jmp      0x13568
0x13568
0x137bf:      nop
0x137c0:      endbr64
0x137c4:      push     r15
0x137c6:      push     r14
0x137c8:      mov      r14, rdi
0x137cb:      push     r13
0x137cd:      push     r12
0x137cf:      mov      r12, rdx
0x137d2:      push     rbp
0x137d3:      push     rbx
0x137d4:      sub      rsp, 0x118
0x137db:      mov      rax, qword ptr fs:[0x28]
0x137e4:      mov      qword ptr [rsp + 0x108], rax
0x137ec:      xor      eax, eax
0x137ee:      test     rdx, rdx
0x137f1:      je       0x13be4
0x137f7:      movsxd   rcx, dword ptr [r14 + 0x68]
0x137fb:      test     ecx, ecx
0x137fd:      jle      0x1382c
0x137ff:      lea      edx, [rcx - 1]
0x13802:      mov      rsi, qword ptr [r14 + 0x38]
0x13806:      lea      rdx, [rdx + rdx*4]
0x1380a:      shl      rdx, 4

```

```

0x1380e:    lea     rax, [rsi + 0x49]
0x13812:    lea     rdx, [rsi + rdx + 0x99]
0x1381a:    nop     word ptr [rax + rax]
0x13820:    mov     byte ptr [rax], 0
0x13823:    add     rax, 0x50
0x13827:    cmp     rax, rdx
0x1382a:    jne     0x13820
0x1382c:    mov     dword ptr [r14 + 0x70], 0
0x13834:    lea     rdi, [rcx*8]
0x1383c:    call    0x1d670
0x13841:    mov     qword ptr [rsp], rax
0x13845:    movsxd  rax, dword ptr [r14 + 0x68]
0x13849:    test    eax, eax
0x1384b:    jle     0x13c59
0x13851:    lea     rdi, [rax + rax*4]
0x13855:    xor     r8d, r8d
0x13858:    xor     eax, eax
0x1385a:    xor     r13d, r13d
0x1385d:    shl     rdi, 4
0x13861:    mov     r9d, 0x400
0x13867:    jmp     0x13879
0x13879:
0x13869:    nop     dword ptr [rax]
0x13870:    add     rax, 0x50
0x13874:    cmp     rdi, rax
0x13877:    je      0x138f8
0x13879:    mov     rcx, qword ptr [r14 + 0x38]
0x1387d:    add     rcx, rax
0x13880:    mov     edx, dword ptr [rcx + 0x10]
0x13883:    and     edx, 0xf000
0x13889:    cmp     edx, 0x8000
0x1388f:    jne     0x13870
0x13891:    mov     rdx, qword ptr [rcx + 8]
0x13895:    test    edx, 0x3ff
0x1389b:    je      0x138be
0x1389d:    mov     r10, rdx
0x138a0:    sar     r10, 0x3f
0x138a4:    shr     r10, 0x36
0x138a8:    lea     rsi, [rdx + r10]
0x138ac:    and     esi, 0x3ff
0x138b2:    sub     rsi, r10
0x138b5:    mov     r10, r9
0x138b8:    sub     r10, rsi
0x138bb:    add     rdx, r10
0x138be:    test    rdx, rdx
0x138c1:    lea     rsi, [rdx + 0x3ff]
0x138c8:    cmovs   rdx, rsi
0x138cc:    sar     rdx, 0xa

```

```

0x138d0:      cmp     r12, rdx
0x138d3:      jl      0x13af0
0x138d9:      mov     rbx, qword ptr [rsp]
0x138dd:      movsxd  rdx, r13d
0x138e0:      add     rax, 0x50
0x138e4:      add     r13d, 1
0x138e8:      mov     qword ptr [rbx + rdx*8], rcx
0x138ec:      cmp     rdi, rax
0x138ef:      jne     0x13879
0x138f1:      nop     dword ptr [rax]
0x138f8:      mov     rbx, qword ptr [rsp]
0x138fc:      movsxd  rsi, r13d
0x138ff:      mov     edx, 8
0x13904:      lea     rcx, [rip - 0x53cb]
0x1390b:      mov     dword ptr [rsp + 0x10], r8d
0x13910:      mov     rdi, rbx
0x13913:      call    0x48e0
0x13918:      test    r13d, r13d
0x1391b:      je      0x13c86
0x13921:      lea     eax, [r13 - 1]
0x13925:      mov     r8d, r13d
0x13928:      xor     r15d, r15d
0x1392b:      xor     r11d, r11d
0x1392e:      lea     r10, [rbx + rax*8 + 8]
0x13933:      mov     rax, r12
0x13936:      xor     r9d, r9d
0x13939:      mov     edx, 0x400
0x1393e:      mov     r13, r10
0x13941:      mov     r12, rbx
0x13944:      mov     rcx, rax
0x13947:      nop     word ptr [rax + rax]
0x13950:      mov     rdi, qword ptr [r12]
0x13954:      mov     rax, qword ptr [rdi + 8]
0x13958:      test    eax, 0x3ff
0x1395d:      je      0x13980
0x1395f:      mov     r10, rax
0x13962:      sar     r10, 0x3f
0x13966:      shr     r10, 0x36
0x1396a:      lea     rsi, [rax + r10]
0x1396e:      and     esi, 0x3ff
0x13974:      sub     rsi, r10
0x13977:      mov     r10, rdx
0x1397a:      sub     r10, rsi
0x1397d:      add     rax, r10
0x13980:      test    rax, rax
0x13983:      lea     rbp, [rax + 0x3ff]
0x1398a:      cmovns  rbp, rax
0x1398e:      sar     rbp, 0xa

```

```

0x13992:      test    r15d, r15d
0x13995:      jle     0x13af9
0x1399b:      mov     rax, r9
0x1399e:      xor     ebx, ebx
0x139a0:      jmp     0x139b8
0x139b8:
0x139a2:      nop     word ptr [rax + rax]
0x139a8:      add     ebx, 1
0x139ab:      add     rax, 8
0x139af:      cmp     ebx, r15d
0x139b2:      je      0x13b10
0x139b8:      mov     rsi, qword ptr [rax]
0x139bb:      cmp     rsi, rbp
0x139be:      jl      0x139a8
0x139c0:      sub     rsi, rbp
0x139c3:      mov     qword ptr [rax], rsi
0x139c6:      test    ebx, ebx
0x139c8:      jne     0x13b00
0x139ce:      mov     byte ptr [rdi + 0x49], 1
0x139d2:      add     dword ptr [r14 + 0x70], 1
0x139d7:      add     r12, 8
0x139db:      cmp     r13, r12
0x139de:      jne     0x13950
0x139e4:      mov     r13d, r8d
0x139e7:      mov     r12, rcx
0x139ea:      xor     eax, eax
0x139ec:      mov     rdi, r14
0x139ef:      mov     qword ptr [rsp + 8], r9
0x139f4:      call    0x10ec0
0x139f9:      xor     eax, eax
0x139fb:      mov     rdi, r14
0x139fe:      call    0x10160
0x13a03:      cmp     byte ptr [r14 + 0x6d], 0
0x13a08:      mov     r9, qword ptr [rsp + 8]
0x13a0d:      je      0x13a1e
0x13a0f:      mov     rdi, r14
0x13a12:      xor     eax, eax
0x13a14:      call    0xec50
0x13a19:      mov     r9, qword ptr [rsp + 8]
0x13a1e:      xor     eax, eax
0x13a20:      mov     rdi, r14
0x13a23:      mov     qword ptr [rsp + 8], r9
0x13a28:      call    0x10240
0x13a2d:      cmp     byte ptr [r14 + 0x6d], 0
0x13a32:      mov     r9, qword ptr [rsp + 8]
0x13a37:      je      0x13a48
0x13a39:      mov     rdi, r14
0x13a3c:      xor     eax, eax

```



```

0x13a3e:    call    0x10d70
0x13a43:    mov     r9, qword ptr [rsp + 8]
0x13a48:    test    r9, r9
0x13a4b:    je      0x13a55
0x13a4d:    mov     rdi, r9
0x13a50:    call    0x18d10
0x13a55:    mov     rdi, qword ptr [rsp]
0x13a59:    call    0x18d10
0x13a5e:    test    r13d, r13d
0x13a61:    je      0x13bad
0x13a67:    mov     eax, dword ptr [rsp + 0x10]
0x13a6b:    lea     rbp, [rsp + 0x60]
0x13a70:    test    eax, eax
0x13a72:    je      0x13b7b
0x13a78:    sub     rsp, 8
0x13a7c:    mov     r9d, r13d
0x13a7f:    mov     edx, 0xa0
0x13a84:    mov     rdi, rbp
0x13a87:    push    rax
0x13a88:    lea     rax, [rip + 0xd8d2]
0x13a8f:    mov     esi, 1
0x13a94:    lea     r8, [rip + 0xd8ba]
0x13a9b:    push    r15
0x13a9d:    lea     rcx, [rip + 0xe104]
0x13aa4:    push    rax
0x13aa5:    xor     eax, eax
0x13aa7:    call    0x4f10
0x13aac:    add     rsp, 0x20
0x13ab0:    xor     esi, esi
0x13ab2:    mov     edx, 0x36
0x13ab7:    mov     rdi, rbp
0x13aba:    call    0x8070
0x13abf:    mov     rax, qword ptr [rsp + 0x108]
0x13ac7:    sub     rax, qword ptr fs:[0x28]
0x13ad0:    jne     0x13c91
0x13ad6:    add     rsp, 0x118
0x13add:    pop     rbx
0x13ade:    pop     rbp
0x13adf:    pop     r12
0x13ae1:    pop     r13
0x13ae3:    pop     r14
0x13ae5:    pop     r15
0x13ae7:    ret
0x13ae8:    nop     dword ptr [rax + rax]
0x13af0:    add     r8d, 1
0x13af4:    jmp     0x13870
0x13870:
0x13af9:    xor     ebx, ebx

```

```

0x13afb:      nop      dword ptr [rax + rax]
0x13b00:      cmp      r15d, ebx
0x13b03:      jne      0x139d7
0x13b09:      nop      dword ptr [rax]
0x13b10:      cmp      r11d, r15d
0x13b13:      je       0x13b40
0x13b15:      mov      rsi, rcx
0x13b18:      movsxd   rax, r15d
0x13b1b:      add      r15d, 1
0x13b1f:      sub      rsi, rbp
0x13b22:      mov      qword ptr [r9 + rax*8], rsi
0x13b26:      test     ebx, ebx
0x13b28:      jne      0x139d7
0x13b2e:      mov      rax, qword ptr [r12]
0x13b32:      mov      byte ptr [rax + 0x49], 1
0x13b36:      add      dword ptr [r14 + 0x70], 1
0x13b3b:      jmp      0x139d7
0x139d7:
0x13b40:      add      r11d, 0x10
0x13b44:      mov      rdi, r9
0x13b47:      mov      qword ptr [rsp + 0x18], rcx
0x13b4c:      movsxd   rsi, r11d
0x13b4f:      mov      dword ptr [rsp + 0x14], r8d
0x13b54:      shl      rsi, 3
0x13b58:      mov      dword ptr [rsp + 8], r11d
0x13b5d:      call     0x1d690
0x13b62:      mov      rcx, qword ptr [rsp + 0x18]
0x13b67:      mov      r8d, dword ptr [rsp + 0x14]
0x13b6c:      mov      edx, 0x400
0x13b71:      mov      r11d, dword ptr [rsp + 8]
0x13b76:      mov      r9, rax
0x13b79:      jmp      0x13b15
0x13b15:
0x13b7b:      sub      rsp, 8
0x13b7f:      lea      rcx, [rip + 0xe05a]
0x13b86:      mov      r9d, r13d
0x13b89:      xor      eax, eax
0x13b8b:      push     r15
0x13b8d:      mov      edx, 0xa0
0x13b92:      mov      esi, 1
0x13b97:      mov      rdi, rbp
0x13b9a:      lea      r8, [rip + 0xd7b4]
0x13ba1:      call     0x4f10
0x13ba6:      pop      rdx
0x13ba7:      pop      rcx
0x13ba8:      jmp      0x13ab0
0x13ab0:
0x13bad:      mov      eax, dword ptr [rsp + 0x10]

```

```

0x13bb1:    lea     rbp, [rsp + 0x60]
0x13bb6:    test    eax, eax
0x13bb8:    je      0x13c32
0x13bba:    mov     r9, r12
0x13bbd:    lea     r8, [rip + 0xd791]
0x13bc4:    mov     rdi, rbp
0x13bc7:    xor     eax, eax
0x13bc9:    lea     rcx, [rip + 0xe048]
0x13bd0:    mov     edx, 0xa0
0x13bd5:    mov     esi, 1
0x13bda:    call    0x4f10
0x13bdf:    jmp     0x13ab0
0x13ab0:
0x13be4:    mov     rdi, qword ptr [rsi + 0x18]
0x13be8:    lea     rdx, [rsp + 0x20]
0x13bed:    xor     esi, esi
0x13bef:    mov     qword ptr [rsp + 0x28], -1
0x13bf8:    call    0x1c490
0x13bfd:    test    eax, eax
0x13bff:    js      0x137f7
0x13c05:    cmp     qword ptr [rsp + 0x28], -1
0x13c0b:    je      0x137f7
0x13c11:    call    0x4a60
0x13c16:    mov     r12, qword ptr [rsp + 0x30]
0x13c1b:    test    eax, eax
0x13c1d:    cmovne  r12, qword ptr [rsp + 0x38]
0x13c23:    imul    r12, qword ptr [rsp + 0x20]
0x13c29:    shr     r12, 0xa
0x13c2d:    jmp     0x137f7
0x137f7:
0x13c32:    mov     edx, 0xa0
0x13c37:    mov     esi, 1
0x13c3c:    mov     rdi, rbp
0x13c3f:    xor     eax, eax
0x13c41:    lea     r8, [rip + 0xd70d]
0x13c48:    lea     rcx, [rip + 0xd729]
0x13c4f:    call    0x4f10
0x13c54:    jmp     0x13ab0
0x13ab0:
0x13c59:    mov     rdi, qword ptr [rsp]
0x13c5d:    lea     rcx, [rip - 0x5724]
0x13c64:    xor     esi, esi
0x13c66:    xor     r13d, r13d
0x13c69:    mov     edx, 8
0x13c6e:    xor     r15d, r15d
0x13c71:    call    0x48e0
0x13c76:    mov     dword ptr [rsp + 0x10], 0
0x13c7e:    xor     r9d, r9d

```

```

0x13c81:      jmp      0x139ea
0x139ea
0x13c86:      xor      r15d, r15d
0x13c89:      xor      r9d, r9d
0x13c8c:      jmp      0x139ea
0x139ea
0x13c91:      call     0x49d0
0x13c96:      nop      word ptr cs:[rax + rax]
0x13ca0:      endbr64
0x13ca4:      push     r15
0x13ca6:      xor      r15d, r15d
0x13ca9:      push     r14
0x13cab:      push     r13
0x13cad:      push     r12
0x13caf:      xor      r12d, r12d
0x13cb2:      push     rbp
0x13cb3:      mov      rbp, rdi
0x13cb6:      push     rbx
0x13cb7:      sub      rsp, 0x18
0x13cbb:      call     0x49b0
0x13cc0:      lea      rdi, [rax + 1]
0x13cc4:      call     0x1d670
0x13cc9:      mov      esi, 8
0x13cce:      mov      edi, 2
0x13cd3:      mov      r13, rax
0x13cd6:      call     0x1d740
0x13cdb:      mov      r14, rax
0x13cde:      xor      eax, eax
0x13ce0:      movzx    ebx, byte ptr [rbp]
0x13ce4:      add      rbp, 1
0x13ce8:      cmp      bl, 0x5c
0x13ceb:      je       0x13d25
0x13ced:      jg       0x13cf9
0x13cef:      cmp      bl, 9
0x13cf2:      je       0x13d40
0x13cf4:      test     bl, 0xdf
0x13cf7:      je       0x13d40
0x13cf9:      test     r12d, r12d
0x13cfc:      je       0x13d0a
0x13cfe:      movsxd   rcx, eax
0x13d01:      add      eax, 1
0x13d04:      mov      byte ptr [r13 + rcx], 0x5c
0x13d0a:      movsxd   rcx, eax
0x13d0d:      add      rbp, 1
0x13d11:      add      eax, 1
0x13d14:      xor      r12d, r12d
0x13d17:      mov      byte ptr [r13 + rcx], bl
0x13d1c:      movzx    ebx, byte ptr [rbp - 1]

```

```

0x13d20:      cmp     bl, 0x5c
0x13d23:      jne     0x13ced
0x13d25:      test    r12d, r12d
0x13d28:      je      0x13d36
0x13d2a:      movsxd   rcx, eax
0x13d2d:      add     eax, 1
0x13d30:      mov     byte ptr [r13 + rcx], 0x5c
0x13d36:      xor     r12d, 1
0x13d3a:      jmp     0x13ce0
0x13ce0
0x13d3c:      nop     dword ptr [rax]
0x13d40:      test    bl, bl
0x13d42:      je      0x13d60
0x13d44:      test    r12b, 1
0x13d48:      je      0x13d60
0x13d4a:      movsxd   rcx, eax
0x13d4d:      mov     r12d, 1
0x13d53:      add     eax, 1
0x13d56:      mov     byte ptr [r13 + rcx], bl
0x13d5b:      jmp     0x13ce0
0x13ce0
0x13d5d:      nop     dword ptr [rax]
0x13d60:      test    eax, eax
0x13d62:      je      0x13da6
0x13d64:      cdqe
0x13d66:      movsxd   rdx, r15d
0x13d69:      mov     rdi, r13
0x13d6c:      mov     byte ptr [r13 + rax], 0
0x13d72:      lea     eax, [r15 + 1]
0x13d76:      lea     r15, [rdx*8]
0x13d7e:      lea     rcx, [r14 + r15]
0x13d82:      mov     dword ptr [rsp + 4], eax
0x13d86:      mov     qword ptr [rsp + 8], rcx
0x13d8b:      call    0x1d7a0
0x13d90:      mov     rcx, qword ptr [rsp + 8]
0x13d95:      mov     qword ptr [rcx], rax
0x13d98:      mov     qword ptr [r14 + r15 + 8], 0
0x13da1:      mov     r15d, dword ptr [rsp + 4]
0x13da6:      test    bl, bl
0x13da8:      je      0x13dd0
0x13daa:      lea     esi, [r15 + 2]
0x13dae:      mov     rdi, r14
0x13db1:      movsxd   rsi, esi
0x13db4:      shl     rsi, 3
0x13db8:      call    0x1d690
0x13dbd:      mov     r14, rax
0x13dc0:      xor     eax, eax
0x13dc2:      jmp     0x13ce0

```

```

0x13ce0
0x13dc7:      nop      word ptr [rax + rax]
0x13dd0:      mov      rdi, r13
0x13dd3:      call     0x18d10
0x13dd8:      add      rsp, 0x18
0x13ddc:      mov      rax, r14
0x13ddf:      pop      rbx
0x13de0:      pop      rbp
0x13de1:      pop      r12
0x13de3:      pop      r13
0x13de5:      pop      r14
0x13de7:      pop      r15
0x13de9:      ret
0x13dea:      nop      word ptr [rax + rax]
0x13df0:      endbr64
0x13df4:      push     rbp
0x13df5:      mov      rbp, rdi
0x13df8:      push     rbx
0x13df9:      sub      rsp, 8
0x13dfd:      mov      rdi, qword ptr [rdi]
0x13e00:      test     rdi, rdi
0x13e03:      je       0x13e22
0x13e05:      mov      rbx, rbp
0x13e08:      nop      dword ptr [rax + rax]
0x13e10:      call     0x18d10
0x13e15:      mov      rdi, qword ptr [rbx + 8]
0x13e19:      add      rbx, 8
0x13e1d:      test     rdi, rdi
0x13e20:      jne      0x13e10
0x13e22:      add      rsp, 8
0x13e26:      mov      rdi, rbp
0x13e29:      pop      rbx
0x13e2a:      pop      rbp
0x13e2b:      jmp      0x18d10
0x18d10
0x13e30:      endbr64
0x13e34:      push     r15
0x13e36:      push     r14
0x13e38:      push     r13
0x13e3a:      push     r12
0x13e3c:      push     rbp
0x13e3d:      push     rbx
0x13e3e:      sub      rsp, 0x48
0x13e42:      mov      rax, qword ptr fs:[0x28]
0x13e4b:      mov      qword ptr [rsp + 0x38], rax
0x13e50:      xor      eax, eax
0x13e52:      cmp      esi, 0x33
0x13e55:      ja       0x14857

```

```

0x13e5b:      mov     r12, rdx
0x13e5e:      mov     eax, esi
0x13e60:      lea     rdx, [rip + 0xda9d]
0x13e67:      mov     rbp, rdi
0x13e6a:      movsxd  rax, dword ptr [rdx + rax*4]
0x13e6e:      mov     r15d, esi
0x13e71:      mov     r13, rcx
0x13e74:      add     rax, rdx
0x13e77:      jmp     rax
rax
->->->                                indirect JMP
0x6f96: mov     rdi, qword ptr [rbx + rax*8]
0x6f9a: lea     edx, [rax + 1]
0x6f9d: mov     dword ptr [rip + 0x238a5], edx
0x6fa3: call    0x1d7a0
0x6fa8: mov     r13, rax
0x6fab: movsxd  rax, dword ptr [rip + 0x23896]
0x6fb2: cmp     eax, ebp
0x6fb4: jge     0x5124
0x6fba: mov     rdi, qword ptr [rbx + rax*8]
0x6fbe: lea     edx, [rax + 1]
0x6fc1: mov     dword ptr [rip + 0x23881], edx
0x6fc7: call    0x1d7a0
0x6fcc: cmp     dword ptr [rip + 0x23876], ebp
0x6fd2: mov     r14, rax
0x6fd5: jge     0x5127
0x6fdb: mov     rcx, qword ptr [rip + 0x23906]
0x6fe2: mov     rdi, qword ptr [rip + 0x23897]
0x6fe9: lea     rdx, [rip + 0x198f8]
0x6ff0: xor     eax, eax
0x6ff2: mov     esi, 1
0x6ff7: call    0x4e60
0x6ffc: jmp     0x5127
0x5127
0x7001: mov     rdi, r13
0x7004: call    0x1d7a0
0x7009: mov     r12, rax
0x700c: jmp     0x5532
0x5532
0x7011: lea     rbx, [rip + 0x19543]
0x7018: jmp     0x529c
0x529c
0x701d: mov     rdi, qword ptr [rip + 0x238ac]
0x7024: xor     edx, edx
0x7026: lea     rsi, [rip + 0x19505]
0x702d: call    0x8070
0x7032: cmp     al, 0x79
0x7034: jne     0x5730

```

```

0x703a: jmp      0x6442
0x6442
0x703f: mov      edi, 1
0x7044: xor      eax, eax
0x7046: call     0x7bd0
0x704b: jmp      0x59ce
0x59ce
0x7050: mov      esi, 1
0x7055: lea      rdi, [rip + 0x19664]
0x705c: call     0x17e50
0x7061: jmp      0x53df
0x53df
0x7066: mov      rcx, qword ptr [rip + 0x2387b]
0x706d: mov      rdi, qword ptr [rip + 0x2380c]
0x7074: xor      eax, eax
0x7076: mov      r8d, 0x800
0x707c: lea      rdx, [rip + 0x19895]
0x7083: mov      esi, 1
0x7088: call     0x4e60
0x708d: jmp      0x54a3
0x54a3
0x7092: xor      esi, esi
0x7094: lea      rdi, [rip + 0x195fa]
0x709b: call     0x17e50
0x70a0: mov      r12d, eax
0x70a3: jmp      0x53a3
0x53a3
0x70a8: mov      r8, r12
0x70ab: lea      rcx, [rip + 0x19696]
0x70b2: jmp      0x5f42
0x5f42
0x70b7: mov      rdi, qword ptr [r13 + 0x10]
0x70bb: test     rdi, rdi
0x70be: je       0x6955
0x70c4: xor      eax, eax
0x70c6: call     0x19940
0x70cb: mov      r8d, 1
0x70d1: mov      esi, 0x21
0x70d6: mov      rdx, qword ptr [rip + 0x237cb]
0x70dd: mov      rdi, qword ptr [rip + 0x237cc]
0x70e4: mov      rcx, rax
0x70e7: mov      rbp, rax
0x70ea: call     0x13e30
0x70ef: mov      rdi, qword ptr [rip + 0x237ba]
0x70f6: call     0x11100
0x70fb: mov      rdi, rax
0x70fe: xor      eax, eax
0x7100: call     0x89d0

```



```

0x7105: mov     rdi, rbp
0x7108: xor     ebp, ebp
0x710a: call    0x18d10
0x710f: jmp     0x565a
0x565a
0x7114: mov     qword ptr [rsp + 0x40], 0
0x711d: lea     r12, [rsp + 0x40]
0x7122: lea     rbp, [rip + 0x1966f]
0x7129: jmp     0x714f
0x714f
0x712b: lea     rdi, [rip + 0x22ade]
0x7132: call    0xdb60
0x7137: xor     eax, eax
0x7139: call    0xa5a0
0x713e: xor     eax, eax
0x7140: call    0xa820
0x7145: mov     edi, 2
0x714a: call    0x4ea0
0x714f: mov     rsi, r12
0x7152: xor     ecx, ecx
0x7154: xor     edx, edx
0x7156: mov     rdi, rbp
0x7159: call    0x83e0
0x715e: xor     eax, eax
0x7160: call    0xa820
0x7165: mov     rsi, qword ptr [rsp + 0x40]
0x716a: test    rsi, rsi
0x716d: je      0x712b
0x716f: mov     rdi, qword ptr [rsp + 0x38]
0x7174: call    0x4b30
0x7179: test    eax, eax
0x717b: jne     0x712b
0x717d: xor     eax, eax
0x717f: call    0xa820
0x7184: mov     rdi, qword ptr [rsp + 0x38]
0x7189: call    0x18d10
0x718e: mov     rdi, qword ptr [rsp + 0x40]
0x7193: call    0x18d10
0x7198: jmp     0x5a59
0x5a59
0x719d: xor     eax, eax
0x719f: xor     ebp, ebp
0x71a1: call    0xa5a0
0x71a6: mov     rdi, qword ptr [r13]
0x71aa: call    0x195e0
0x71af: jmp     0x565a
0x565a
0x71b4: mov     rdi, qword ptr [rip + 0x236f5]

```

```

0x71bb: xor     ebp, ebp
0x71bd: call    0x110b0
0x71c2: jmp     0x565a
0x565a
0x71c7: cmp     byte ptr [rbp + 1], 0
0x71cb: mov     rdx, qword ptr [rip + 0x236d6]
0x71d2: mov     rdi, qword ptr [rip + 0x236d7]
0x71d9: jne     0x72a7
0x71df: mov     r8d, 1
0x71e5: xor     ecx, ecx
0x71e7: mov     esi, 5
0x71ec: call    0x13e30
0x71f1: jmp     0x6f11
0x6f11
0x71f6: mov     rcx, qword ptr [rip + 0x236cb]
0x71fd: mov     rdx, qword ptr [rip + 0x236a4]
0x7204: mov     esi, 1
0x7209: mov     r8d, 1
0x720f: mov     rdi, qword ptr [rip + 0x2369a]
0x7216: call    0x13e30
0x721b: mov     rdi, qword ptr [rip + 0x2368e]
0x7222: mov     ebp, eax
0x7224: call    0x11100
0x7229: mov     rdi, rax
0x722c: call    0xc4b0
0x7231: xor     edi, edi
0x7233: call    0xd1a0
0x7238: xor     eax, eax
0x723a: call    0x9640
0x723f: xor     eax, eax
0x7241: call    0xc8c0
0x7246: mov     r14, rax
0x7249: test    ebp, ebp
0x724b: je      0x5730
0x7251: xor     ebp, ebp
0x7253: jmp     0x565a
0x565a
0x7258: mov     rdi, qword ptr [rsp + 0x48]
0x725d: mov     edx, 0x1c
0x7262: xor     esi, esi
0x7264: call    0x8070
0x7269: jmp     0x5723
0x5723
0x726e: lea     rcx, [rbp + 1]
0x7272: mov     r8d, 1
0x7278: mov     esi, 0x25
0x727d: call    0x13e30
0x7282: jmp     0x6f11

```

```

0x6f11
0x7287: mov     rdi, qword ptr [rsp + 0x48]
0x728c: mov     edx, 0x1c
0x7291: xor      esi, esi
0x7293: call    0x8070
0x7298: xor      eax, eax
0x729a: call    0xc8c0
0x729f: mov     r14, rax
0x72a2: jmp     0x6611
0x6611
0x72a7: lea     rcx, [rbp + 1]
0x72ab: mov     r8d, 1
0x72b1: mov     esi, 0x24
0x72b6: call    0x13e30
0x72bb: jmp     0x6f11
0x6f11
0x72c0: mov     rdi, qword ptr [rsp + 0x48]
0x72c5: call    0x185d0
0x72ca: mov     ebp, eax
0x72cc: test    eax, eax
0x72ce: je      0x73fa
0x72d4: xor     edi, edi
0x72d6: call    0xd1a0
0x72db: mov     rdi, qword ptr [rsp + 0x48]
0x72e0: call    0xd610
0x72e5: mov     rdi, qword ptr [rsp + 0x48]
0x72ea: mov     esi, ebp
0x72ec: call    0x193f0
0x72f1: xor     edi, edi
0x72f3: call    0xdbb0
0x72f8: jmp     0x6f11
0x6f11
0x72fd: mov     rdx, qword ptr [rip + 0x235a4]
0x7304: mov     rdi, qword ptr [rip + 0x235a5]
0x730b: xor     ecx, ecx
0x730d: mov     esi, 7
0x7312: mov     r8d, 1
0x7318: call    0x13e30
0x731d: xor     edi, edi
0x731f: call    0xd1a0
0x7324: jmp     0x5723
0x5723
0x7329: mov     rdi, qword ptr [rip + 0x23580]
0x7330: mov     r15d, dword ptr [rsp + 0x10]
0x7335: call    0xf480
0x733a: mov     rdi, qword ptr [rip + 0x23567]
0x7341: call    0xf480
0x7346: xor     eax, eax

```

```

0x7348: call    0xa480
0x734d: xor     eax, eax
0x734f: call    0xe4e0
0x7354: xor     eax, eax
0x7356: call    0xa820
0x735b: xor     eax, eax
0x735d: call    0x18330
0x7362: mov     r8, rax
0x7365: mov     eax, 0x3c
0x736a: sub     eax, dword ptr [r8]
0x736d: mov     edi, eax
0x736f: call    0x4aa0
0x7374: mov     dword ptr [rip + 0x23566], 0
0x737e: jmp     0x6447
0x6447
0x7383: mov     rdi, r14
0x7386: xor     ebp, ebp
0x7388: call    0xc7e0
0x738d: xor     eax, eax
0x738f: call    0xc8c0
0x7394: mov     rdi, qword ptr [rip + 0x23515]
0x739b: mov     r14, rax
0x739e: call    0xf480
0x73a3: mov     rdi, qword ptr [rip + 0x234fe]
0x73aa: call    0xf480
0x73af: xor     eax, eax
0x73b1: call    0xa480
0x73b6: xor     eax, eax
0x73b8: call    0xe4e0
0x73bd: xor     eax, eax
0x73bf: call    0xa820
0x73c4: xor     eax, eax
0x73c6: call    0x18330
0x73cb: mov     r8, rax
0x73ce: mov     eax, 0x3c
0x73d3: sub     eax, dword ptr [r8]
0x73d6: mov     edi, eax
0x73d8: call    0x4aa0
0x73dd: mov     dword ptr [rip + 0x234fd], 0
0x73e7: jmp     0x565a
0x565a
0x73ec: lea     rdi, [rip + 0x1955d]
0x73f3: xor     eax, eax
0x73f5: call    0x8b00
0x73fa: mov     rdi, qword ptr [rsp + 0x48]
0x73ff: xor     ebp, ebp
0x7401: call    0xc4b0
0x7406: xor     edi, edi

```

```

0x7408: call    0xd1a0
0x740d: mov     rdi, qword ptr [rsp + 0x48]
0x7412: call    0xd610
0x7417: mov     rdi, qword ptr [rsp + 0x48]
0x741c: xor     esi, esi
0x741e: call    0x193f0
0x7423: xor     edi, edi
0x7425: call    0xdbb0
0x742a: xor     edi, edi
0x742c: call    0xd1a0
0x7431: mov     rdi, qword ptr [rip + 0x23478]
0x7438: call    0xf480
0x743d: mov     rdi, qword ptr [rip + 0x23464]
0x7444: call    0xf480
0x7449: xor     eax, eax
0x744b: call    0xa480
0x7450: mov     rdi, qword ptr [rip + 0x23471]
0x7457: call    0xbb10
0x745c: xor     eax, eax
0x745e: mov     dword ptr [rip + 0x23474], 1
0x7468: call    0xc8c0
0x746d: mov     r14, rax
0x7470: jmp     0x565a
0x565a
0x7475: call    0x49d0
0x747a: nop     word ptr [rax + rax]
0x7480: endbr64
0x7484: xor     ebp, ebp
0x7486: mov     r9, rdx
0x7489: pop     rsi
0x748a: mov     rdx, rsp
0x748d: and     rsp, 0xffffffffffffffff
0x7491: push    rax
0x7492: push    rsp
0x7493: xor     r8d, r8d
0x7496: xor     ecx, ecx
0x7498: lea     rdi, [rip - 0x256f]
0x749f: call    qword ptr [rip + 0x21b23]
0x74a5: hlt
0x74a6: nop     word ptr cs:[rax + rax]
0x74b0: lea     rdi, [rip + 0x23379]
0x74b7: lea     rax, [rip + 0x23372]
0x74be: cmp     rax, rdi
0x74c1: je      0x74d8
0x74c3: mov     rax, qword ptr [rip + 0x21b06]
0x74ca: test    rax, rax
0x74cd: je      0x74d8
0x74cf: jmp     rax

```

rax	->->->	indirect JMP
0x6f80:	xor esi, esi	
0x6f82:	lea rdi, [rip + 0x19717]	
0x6f89:	call 0x17e50	
0x6f8e:	mov dword ptr [rsp], eax	
0x6f91:	jmp 0x53b4	
0x53b4		
0x6f96:	mov rdi, qword ptr [rbx + rax*8]	
0x6f9a:	lea edx, [rax + 1]	
0x6f9d:	mov dword ptr [rip + 0x238a5], edx	
0x6fa3:	call 0x1d7a0	
0x6fa8:	mov r13, rax	
0x6fab:	movsxd rax, dword ptr [rip + 0x23896]	
0x6fb2:	cmp eax, ebp	
0x6fb4:	jge 0x5124	
0x6fba:	mov rdi, qword ptr [rbx + rax*8]	
0x6fbe:	lea edx, [rax + 1]	
0x6fc1:	mov dword ptr [rip + 0x23881], edx	
0x6fc7:	call 0x1d7a0	
0x6fcc:	cmp dword ptr [rip + 0x23876], ebp	
0x6fd2:	mov r14, rax	
0x6fd5:	jge 0x5127	
0x6fdb:	mov rcx, qword ptr [rip + 0x23906]	
0x6fe2:	mov rdi, qword ptr [rip + 0x23897]	
0x6fe9:	lea rdx, [rip + 0x198f8]	
0x6ff0:	xor eax, eax	
0x6ff2:	mov esi, 1	
0x6ff7:	call 0x4e60	
0x6ffc:	jmp 0x5127	
0x5127		
0x7001:	mov rdi, r13	
0x7004:	call 0x1d7a0	
0x7009:	mov r12, rax	
0x700c:	jmp 0x5532	
0x5532		
0x7011:	lea rbx, [rip + 0x19543]	
0x7018:	jmp 0x529c	
0x529c		
0x701d:	mov rdi, qword ptr [rip + 0x238ac]	
0x7024:	xor edx, edx	
0x7026:	lea rsi, [rip + 0x19505]	
0x702d:	call 0x8070	
0x7032:	cmp al, 0x79	
0x7034:	jne 0x5730	
0x703a:	jmp 0x6442	
0x6442		
0x703f:	mov edi, 1	

```

0x7044: xor     eax, eax
0x7046: call    0x7bd0
0x704b: jmp     0x59ce
0x59ce
0x7050: mov     esi, 1
0x7055: lea     rdi, [rip + 0x19664]
0x705c: call    0x17e50
0x7061: jmp     0x53df
0x53df
0x7066: mov     rcx, qword ptr [rip + 0x2387b]
0x706d: mov     rdi, qword ptr [rip + 0x2380c]
0x7074: xor     eax, eax
0x7076: mov     r8d, 0x800
0x707c: lea     rdx, [rip + 0x19895]
0x7083: mov     esi, 1
0x7088: call    0x4e60
0x708d: jmp     0x54a3
0x54a3
0x7092: xor     esi, esi
0x7094: lea     rdi, [rip + 0x195fa]
0x709b: call    0x17e50
0x70a0: mov     r12d, eax
0x70a3: jmp     0x53a3
0x53a3
0x70a8: mov     r8, r12
0x70ab: lea     rcx, [rip + 0x19696]
0x70b2: jmp     0x5f42
0x5f42
0x70b7: mov     rdi, qword ptr [r13 + 0x10]
0x70bb: test    rdi, rdi
0x70be: je      0x6955
0x70c4: xor     eax, eax
0x70c6: call    0x19940
0x70cb: mov     r8d, 1
0x70d1: mov     esi, 0x21
0x70d6: mov     rdx, qword ptr [rip + 0x237cb]
0x70dd: mov     rdi, qword ptr [rip + 0x237cc]
0x70e4: mov     rcx, rax
0x70e7: mov     rbp, rax
0x70ea: call    0x13e30
0x70ef: mov     rdi, qword ptr [rip + 0x237ba]
0x70f6: call    0x11100
0x70fb: mov     rdi, rax
0x70fe: xor     eax, eax
0x7100: call    0x89d0
0x7105: mov     rdi, rbp
0x7108: xor     ebp, ebp
0x710a: call    0x18d10

```

```

0x710f: jmp      0x565a
0x565a
0x7114: mov      qword ptr [rsp + 0x40], 0
0x711d: lea      r12, [rsp + 0x40]
0x7122: lea      rbp, [rip + 0x1966f]
0x7129: jmp      0x714f
0x714f
0x712b: lea      rdi, [rip + 0x22ade]
0x7132: call     0xdb60
0x7137: xor      eax, eax
0x7139: call     0xa5a0
0x713e: xor      eax, eax
0x7140: call     0xa820
0x7145: mov      edi, 2
0x714a: call     0x4ea0
0x714f: mov      rsi, r12
0x7152: xor      ecx, ecx
0x7154: xor      edx, edx
0x7156: mov      rdi, rbp
0x7159: call     0x83e0
0x715e: xor      eax, eax
0x7160: call     0xa820
0x7165: mov      rsi, qword ptr [rsp + 0x40]
0x716a: test     rsi, rsi
0x716d: je       0x712b
0x716f: mov      rdi, qword ptr [rsp + 0x38]
0x7174: call     0x4b30
0x7179: test     eax, eax
0x717b: jne      0x712b
0x717d: xor      eax, eax
0x717f: call     0xa820
0x7184: mov      rdi, qword ptr [rsp + 0x38]
0x7189: call     0x18d10
0x718e: mov      rdi, qword ptr [rsp + 0x40]
0x7193: call     0x18d10
0x7198: jmp      0x5a59
0x5a59
0x719d: xor      eax, eax
0x719f: xor      ebp, ebp
0x71a1: call     0xa5a0
0x71a6: mov      rdi, qword ptr [r13]
0x71aa: call     0x195e0
0x71af: jmp      0x565a
0x565a
0x71b4: mov      rdi, qword ptr [rip + 0x236f5]
0x71bb: xor      ebp, ebp
0x71bd: call     0x110b0
0x71c2: jmp      0x565a

```



```

0x565a
0x71c7: cmp     byte ptr [rbp + 1], 0
0x71cb: mov     rdx, qword ptr [rip + 0x236d6]
0x71d2: mov     rdi, qword ptr [rip + 0x236d7]
0x71d9: jne     0x72a7
0x71df: mov     r8d, 1
0x71e5: xor     ecx, ecx
0x71e7: mov     esi, 5
0x71ec: call    0x13e30
0x71f1: jmp     0x6f11
0x6f11
0x71f6: mov     rcx, qword ptr [rip + 0x236cb]
0x71fd: mov     rdx, qword ptr [rip + 0x236a4]
0x7204: mov     esi, 1
0x7209: mov     r8d, 1
0x720f: mov     rdi, qword ptr [rip + 0x2369a]
0x7216: call    0x13e30
0x721b: mov     rdi, qword ptr [rip + 0x2368e]
0x7222: mov     ebp, eax
0x7224: call    0x11100
0x7229: mov     rdi, rax
0x722c: call    0xc4b0
0x7231: xor     edi, edi
0x7233: call    0xd1a0
0x7238: xor     eax, eax
0x723a: call    0x9640
0x723f: xor     eax, eax
0x7241: call    0xc8c0
0x7246: mov     r14, rax
0x7249: test    ebp, ebp
0x724b: je      0x5730
0x7251: xor     ebp, ebp
0x7253: jmp     0x565a
0x565a
0x7258: mov     rdi, qword ptr [rsp + 0x48]
0x725d: mov     edx, 0x1c
0x7262: xor     esi, esi
0x7264: call    0x8070
0x7269: jmp     0x5723
0x5723
0x726e: lea     rcx, [rbp + 1]
0x7272: mov     r8d, 1
0x7278: mov     esi, 0x25
0x727d: call    0x13e30
0x7282: jmp     0x6f11
0x6f11
0x7287: mov     rdi, qword ptr [rsp + 0x48]
0x728c: mov     edx, 0x1c

```

```

0x7291: xor     esi, esi
0x7293: call    0x8070
0x7298: xor     eax, eax
0x729a: call    0xc8c0
0x729f: mov     r14, rax
0x72a2: jmp     0x6611
0x6611
0x72a7: lea     rcx, [rbp + 1]
0x72ab: mov     r8d, 1
0x72b1: mov     esi, 0x24
0x72b6: call    0x13e30
0x72bb: jmp     0x6f11
0x6f11
0x72c0: mov     rdi, qword ptr [rsp + 0x48]
0x72c5: call    0x185d0
0x72ca: mov     ebp, eax
0x72cc: test    eax, eax
0x72ce: je      0x73fa
0x72d4: xor     edi, edi
0x72d6: call    0xd1a0
0x72db: mov     rdi, qword ptr [rsp + 0x48]
0x72e0: call    0xd610
0x72e5: mov     rdi, qword ptr [rsp + 0x48]
0x72ea: mov     esi, ebp
0x72ec: call    0x193f0
0x72f1: xor     edi, edi
0x72f3: call    0xdbb0
0x72f8: jmp     0x6f11
0x6f11
0x72fd: mov     rdx, qword ptr [rip + 0x235a4]
0x7304: mov     rdi, qword ptr [rip + 0x235a5]
0x730b: xor     ecx, ecx
0x730d: mov     esi, 7
0x7312: mov     r8d, 1
0x7318: call    0x13e30
0x731d: xor     edi, edi
0x731f: call    0xd1a0
0x7324: jmp     0x5723
0x5723
0x7329: mov     rdi, qword ptr [rip + 0x23580]
0x7330: mov     r15d, dword ptr [rsp + 0x10]
0x7335: call    0xf480
0x733a: mov     rdi, qword ptr [rip + 0x23567]
0x7341: call    0xf480
0x7346: xor     eax, eax
0x7348: call    0xa480
0x734d: xor     eax, eax
0x734f: call    0xe4e0

```

```

0x7354: xor     eax, eax
0x7356: call    0xa820
0x735b: xor     eax, eax
0x735d: call    0x18330
0x7362: mov     r8, rax
0x7365: mov     eax, 0x3c
0x736a: sub     eax, dword ptr [r8]
0x736d: mov     edi, eax
0x736f: call    0x4aa0
0x7374: mov     dword ptr [rip + 0x23566], 0
0x737e: jmp     0x6447
0x6447
0x7383: mov     rdi, r14
0x7386: xor     ebp, ebp
0x7388: call    0xc7e0
0x738d: xor     eax, eax
0x738f: call    0xc8c0
0x7394: mov     rdi, qword ptr [rip + 0x23515]
0x739b: mov     r14, rax
0x739e: call    0xf480
0x73a3: mov     rdi, qword ptr [rip + 0x234fe]
0x73aa: call    0xf480
0x73af: xor     eax, eax
0x73b1: call    0xa480
0x73b6: xor     eax, eax
0x73b8: call    0xe4e0
0x73bd: xor     eax, eax
0x73bf: call    0xa820
0x73c4: xor     eax, eax
0x73c6: call    0x18330
0x73cb: mov     r8, rax
0x73ce: mov     eax, 0x3c
0x73d3: sub     eax, dword ptr [r8]
0x73d6: mov     edi, eax
0x73d8: call    0x4aa0
0x73dd: mov     dword ptr [rip + 0x234fd], 0
0x73e7: jmp     0x565a
0x565a
0x73ec: lea     rdi, [rip + 0x1955d]
0x73f3: xor     eax, eax
0x73f5: call    0x8b00
0x73fa: mov     rdi, qword ptr [rsp + 0x48]
0x73ff: xor     ebp, ebp
0x7401: call    0xc4b0
0x7406: xor     edi, edi
0x7408: call    0xd1a0
0x740d: mov     rdi, qword ptr [rsp + 0x48]
0x7412: call    0xd610

```

```

0x7417: mov     rdi, qword ptr [rsp + 0x48]
0x741c: xor     esi, esi
0x741e: call    0x193f0
0x7423: xor     edi, edi
0x7425: call    0xd1a0
0x742a: xor     edi, edi
0x742c: call    0xd1a0
0x7431: mov     rdi, qword ptr [rip + 0x23478]
0x7438: call    0xf480
0x743d: mov     rdi, qword ptr [rip + 0x23464]
0x7444: call    0xf480
0x7449: xor     eax, eax
0x744b: call    0xa480
0x7450: mov     rdi, qword ptr [rip + 0x23471]
0x7457: call    0xb10
0x745c: xor     eax, eax
0x745e: mov     dword ptr [rip + 0x23474], 1
0x7468: call    0xc8c0
0x746d: mov     r14, rax
0x7470: jmp     0x565a
0x565a
0x7475: call    0x49d0
0x747a: nop     word ptr [rax + rax]
0x7480: endbr64
0x7484: xor     ebp, ebp
0x7486: mov     r9, rdx
0x7489: pop     rsi
0x748a: mov     rdx, rsp
0x748d: and     rsp, 0xfffffffffffffff0
0x7491: push    rax
0x7492: push    rsp
0x7493: xor     r8d, r8d
0x7496: xor     ecx, ecx
0x7498: lea     rdi, [rip - 0x256f]
0x749f: call    qword ptr [rip + 0x21b23]
0x74a5: hlt
0x74a6: nop     word ptr cs:[rax + rax]
0x74b0: lea     rdi, [rip + 0x23379]
0x74b7: lea     rax, [rip + 0x23372]
0x74be: cmp     rax, rdi
0x74c1: je      0x74d8
0x74c3: mov     rax, qword ptr [rip + 0x21b06]
0x74ca: test    rax, rax
0x74cd: je      0x74d8
0x74cf: jmp     rax
rax
->->->                                indirect JMP
0x6f70: mov     rdi, r14

```

```

0x6f73: call    0x1d7a0
0x6f78: mov     r12, rax
0x6f7b: jmp     0x5593
0x5593
0x6f80: xor     esi, esi
0x6f82: lea     rdi, [rip + 0x19717]
0x6f89: call    0x17e50
0x6f8e: mov     dword ptr [rsp], eax
0x6f91: jmp     0x53b4
0x53b4
0x6f96: mov     rdi, qword ptr [rbx + rax*8]
0x6f9a: lea     edx, [rax + 1]
0x6f9d: mov     dword ptr [rip + 0x238a5], edx
0x6fa3: call    0x1d7a0
0x6fa8: mov     r13, rax
0x6fab: movsxd  rax, dword ptr [rip + 0x23896]
0x6fb2: cmp     eax, ebp
0x6fb4: jge     0x5124
0x6fba: mov     rdi, qword ptr [rbx + rax*8]
0x6fbe: lea     edx, [rax + 1]
0x6fc1: mov     dword ptr [rip + 0x23881], edx
0x6fc7: call    0x1d7a0
0x6fcc: cmp     dword ptr [rip + 0x23876], ebp
0x6fd2: mov     r14, rax
0x6fd5: jge     0x5127
0x6fdb: mov     rcx, qword ptr [rip + 0x23906]
0x6fe2: mov     rdi, qword ptr [rip + 0x23897]
0x6fe9: lea     rdx, [rip + 0x198f8]
0x6ff0: xor     eax, eax
0x6ff2: mov     esi, 1
0x6ff7: call    0x4e60
0x6ffc: jmp     0x5127
0x5127
0x7001: mov     rdi, r13
0x7004: call    0x1d7a0
0x7009: mov     r12, rax
0x700c: jmp     0x5532
0x5532
0x7011: lea     rbx, [rip + 0x19543]
0x7018: jmp     0x529c
0x529c
0x701d: mov     rdi, qword ptr [rip + 0x238ac]
0x7024: xor     edx, edx
0x7026: lea     rsi, [rip + 0x19505]
0x702d: call    0x8070
0x7032: cmp     al, 0x79
0x7034: jne     0x5730
0x703a: jmp     0x6442

```

```

0x6442
0x703f: mov     edi, 1
0x7044: xor      eax, eax
0x7046: call     0x7bd0
0x704b: jmp      0x59ce
0x59ce
0x7050: mov     esi, 1
0x7055: lea     rdi, [rip + 0x19664]
0x705c: call    0x17e50
0x7061: jmp     0x53df
0x53df
0x7066: mov     rcx, qword ptr [rip + 0x2387b]
0x706d: mov     rdi, qword ptr [rip + 0x2380c]
0x7074: xor     eax, eax
0x7076: mov     r8d, 0x800
0x707c: lea     rdx, [rip + 0x19895]
0x7083: mov     esi, 1
0x7088: call    0x4e60
0x708d: jmp     0x54a3
0x54a3
0x7092: xor     esi, esi
0x7094: lea     rdi, [rip + 0x195fa]
0x709b: call    0x17e50
0x70a0: mov     r12d, eax
0x70a3: jmp     0x53a3
0x53a3
0x70a8: mov     r8, r12
0x70ab: lea     rcx, [rip + 0x19696]
0x70b2: jmp     0x5f42
0x5f42
0x70b7: mov     rdi, qword ptr [r13 + 0x10]
0x70bb: test    rdi, rdi
0x70be: je      0x6955
0x70c4: xor     eax, eax
0x70c6: call    0x19940
0x70cb: mov     r8d, 1
0x70d1: mov     esi, 0x21
0x70d6: mov     rdx, qword ptr [rip + 0x237cb]
0x70dd: mov     rdi, qword ptr [rip + 0x237cc]
0x70e4: mov     rcx, rax
0x70e7: mov     rbp, rax
0x70ea: call    0x13e30
0x70ef: mov     rdi, qword ptr [rip + 0x237ba]
0x70f6: call    0x11100
0x70fb: mov     rdi, rax
0x70fe: xor     eax, eax
0x7100: call    0x89d0
0x7105: mov     rdi, rbp

```

```

0x7108: xor     ebp, ebp
0x710a: call    0x18d10
0x710f: jmp     0x565a
0x565a
0x7114: mov     qword ptr [rsp + 0x40], 0
0x711d: lea     r12, [rsp + 0x40]
0x7122: lea     rbp, [rip + 0x1966f]
0x7129: jmp     0x714f
0x714f
0x712b: lea     rdi, [rip + 0x22ade]
0x7132: call    0xdb60
0x7137: xor     eax, eax
0x7139: call    0xa5a0
0x713e: xor     eax, eax
0x7140: call    0xa820
0x7145: mov     edi, 2
0x714a: call    0x4ea0
0x714f: mov     rsi, r12
0x7152: xor     ecx, ecx
0x7154: xor     edx, edx
0x7156: mov     rdi, rbp
0x7159: call    0x83e0
0x715e: xor     eax, eax
0x7160: call    0xa820
0x7165: mov     rsi, qword ptr [rsp + 0x40]
0x716a: test    rsi, rsi
0x716d: je      0x712b
0x716f: mov     rdi, qword ptr [rsp + 0x38]
0x7174: call    0x4b30
0x7179: test    eax, eax
0x717b: jne     0x712b
0x717d: xor     eax, eax
0x717f: call    0xa820
0x7184: mov     rdi, qword ptr [rsp + 0x38]
0x7189: call    0x18d10
0x718e: mov     rdi, qword ptr [rsp + 0x40]
0x7193: call    0x18d10
0x7198: jmp     0x5a59
0x5a59
0x719d: xor     eax, eax
0x719f: xor     ebp, ebp
0x71a1: call    0xa5a0
0x71a6: mov     rdi, qword ptr [r13]
0x71aa: call    0x195e0
0x71af: jmp     0x565a
0x565a
0x71b4: mov     rdi, qword ptr [rip + 0x236f5]
0x71bb: xor     ebp, ebp

```

```

0x71bd: call    0x110b0
0x71c2: jmp      0x565a
0x565a
0x71c7: cmp      byte ptr [rbp + 1], 0
0x71cb: mov      rdx, qword ptr [rip + 0x236d6]
0x71d2: mov      rdi, qword ptr [rip + 0x236d7]
0x71d9: jne      0x72a7
0x71df: mov      r8d, 1
0x71e5: xor      ecx, ecx
0x71e7: mov      esi, 5
0x71ec: call    0x13e30
0x71f1: jmp      0x6f11
0x6f11
0x71f6: mov      rcx, qword ptr [rip + 0x236cb]
0x71fd: mov      rdx, qword ptr [rip + 0x236a4]
0x7204: mov      esi, 1
0x7209: mov      r8d, 1
0x720f: mov      rdi, qword ptr [rip + 0x2369a]
0x7216: call    0x13e30
0x721b: mov      rdi, qword ptr [rip + 0x2368e]
0x7222: mov      ebp, eax
0x7224: call    0x11100
0x7229: mov      rdi, rax
0x722c: call    0xc4b0
0x7231: xor      edi, edi
0x7233: call    0xd1a0
0x7238: xor      eax, eax
0x723a: call    0x9640
0x723f: xor      eax, eax
0x7241: call    0xc8c0
0x7246: mov      r14, rax
0x7249: test     ebp, ebp
0x724b: je       0x5730
0x7251: xor      ebp, ebp
0x7253: jmp      0x565a
0x565a
0x7258: mov      rdi, qword ptr [rsp + 0x48]
0x725d: mov      edx, 0x1c
0x7262: xor      esi, esi
0x7264: call    0x8070
0x7269: jmp      0x5723
0x5723
0x726e: lea      rcx, [rbp + 1]
0x7272: mov      r8d, 1
0x7278: mov      esi, 0x25
0x727d: call    0x13e30
0x7282: jmp      0x6f11
0x6f11

```



```

0x7287: mov     rdi, qword ptr [rsp + 0x48]
0x728c: mov     edx, 0x1c
0x7291: xor      esi, esi
0x7293: call    0x8070
0x7298: xor      eax, eax
0x729a: call    0xc8c0
0x729f: mov     r14, rax
0x72a2: jmp     0x6611
0x6611
0x72a7: lea     rcx, [rbp + 1]
0x72ab: mov     r8d, 1
0x72b1: mov     esi, 0x24
0x72b6: call    0x13e30
0x72bb: jmp     0x6f11
0x6f11
0x72c0: mov     rdi, qword ptr [rsp + 0x48]
0x72c5: call    0x185d0
0x72ca: mov     ebp, eax
0x72cc: test    eax, eax
0x72ce: je      0x73fa
0x72d4: xor     edi, edi
0x72d6: call    0xd1a0
0x72db: mov     rdi, qword ptr [rsp + 0x48]
0x72e0: call    0xd610
0x72e5: mov     rdi, qword ptr [rsp + 0x48]
0x72ea: mov     esi, ebp
0x72ec: call    0x193f0
0x72f1: xor     edi, edi
0x72f3: call    0xdbb0
0x72f8: jmp     0x6f11
0x6f11
0x72fd: mov     rdx, qword ptr [rip + 0x235a4]
0x7304: mov     rdi, qword ptr [rip + 0x235a5]
0x730b: xor     ecx, ecx
0x730d: mov     esi, 7
0x7312: mov     r8d, 1
0x7318: call    0x13e30
0x731d: xor     edi, edi
0x731f: call    0xd1a0
0x7324: jmp     0x5723
0x5723
0x7329: mov     rdi, qword ptr [rip + 0x23580]
0x7330: mov     r15d, dword ptr [rsp + 0x10]
0x7335: call    0xf480
0x733a: mov     rdi, qword ptr [rip + 0x23567]
0x7341: call    0xf480
0x7346: xor     eax, eax
0x7348: call    0xa480

```

```

0x734d: xor     eax, eax
0x734f: call    0xe4e0
0x7354: xor     eax, eax
0x7356: call    0xa820
0x735b: xor     eax, eax
0x735d: call    0x18330
0x7362: mov     r8, rax
0x7365: mov     eax, 0x3c
0x736a: sub     eax, dword ptr [r8]
0x736d: mov     edi, eax
0x736f: call    0x4aa0
0x7374: mov     dword ptr [rip + 0x23566], 0
0x737e: jmp     0x6447
0x6447:
0x7383: mov     rdi, r14
0x7386: xor     ebp, ebp
0x7388: call    0xc7e0
0x738d: xor     eax, eax
0x738f: call    0xc8c0
0x7394: mov     rdi, qword ptr [rip + 0x23515]
0x739b: mov     r14, rax
0x739e: call    0xf480
0x73a3: mov     rdi, qword ptr [rip + 0x234fe]
0x73aa: call    0xf480
0x73af: xor     eax, eax
0x73b1: call    0xa480
0x73b6: xor     eax, eax
0x73b8: call    0xe4e0
0x73bd: xor     eax, eax
0x73bf: call    0xa820
0x73c4: xor     eax, eax
0x73c6: call    0x18330
0x73cb: mov     r8, rax
0x73ce: mov     eax, 0x3c
0x73d3: sub     eax, dword ptr [r8]
0x73d6: mov     edi, eax
0x73d8: call    0x4aa0
0x73dd: mov     dword ptr [rip + 0x234fd], 0
0x73e7: jmp     0x565a
0x565a:
0x73ec: lea     rdi, [rip + 0x1955d]
0x73f3: xor     eax, eax
0x73f5: call    0x8b00
0x73fa: mov     rdi, qword ptr [rsp + 0x48]
0x73ff: xor     ebp, ebp
0x7401: call    0xc4b0
0x7406: xor     edi, edi
0x7408: call    0xd1a0

```

```

0x740d: mov     rdi, qword ptr [rsp + 0x48]
0x7412: call    0xd610
0x7417: mov     rdi, qword ptr [rsp + 0x48]
0x741c: xor     esi, esi
0x741e: call    0x193f0
0x7423: xor     edi, edi
0x7425: call    0xdbb0
0x742a: xor     edi, edi
0x742c: call    0xd1a0
0x7431: mov     rdi, qword ptr [rip + 0x23478]
0x7438: call    0xf480
0x743d: mov     rdi, qword ptr [rip + 0x23464]
0x7444: call    0xf480
0x7449: xor     eax, eax
0x744b: call    0xa480
0x7450: mov     rdi, qword ptr [rip + 0x23471]
0x7457: call    0xbb10
0x745c: xor     eax, eax
0x745e: mov     dword ptr [rip + 0x23474], 1
0x7468: call    0xc8c0
0x746d: mov     r14, rax
0x7470: jmp     0x565a
0x565a
0x7475: call    0x49d0
0x747a: nop     word ptr [rax + rax]
0x7480: endbr64
0x7484: xor     ebp, ebp
0x7486: mov     r9, rdx
0x7489: pop     rsi
0x748a: mov     rdx, rsp
0x748d: and     rsp, 0xfffffffffffffff0
0x7491: push    rax
0x7492: push    rsp
0x7493: xor     r8d, r8d
0x7496: xor     ecx, ecx
0x7498: lea     rdi, [rip - 0x256f]
0x749f: call    qword ptr [rip + 0x21b23]
0x74a5: hlt
0x74a6: nop     word ptr cs:[rax + rax]
0x74b0: lea     rdi, [rip + 0x23379]
0x74b7: lea     rax, [rip + 0x23372]
0x74be: cmp     rax, rdi
0x74c1: je      0x74d8
0x74c3: mov     rax, qword ptr [rip + 0x21b06]
0x74ca: test    rax, rax
0x74cd: je      0x74d8
0x74cf: jmp     rax
rax

```

```

->->->                                indirect JMP
0x6d10: xor      edi, edi
0x6d12: call     0x4aa0
0x6d17: mov      edx, 0x21
0x6d1c: mov      rcx, qword ptr [rip + 0x23b1d]
0x6d23: mov      esi, 1
0x6d28: lea      rdi, [rip + 0x19c49]
0x6d2f: call     0x4e40
0x6d34: mov      eax, dword ptr [rip + 0x23c62]
0x6d3a: xor      esi, esi
0x6d3c: lea      edi, [rax - 1]
0x6d3f: call     0xa4b0
0x6d44: xor      edi, edi
0x6d46: call     0xb180
0x6d4b: mov      dword ptr [rip + 0x23b87], 0
0x6d55: jmp      0x5667
0x5667
0x6d5a: mov      rdi, qword ptr [rip + 0x23b4f]
0x6d61: call     0x11100
0x6d66: mov      rdi, rax
0x6d69: xor      eax, eax
0x6d6b: call     0x89d0
0x6d70: jmp      0x571c
0x571c
0x6d75: xor      eax, eax
0x6d77: call     0xa6c0
0x6d7c: xor      eax, eax
0x6d7e: call     0xa300
0x6d83: jmp      0x56c4
0x56c4
0x6d88: mov      r13d, 1
0x6d8e: xor      eax, eax
0x6d90: call     0xa5a0
0x6d95: mov      dword ptr [rsp + 0xc], 1
0x6d9d: jmp      0x619a
0x619a
0x6da2: mov      r13d, 1
0x6da8: xor      eax, eax
0x6daa: call     0xa5a0
0x6daf: mov      dword ptr [rsp + 0xc], 1
0x6db7: jmp      0x6293
0x6293
0x6dbc: mov      rdi, rbp
0x6dbf: call     0xc4b0
0x6dc4: jmp      0x68aa
0x68aa
0x6dc9: mov      rdi, qword ptr [r13]
0x6dcd: call     0x49b0

```

```

0x6dd2: lea    rdi, [rax + 0x41]
0x6dd6: call   0x1d670
0x6ddb: mov     r9, qword ptr [r13]
0x6ddf: lea     r8, [rip + 0x19921]
0x6de6: lea     rcx, [rip + 0x19bb3]
0x6ded: mov     r12, rax
0x6df0: or      rdx, 0xffffffffffffffff
0x6df4: mov     esi, 1
0x6df9: mov     rdi, rax
0x6dfc: xor     eax, eax
0x6dfe: call   0x4f10
0x6e03: mov     rdi, r12
0x6e06: call   0xdb60
0x6e0b: xor     eax, eax
0x6e0d: call   0xa820
0x6e12: mov     rdi, r12
0x6e15: call   0x18d10
0x6e1a: jmp     0x689a
0x689a:
0x6e1f: cmp     r12d, -0x3d
0x6e23: jne     0x696f
0x6e29: mov     rax, qword ptr [rsp + 0x50]
0x6e2e: cmp     byte ptr [rax], 0
0x6e31: je      0x6611
0x6e37: mov     rdi, r14
0x6e3a: call   0xc7e0
0x6e3f: mov     rdi, qword ptr [rsp + 0x50]
0x6e44: xor     eax, eax
0x6e46: lea     rsi, [rsp + 0x48]
0x6e4b: call   0x1b5b0
0x6e50: test    eax, eax
0x6e52: js      0x7287
0x6e58: mov     rdi, qword ptr [rip + 0x23a69]
0x6e5f: call   0xbb40
0x6e64: xor     edi, edi
0x6e66: call   0xd1a0
0x6e6b: mov     rdi, qword ptr [rsp + 0x48]
0x6e70: call   0xd610
0x6e75: mov     rdi, qword ptr [rsp + 0x48]
0x6e7a: call   0xc4b0
0x6e7f: mov     rdi, qword ptr [rsp + 0x48]
0x6e84: xor     esi, esi
0x6e86: call   0x193f0
0x6e8b: mov     rdi, qword ptr [rip + 0x23a36]
0x6e92: call   0xbb10
0x6e97: xor     edi, edi
0x6e99: call   0xdbb0
0x6e9e: mov     rdi, qword ptr [rip + 0x22dc3]

```

```

0x6ea5: xor     edx, edx
0x6ea7: xor     esi, esi
0x6ea9: call    0xe480
0x6eae: xor     edi, edi
0x6eb0: call    0xd1a0
0x6eb5: xor     eax, eax
0x6eb7: call    0xc8c0
0x6ebc: mov     rdi, qword ptr [rip + 0x239ed]
0x6ec3: mov     r14, rax
0x6ec6: call    0x11100
0x6ecb: mov     rdi, rax
0x6ece: call    0xc4b0
0x6ed3: xor     eax, eax
0x6ed5: call    0xa820
0x6eda: jmp     0x6611
0x6611
0x6edf: cmp     al, 0x2d
0x6ee1: jne     0x60c3
0x6ee7: cmp     byte ptr [rbp + 1], 0
0x6eeb: mov     rdx, qword ptr [rip + 0x239b6]
0x6ef2: mov     rdi, qword ptr [rip + 0x239b7]
0x6ef9: jne     0x726e
0x6eff: mov     r8d, 1
0x6f05: xor     ecx, ecx
0x6f07: mov     esi, 6
0x6f0c: call    0x13e30
0x6f11: xor     edi, edi
0x6f13: call    0xd1a0
0x6f18: jmp     0x5723
0x5723
0x6f1d: mov     edx, 0xc
0x6f22: xor     esi, esi
0x6f24: lea     rdi, [rip + 0x197e9]
0x6f2b: call    0x8070
0x6f30: jmp     0x6cf2
0x6cf2
0x6f35: mov     rdi, qword ptr [rip + 0x2398c]
0x6f3c: call    0xbb10
0x6f41: jmp     0x6cda
0x6cda
0x6f46: mov     rdi, r12
0x6f49: call    0x49b0
0x6f4e: mov     rdi, r12
0x6f51: mov     rsi, rax
0x6f54: call    0x18530
0x6f59: test    eax, eax
0x6f5b: je      0x70a8
0x6f61: mov     r8, r12

```

```

0x6f64: lea      rcx, [rip + 0x197d6]
0x6f6b: jmp      0x5f42
0x5f42:
0x6f70: mov      rdi, r14
0x6f73: call     0x1d7a0
0x6f78: mov      r12, rax
0x6f7b: jmp      0x5593
0x5593:
0x6f80: xor      esi, esi
0x6f82: lea      rdi, [rip + 0x19717]
0x6f89: call     0x17e50
0x6f8e: mov      dword ptr [rsp], eax
0x6f91: jmp      0x53b4
0x53b4:
0x6f96: mov      rdi, qword ptr [rbx + rax*8]
0x6f9a: lea      edx, [rax + 1]
0x6f9d: mov      dword ptr [rip + 0x238a5], edx
0x6fa3: call     0x1d7a0
0x6fa8: mov      r13, rax
0x6fab: movsxd   rax, dword ptr [rip + 0x23896]
0x6fb2: cmp      eax, ebp
0x6fb4: jge      0x5124
0x6fba: mov      rdi, qword ptr [rbx + rax*8]
0x6fbe: lea      edx, [rax + 1]
0x6fc1: mov      dword ptr [rip + 0x23881], edx
0x6fc7: call     0x1d7a0
0x6fcc: cmp      dword ptr [rip + 0x23876], ebp
0x6fd2: mov      r14, rax
0x6fd5: jge      0x5127
0x6fdb: mov      rcx, qword ptr [rip + 0x23906]
0x6fe2: mov      rdi, qword ptr [rip + 0x23897]
0x6fe9: lea      rdx, [rip + 0x198f8]
0x6ff0: xor      eax, eax
0x6ff2: mov      esi, 1
0x6ff7: call     0x4e60
0x6ffc: jmp      0x5127
0x5127:
0x7001: mov      rdi, r13
0x7004: call     0x1d7a0
0x7009: mov      r12, rax
0x700c: jmp      0x5532
0x5532:
0x7011: lea      rbx, [rip + 0x19543]
0x7018: jmp      0x529c
0x529c:
0x701d: mov      rdi, qword ptr [rip + 0x238ac]
0x7024: xor      edx, edx
0x7026: lea      rsi, [rip + 0x19505]

```

```

0x702d: call    0x8070
0x7032: cmp     al, 0x79
0x7034: jne     0x5730
0x703a: jmp     0x6442
0x6442
0x703f: mov     edi, 1
0x7044: xor     eax, eax
0x7046: call    0x7bd0
0x704b: jmp     0x59ce
0x59ce
0x7050: mov     esi, 1
0x7055: lea     rdi, [rip + 0x19664]
0x705c: call    0x17e50
0x7061: jmp     0x53df
0x53df
0x7066: mov     rcx, qword ptr [rip + 0x2387b]
0x706d: mov     rdi, qword ptr [rip + 0x2380c]
0x7074: xor     eax, eax
0x7076: mov     r8d, 0x800
0x707c: lea     rdx, [rip + 0x19895]
0x7083: mov     esi, 1
0x7088: call    0x4e60
0x708d: jmp     0x54a3
0x54a3
0x7092: xor     esi, esi
0x7094: lea     rdi, [rip + 0x195fa]
0x709b: call    0x17e50
0x70a0: mov     r12d, eax
0x70a3: jmp     0x53a3
0x53a3
0x70a8: mov     r8, r12
0x70ab: lea     rcx, [rip + 0x19696]
0x70b2: jmp     0x5f42
0x5f42
0x70b7: mov     rdi, qword ptr [r13 + 0x10]
0x70bb: test    rdi, rdi
0x70be: je      0x6955
0x70c4: xor     eax, eax
0x70c6: call    0x19940
0x70cb: mov     r8d, 1
0x70d1: mov     esi, 0x21
0x70d6: mov     rdx, qword ptr [rip + 0x237cb]
0x70dd: mov     rdi, qword ptr [rip + 0x237cc]
0x70e4: mov     rcx, rax
0x70e7: mov     rbp, rax
0x70ea: call    0x13e30
0x70ef: mov     rdi, qword ptr [rip + 0x237ba]
0x70f6: call    0x11100

```



```

0x70fb: mov     rdi, rax
0x70fe: xor     eax, eax
0x7100: call    0x89d0
0x7105: mov     rdi, rbp
0x7108: xor     ebp, ebp
0x710a: call    0x18d10
0x710f: jmp     0x565a
0x565a
0x7114: mov     qword ptr [rsp + 0x40], 0
0x711d: lea     r12, [rsp + 0x40]
0x7122: lea     rbp, [rip + 0x1966f]
0x7129: jmp     0x714f
0x714f
0x712b: lea     rdi, [rip + 0x22ade]
0x7132: call    0xdb60
0x7137: xor     eax, eax
0x7139: call    0xa5a0
0x713e: xor     eax, eax
0x7140: call    0xa820
0x7145: mov     edi, 2
0x714a: call    0x4ea0
0x714f: mov     rsi, r12
0x7152: xor     ecx, ecx
0x7154: xor     edx, edx
0x7156: mov     rdi, rbp
0x7159: call    0x83e0
0x715e: xor     eax, eax
0x7160: call    0xa820
0x7165: mov     rsi, qword ptr [rsp + 0x40]
0x716a: test    rsi, rsi
0x716d: je      0x712b
0x716f: mov     rdi, qword ptr [rsp + 0x38]
0x7174: call    0x4b30
0x7179: test    eax, eax
0x717b: jne     0x712b
0x717d: xor     eax, eax
0x717f: call    0xa820
0x7184: mov     rdi, qword ptr [rsp + 0x38]
0x7189: call    0x18d10
0x718e: mov     rdi, qword ptr [rsp + 0x40]
0x7193: call    0x18d10
0x7198: jmp     0x5a59
0x5a59
0x719d: xor     eax, eax
0x719f: xor     ebp, ebp
0x71a1: call    0xa5a0
0x71a6: mov     rdi, qword ptr [r13]
0x71aa: call    0x195e0

```

```

0x71af: jmp      0x565a
0x565a
0x71b4: mov      rdi, qword ptr [rip + 0x236f5]
0x71bb: xor       ebp, ebp
0x71bd: call     0x110b0
0x71c2: jmp      0x565a
0x565a
0x71c7: cmp      byte ptr [rbp + 1], 0
0x71cb: mov      rdx, qword ptr [rip + 0x236d6]
0x71d2: mov      rdi, qword ptr [rip + 0x236d7]
0x71d9: jne      0x72a7
0x71df: mov      r8d, 1
0x71e5: xor      ecx, ecx
0x71e7: mov      esi, 5
0x71ec: call     0x13e30
0x71f1: jmp      0x6f11
0x6f11
0x71f6: mov      rcx, qword ptr [rip + 0x236cb]
0x71fd: mov      rdx, qword ptr [rip + 0x236a4]
0x7204: mov      esi, 1
0x7209: mov      r8d, 1
0x720f: mov      rdi, qword ptr [rip + 0x2369a]
0x7216: call     0x13e30
0x721b: mov      rdi, qword ptr [rip + 0x2368e]
0x7222: mov      ebp, eax
0x7224: call     0x11100
0x7229: mov      rdi, rax
0x722c: call     0xc4b0
0x7231: xor      edi, edi
0x7233: call     0xd1a0
0x7238: xor      eax, eax
0x723a: call     0x9640
0x723f: xor      eax, eax
0x7241: call     0xc8c0
0x7246: mov      r14, rax
0x7249: test     ebp, ebp
0x724b: je       0x5730
0x7251: xor      ebp, ebp
0x7253: jmp      0x565a
0x565a
0x7258: mov      rdi, qword ptr [rsp + 0x48]
0x725d: mov      edx, 0x1c
0x7262: xor      esi, esi
0x7264: call     0x8070
0x7269: jmp      0x5723
0x5723
0x726e: lea      rcx, [rbp + 1]
0x7272: mov      r8d, 1

```

```

0x7278: mov     esi, 0x25
0x727d: call    0x13e30
0x7282: jmp     0x6f11
0x6f11
0x7287: mov     rdi, qword ptr [rsp + 0x48]
0x728c: mov     edx, 0x1c
0x7291: xor     esi, esi
0x7293: call    0x8070
0x7298: xor     eax, eax
0x729a: call    0xc8c0
0x729f: mov     r14, rax
0x72a2: jmp     0x6611
0x6611
0x72a7: lea     rcx, [rbp + 1]
0x72ab: mov     r8d, 1
0x72b1: mov     esi, 0x24
0x72b6: call    0x13e30
0x72bb: jmp     0x6f11
0x6f11
0x72c0: mov     rdi, qword ptr [rsp + 0x48]
0x72c5: call    0x185d0
0x72ca: mov     ebp, eax
0x72cc: test    eax, eax
0x72ce: je      0x73fa
0x72d4: xor     edi, edi
0x72d6: call    0xd1a0
0x72db: mov     rdi, qword ptr [rsp + 0x48]
0x72e0: call    0xd610
0x72e5: mov     rdi, qword ptr [rsp + 0x48]
0x72ea: mov     esi, ebp
0x72ec: call    0x193f0
0x72f1: xor     edi, edi
0x72f3: call    0xdbb0
0x72f8: jmp     0x6f11
0x6f11
0x72fd: mov     rdx, qword ptr [rip + 0x235a4]
0x7304: mov     rdi, qword ptr [rip + 0x235a5]
0x730b: xor     ecx, ecx
0x730d: mov     esi, 7
0x7312: mov     r8d, 1
0x7318: call    0x13e30
0x731d: xor     edi, edi
0x731f: call    0xd1a0
0x7324: jmp     0x5723
0x5723
0x7329: mov     rdi, qword ptr [rip + 0x23580]
0x7330: mov     r15d, dword ptr [rsp + 0x10]
0x7335: call    0xf480

```

```

0x733a: mov     rdi, qword ptr [rip + 0x23567]
0x7341: call    0xf480
0x7346: xor     eax, eax
0x7348: call    0xa480
0x734d: xor     eax, eax
0x734f: call    0xe4e0
0x7354: xor     eax, eax
0x7356: call    0xa820
0x735b: xor     eax, eax
0x735d: call    0x18330
0x7362: mov     r8, rax
0x7365: mov     eax, 0x3c
0x736a: sub     eax, dword ptr [r8]
0x736d: mov     edi, eax
0x736f: call    0x4aa0
0x7374: mov     dword ptr [rip + 0x23566], 0
0x737e: jmp     0x6447
0x6447
0x7383: mov     rdi, r14
0x7386: xor     ebp, ebp
0x7388: call    0xc7e0
0x738d: xor     eax, eax
0x738f: call    0xc8c0
0x7394: mov     rdi, qword ptr [rip + 0x23515]
0x739b: mov     r14, rax
0x739e: call    0xf480
0x73a3: mov     rdi, qword ptr [rip + 0x234fe]
0x73aa: call    0xf480
0x73af: xor     eax, eax
0x73b1: call    0xa480
0x73b6: xor     eax, eax
0x73b8: call    0xe4e0
0x73bd: xor     eax, eax
0x73bf: call    0xa820
0x73c4: xor     eax, eax
0x73c6: call    0x18330
0x73cb: mov     r8, rax
0x73ce: mov     eax, 0x3c
0x73d3: sub     eax, dword ptr [r8]
0x73d6: mov     edi, eax
0x73d8: call    0x4aa0
0x73dd: mov     dword ptr [rip + 0x234fd], 0
0x73e7: jmp     0x565a
0x565a
0x73ec: lea     rdi, [rip + 0x1955d]
0x73f3: xor     eax, eax
0x73f5: call    0x8b00
0x73fa: mov     rdi, qword ptr [rsp + 0x48]

```

```

0x73ff: xor     ebp, ebp
0x7401: call    0xc4b0
0x7406: xor     edi, edi
0x7408: call    0xd1a0
0x740d: mov     rdi, qword ptr [rsp + 0x48]
0x7412: call    0xd610
0x7417: mov     rdi, qword ptr [rsp + 0x48]
0x741c: xor     esi, esi
0x741e: call    0x193f0
0x7423: xor     edi, edi
0x7425: call    0xdbb0
0x742a: xor     edi, edi
0x742c: call    0xd1a0
0x7431: mov     rdi, qword ptr [rip + 0x23478]
0x7438: call    0xf480
0x743d: mov     rdi, qword ptr [rip + 0x23464]
0x7444: call    0xf480
0x7449: xor     eax, eax
0x744b: call    0xa480
0x7450: mov     rdi, qword ptr [rip + 0x23471]
0x7457: call    0xbb10
0x745c: xor     eax, eax
0x745e: mov     dword ptr [rip + 0x23474], 1
0x7468: call    0xc8c0
0x746d: mov     r14, rax
0x7470: jmp     0x565a
0x565a:
0x7475: call    0x49d0
0x747a: nop     word ptr [rax + rax]
0x7480: endbr64
0x7484: xor     ebp, ebp
0x7486: mov     r9, rdx
0x7489: pop     rsi
0x748a: mov     rdx, rsp
0x748d: and     rsp, 0xffffffffffffffff
0x7491: push    rax
0x7492: push    rsp
0x7493: xor     r8d, r8d
0x7496: xor     ecx, ecx
0x7498: lea     rdi, [rip - 0x256f]
0x749f: call    qword ptr [rip + 0x21b23]
0x74a5: hlt
0x74a6: nop     word ptr cs:[rax + rax]
0x74b0: lea     rdi, [rip + 0x23379]
0x74b7: lea     rax, [rip + 0x23372]
0x74be: cmp     rax, rdi
0x74c1: je      0x74d8
0x74c3: mov     rax, qword ptr [rip + 0x21b06]

```

```

0x74ca: test    rax, rax
0x74cd: je      0x74d8
0x74cf: jmp     rax
rax
->->->                                indirect JMP

0x6ba2: mov     rax, qword ptr [rip + 0x23d17]
0x6ba9: mov     qword ptr [rip + 0x23d00], rax
0x6bb0: mov     rax, qword ptr [rip + 0x23d01]
0x6bb7: jmp     0x56b0
0x56b0
0x6bbc: cmp     dword ptr [rsp + 0xc], 0
0x6bc1: je      0x62d5
0x6bc7: xor     eax, eax
0x6bc9: call    0xa5a0
0x6bce: mov     rdi, qword ptr [rip + 0x23cdb]
0x6bd5: mov     esi, 1
0x6bda: call    0xf660
0x6bdf: mov     dword ptr [rsp + 0xc], 0
0x6be7: mov     rdi, qword ptr [rsp + 0x58]
0x6bec: jmp     0x62d5
0x62d5
0x6bf1: cmp     dword ptr [rsp + 0xc], 0
0x6bf6: je      0x61dc
0x6bfc: mov     rdi, qword ptr [rip + 0x23cad]
0x6c03: mov     esi, 1
0x6c08: call    0xf660
0x6c0d: mov     dword ptr [rsp + 0xc], 0
0x6c15: mov     rdi, qword ptr [rsp + 0x58]
0x6c1a: jmp     0x61dc
0x61dc
0x6c1f: mov     rdi, qword ptr [r12]
0x6c23: call    0x49b0
0x6c28: lea     rdi, [rax + 0x51]
0x6c2c: call    0x1d670
0x6c31: mov     r9, qword ptr [r13 + 0x20]
0x6c35: mov     r8, qword ptr [r13]
0x6c39: lea     rcx, [rip + 0x19d80]
0x6c40: mov     rbp, rax
0x6c43: or      rdx, 0xffffffffffffffff
0x6c47: mov     rdi, rax
0x6c4a: mov     esi, 1
0x6c4f: xor     eax, eax
0x6c51: call    0x4f10
0x6c56: mov     rdi, rbp
0x6c59: mov     edx, 0x1c
0x6c5e: xor     esi, esi
0x6c60: call    0x8070
0x6c65: mov     rdi, rbp

```

```

0x6c68: call    0x18d10
0x6c6d: jmp      0x5730
0x5730
0x6c72: mov      rax, qword ptr [rsp]
0x6c76: test     rax, rax
0x6c79: je       0x6c83
0x6c7b: mov      rdi, rax
0x6c7e: call     0x18d10
0x6c83: mov      rdi, qword ptr [rsp + 0x58]
0x6c88: test     rdi, rdi
0x6c8b: je       0x5730
0x6c91: call     0x4e70
0x6c96: mov      rdx, qword ptr [rip + 0x23c0b]
0x6c9d: xor      ecx, ecx
0x6c9f: mov      rdi, qword ptr [rip + 0x23c0a]
0x6ca6: mov      r8d, 1
0x6cac: mov      esi, 0x2a
0x6cb1: mov      qword ptr [rsp], rax
0x6cb5: call     0x13e30
0x6cba: xor      ecx, ecx
0x6cbc: mov      edx, 3
0x6cc1: xor      esi, esi
0x6cc3: xor      edi, edi
0x6cc5: call     0x86c0
0x6cca: jmp      0x5730
0x5730
0x6ccf: cmp      byte ptr [r13 + 0x18], 0
0x6cd4: jne      0x6f35
0x6cda: xor      eax, eax
0x6cdc: call     0xa480
0x6ce1: cmp      byte ptr [r13 + 0x19], 0
0x6ce6: je       0x6cf2
0x6ce8: mov      dword ptr [rip + 0x23bea], 1
0x6cf2: test     r12b, 0x7f
0x6cf6: jne      0x6955
0x6cfc: and      r12d, 0xff00
0x6d03: je       0x68f1
0x6d09: xor      ebp, ebp
0x6d0b: jmp      0x565a
0x565a
0x6d10: xor      edi, edi
0x6d12: call     0x4aa0
0x6d17: mov      edx, 0x21
0x6d1c: mov      rcx, qword ptr [rip + 0x23b1d]
0x6d23: mov      esi, 1
0x6d28: lea      rdi, [rip + 0x19c49]
0x6d2f: call     0x4e40
0x6d34: mov      eax, dword ptr [rip + 0x23c62]

```

```

0x6d3a: xor     esi, esi
0x6d3c: lea     edi, [rax - 1]
0x6d3f: call    0xa4b0
0x6d44: xor     edi, edi
0x6d46: call    0xb180
0x6d4b: mov     dword ptr [rip + 0x23b87], 0
0x6d55: jmp     0x5667
0x5667
0x6d5a: mov     rdi, qword ptr [rip + 0x23b4f]
0x6d61: call    0x11100
0x6d66: mov     rdi, rax
0x6d69: xor     eax, eax
0x6d6b: call    0x89d0
0x6d70: jmp     0x571c
0x571c
0x6d75: xor     eax, eax
0x6d77: call    0xa6c0
0x6d7c: xor     eax, eax
0x6d7e: call    0xa300
0x6d83: jmp     0x56c4
0x56c4
0x6d88: mov     r13d, 1
0x6d8e: xor     eax, eax
0x6d90: call    0xa5a0
0x6d95: mov     dword ptr [rsp + 0xc], 1
0x6d9d: jmp     0x619a
0x619a
0x6da2: mov     r13d, 1
0x6da8: xor     eax, eax
0x6daa: call    0xa5a0
0x6daf: mov     dword ptr [rsp + 0xc], 1
0x6db7: jmp     0x6293
0x6293
0x6dbc: mov     rdi, rbp
0x6dbf: call    0xc4b0
0x6dc4: jmp     0x68aa
0x68aa
0x6dc9: mov     rdi, qword ptr [r13]
0x6dcd: call    0x49b0
0x6dd2: lea     rdi, [rax + 0x41]
0x6dd6: call    0x1d670
0x6ddb: mov     r9, qword ptr [r13]
0x6ddf: lea     r8, [rip + 0x19921]
0x6de6: lea     rcx, [rip + 0x19bb3]
0x6ded: mov     r12, rax
0x6df0: or      rdx, 0xffffffffffffffff
0x6df4: mov     esi, 1
0x6df9: mov     rdi, rax

```



```

0x6dfc: xor     eax, eax
0x6dfe: call    0x4f10
0x6e03: mov     rdi, r12
0x6e06: call    0xdb60
0x6e0b: xor     eax, eax
0x6e0d: call    0xa820
0x6e12: mov     rdi, r12
0x6e15: call    0x18d10
0x6e1a: jmp     0x689a
0x689a:
0x6e1f: cmp     r12d, -0x3d
0x6e23: jne     0x696f
0x6e29: mov     rax, qword ptr [rsp + 0x50]
0x6e2e: cmp     byte ptr [rax], 0
0x6e31: je      0x6611
0x6e37: mov     rdi, r14
0x6e3a: call    0xc7e0
0x6e3f: mov     rdi, qword ptr [rsp + 0x50]
0x6e44: xor     eax, eax
0x6e46: lea     rsi, [rsp + 0x48]
0x6e4b: call    0x1b5b0
0x6e50: test    eax, eax
0x6e52: js      0x7287
0x6e58: mov     rdi, qword ptr [rip + 0x23a69]
0x6e5f: call    0xbb40
0x6e64: xor     edi, edi
0x6e66: call    0xd1a0
0x6e6b: mov     rdi, qword ptr [rsp + 0x48]
0x6e70: call    0xd610
0x6e75: mov     rdi, qword ptr [rsp + 0x48]
0x6e7a: call    0xc4b0
0x6e7f: mov     rdi, qword ptr [rsp + 0x48]
0x6e84: xor     esi, esi
0x6e86: call    0x193f0
0x6e8b: mov     rdi, qword ptr [rip + 0x23a36]
0x6e92: call    0xbb10
0x6e97: xor     edi, edi
0x6e99: call    0xdbb0
0x6e9e: mov     rdi, qword ptr [rip + 0x22dc3]
0x6ea5: xor     edx, edx
0x6ea7: xor     esi, esi
0x6ea9: call    0xe480
0x6eae: xor     edi, edi
0x6eb0: call    0xd1a0
0x6eb5: xor     eax, eax
0x6eb7: call    0xc8c0
0x6ebc: mov     rdi, qword ptr [rip + 0x239ed]
0x6ec3: mov     r14, rax

```

```

0x6ec6: call    0x11100
0x6ecb: mov     rdi, rax
0x6ece: call    0xc4b0
0x6ed3: xor     eax, eax
0x6ed5: call    0xa820
0x6eda: jmp     0x6611
0x6611
0x6edf: cmp     al, 0x2d
0x6ee1: jne     0x60c3
0x6ee7: cmp     byte ptr [rbp + 1], 0
0x6eeb: mov     rdx, qword ptr [rip + 0x239b6]
0x6ef2: mov     rdi, qword ptr [rip + 0x239b7]
0x6ef9: jne     0x726e
0x6eff: mov     r8d, 1
0x6f05: xor     ecx, ecx
0x6f07: mov     esi, 6
0x6f0c: call    0x13e30
0x6f11: xor     edi, edi
0x6f13: call    0xd1a0
0x6f18: jmp     0x5723
0x5723
0x6f1d: mov     edx, 0xc
0x6f22: xor     esi, esi
0x6f24: lea     rdi, [rip + 0x197e9]
0x6f2b: call    0x8070
0x6f30: jmp     0x6cf2
0x6cf2
0x6f35: mov     rdi, qword ptr [rip + 0x2398c]
0x6f3c: call    0xbb10
0x6f41: jmp     0x6cda
0x6cda
0x6f46: mov     rdi, r12
0x6f49: call    0x49b0
0x6f4e: mov     rdi, r12
0x6f51: mov     rsi, rax
0x6f54: call    0x18530
0x6f59: test    eax, eax
0x6f5b: je      0x70a8
0x6f61: mov     r8, r12
0x6f64: lea     rcx, [rip + 0x197d6]
0x6f6b: jmp     0x5f42
0x5f42
0x6f70: mov     rdi, r14
0x6f73: call    0x1d7a0
0x6f78: mov     r12, rax
0x6f7b: jmp     0x5593
0x5593
0x6f80: xor     esi, esi

```

```

0x6f82: lea      rdi, [rip + 0x19717]
0x6f89: call     0x17e50
0x6f8e: mov      dword ptr [rsp], eax
0x6f91: jmp      0x53b4
0x53b4
0x6f96: mov      rdi, qword ptr [rbx + rax*8]
0x6f9a: lea      edx, [rax + 1]
0x6f9d: mov      dword ptr [rip + 0x238a5], edx
0x6fa3: call     0x1d7a0
0x6fa8: mov      r13, rax
0x6fab: movsxd   rax, dword ptr [rip + 0x23896]
0x6fb2: cmp      eax, ebp
0x6fb4: jge      0x5124
0x6fba: mov      rdi, qword ptr [rbx + rax*8]
0x6fbe: lea      edx, [rax + 1]
0x6fc1: mov      dword ptr [rip + 0x23881], edx
0x6fc7: call     0x1d7a0
0x6fcc: cmp      dword ptr [rip + 0x23876], ebp
0x6fd2: mov      r14, rax
0x6fd5: jge      0x5127
0x6fdb: mov      rcx, qword ptr [rip + 0x23906]
0x6fe2: mov      rdi, qword ptr [rip + 0x23897]
0x6fe9: lea      rdx, [rip + 0x198f8]
0x6ff0: xor      eax, eax
0x6ff2: mov      esi, 1
0x6ff7: call     0x4e60
0x6ffc: jmp      0x5127
0x5127
0x7001: mov      rdi, r13
0x7004: call     0x1d7a0
0x7009: mov      r12, rax
0x700c: jmp      0x5532
0x5532
0x7011: lea      rbx, [rip + 0x19543]
0x7018: jmp      0x529c
0x529c
0x701d: mov      rdi, qword ptr [rip + 0x238ac]
0x7024: xor      edx, edx
0x7026: lea      rsi, [rip + 0x19505]
0x702d: call     0x8070
0x7032: cmp      al, 0x79
0x7034: jne      0x5730
0x703a: jmp      0x6442
0x6442
0x703f: mov      edi, 1
0x7044: xor      eax, eax
0x7046: call     0x7bd0
0x704b: jmp      0x59ce

```

```

0x59ce
0x7050: mov     esi, 1
0x7055: lea     rdi, [rip + 0x19664]
0x705c: call    0x17e50
0x7061: jmp     0x53df
0x53df
0x7066: mov     rcx, qword ptr [rip + 0x2387b]
0x706d: mov     rdi, qword ptr [rip + 0x2380c]
0x7074: xor     eax, eax
0x7076: mov     r8d, 0x800
0x707c: lea     rdx, [rip + 0x19895]
0x7083: mov     esi, 1
0x7088: call    0x4e60
0x708d: jmp     0x54a3
0x54a3
0x7092: xor     esi, esi
0x7094: lea     rdi, [rip + 0x195fa]
0x709b: call    0x17e50
0x70a0: mov     r12d, eax
0x70a3: jmp     0x53a3
0x53a3
0x70a8: mov     r8, r12
0x70ab: lea     rcx, [rip + 0x19696]
0x70b2: jmp     0x5f42
0x5f42
0x70b7: mov     rdi, qword ptr [r13 + 0x10]
0x70bb: test    rdi, rdi
0x70be: je      0x6955
0x70c4: xor     eax, eax
0x70c6: call    0x19940
0x70cb: mov     r8d, 1
0x70d1: mov     esi, 0x21
0x70d6: mov     rdx, qword ptr [rip + 0x237cb]
0x70dd: mov     rdi, qword ptr [rip + 0x237cc]
0x70e4: mov     rcx, rax
0x70e7: mov     rbp, rax
0x70ea: call    0x13e30
0x70ef: mov     rdi, qword ptr [rip + 0x237ba]
0x70f6: call    0x11100
0x70fb: mov     rdi, rax
0x70fe: xor     eax, eax
0x7100: call    0x89d0
0x7105: mov     rdi, rbp
0x7108: xor     ebp, ebp
0x710a: call    0x18d10
0x710f: jmp     0x565a
0x565a
0x7114: mov     qword ptr [rsp + 0x40], 0

```

```

0x711d: lea     r12, [rsp + 0x40]
0x7122: lea     rbp, [rip + 0x1966f]
0x7129: jmp     0x714f
0x714f:
0x712b: lea     rdi, [rip + 0x22ade]
0x7132: call    0xdb60
0x7137: xor     eax, eax
0x7139: call    0xa5a0
0x713e: xor     eax, eax
0x7140: call    0xa820
0x7145: mov     edi, 2
0x714a: call    0x4ea0
0x714f: mov     rsi, r12
0x7152: xor     ecx, ecx
0x7154: xor     edx, edx
0x7156: mov     rdi, rbp
0x7159: call    0x83e0
0x715e: xor     eax, eax
0x7160: call    0xa820
0x7165: mov     rsi, qword ptr [rsp + 0x40]
0x716a: test    rsi, rsi
0x716d: je      0x712b
0x716f: mov     rdi, qword ptr [rsp + 0x38]
0x7174: call    0x4b30
0x7179: test    eax, eax
0x717b: jne     0x712b
0x717d: xor     eax, eax
0x717f: call    0xa820
0x7184: mov     rdi, qword ptr [rsp + 0x38]
0x7189: call    0x18d10
0x718e: mov     rdi, qword ptr [rsp + 0x40]
0x7193: call    0x18d10
0x7198: jmp     0x5a59
0x5a59:
0x719d: xor     eax, eax
0x719f: xor     ebp, ebp
0x71a1: call    0xa5a0
0x71a6: mov     rdi, qword ptr [r13]
0x71aa: call    0x195e0
0x71af: jmp     0x565a
0x565a:
0x71b4: mov     rdi, qword ptr [rip + 0x236f5]
0x71bb: xor     ebp, ebp
0x71bd: call    0x110b0
0x71c2: jmp     0x565a
0x565a:
0x71c7: cmp     byte ptr [rbp + 1], 0
0x71cb: mov     rdx, qword ptr [rip + 0x236d6]

```

```

0x71d2: mov     rdi, qword ptr [rip + 0x236d7]
0x71d9: jne     0x72a7
0x71df: mov     r8d, 1
0x71e5: xor     ecx, ecx
0x71e7: mov     esi, 5
0x71ec: call    0x13e30
0x71f1: jmp     0x6f11
0x6f11
0x71f6: mov     rcx, qword ptr [rip + 0x236cb]
0x71fd: mov     rdx, qword ptr [rip + 0x236a4]
0x7204: mov     esi, 1
0x7209: mov     r8d, 1
0x720f: mov     rdi, qword ptr [rip + 0x2369a]
0x7216: call    0x13e30
0x721b: mov     rdi, qword ptr [rip + 0x2368e]
0x7222: mov     ebp, eax
0x7224: call    0x11100
0x7229: mov     rdi, rax
0x722c: call    0xc4b0
0x7231: xor     edi, edi
0x7233: call    0xd1a0
0x7238: xor     eax, eax
0x723a: call    0x9640
0x723f: xor     eax, eax
0x7241: call    0xc8c0
0x7246: mov     r14, rax
0x7249: test    ebp, ebp
0x724b: je      0x5730
0x7251: xor     ebp, ebp
0x7253: jmp     0x565a
0x565a
0x7258: mov     rdi, qword ptr [rsp + 0x48]
0x725d: mov     edx, 0x1c
0x7262: xor     esi, esi
0x7264: call    0x8070
0x7269: jmp     0x5723
0x5723
0x726e: lea     rcx, [rbp + 1]
0x7272: mov     r8d, 1
0x7278: mov     esi, 0x25
0x727d: call    0x13e30
0x7282: jmp     0x6f11
0x6f11
0x7287: mov     rdi, qword ptr [rsp + 0x48]
0x728c: mov     edx, 0x1c
0x7291: xor     esi, esi
0x7293: call    0x8070
0x7298: xor     eax, eax

```

```

0x729a: call    0xc8c0
0x729f: mov     r14, rax
0x72a2: jmp     0x6611
0x6611
0x72a7: lea     rcx, [rbp + 1]
0x72ab: mov     r8d, 1
0x72b1: mov     esi, 0x24
0x72b6: call    0x13e30
0x72bb: jmp     0x6f11
0x6f11
0x72c0: mov     rdi, qword ptr [rsp + 0x48]
0x72c5: call    0x185d0
0x72ca: mov     ebp, eax
0x72cc: test    eax, eax
0x72ce: je      0x73fa
0x72d4: xor     edi, edi
0x72d6: call    0xd1a0
0x72db: mov     rdi, qword ptr [rsp + 0x48]
0x72e0: call    0xd610
0x72e5: mov     rdi, qword ptr [rsp + 0x48]
0x72ea: mov     esi, ebp
0x72ec: call    0x193f0
0x72f1: xor     edi, edi
0x72f3: call    0xdbb0
0x72f8: jmp     0x6f11
0x6f11
0x72fd: mov     rdx, qword ptr [rip + 0x235a4]
0x7304: mov     rdi, qword ptr [rip + 0x235a5]
0x730b: xor     ecx, ecx
0x730d: mov     esi, 7
0x7312: mov     r8d, 1
0x7318: call    0x13e30
0x731d: xor     edi, edi
0x731f: call    0xd1a0
0x7324: jmp     0x5723
0x5723
0x7329: mov     rdi, qword ptr [rip + 0x23580]
0x7330: mov     r15d, dword ptr [rsp + 0x10]
0x7335: call    0xf480
0x733a: mov     rdi, qword ptr [rip + 0x23567]
0x7341: call    0xf480
0x7346: xor     eax, eax
0x7348: call    0xa480
0x734d: xor     eax, eax
0x734f: call    0xe4e0
0x7354: xor     eax, eax
0x7356: call    0xa820
0x735b: xor     eax, eax

```

```

0x735d: call    0x18330
0x7362: mov     r8, rax
0x7365: mov     eax, 0x3c
0x736a: sub     eax, dword ptr [r8]
0x736d: mov     edi, eax
0x736f: call    0x4aa0
0x7374: mov     dword ptr [rip + 0x23566], 0
0x737e: jmp     0x6447
0x6447
0x7383: mov     rdi, r14
0x7386: xor     ebp, ebp
0x7388: call    0xc7e0
0x738d: xor     eax, eax
0x738f: call    0xc8c0
0x7394: mov     rdi, qword ptr [rip + 0x23515]
0x739b: mov     r14, rax
0x739e: call    0xf480
0x73a3: mov     rdi, qword ptr [rip + 0x234fe]
0x73aa: call    0xf480
0x73af: xor     eax, eax
0x73b1: call    0xa480
0x73b6: xor     eax, eax
0x73b8: call    0xe4e0
0x73bd: xor     eax, eax
0x73bf: call    0xa820
0x73c4: xor     eax, eax
0x73c6: call    0x18330
0x73cb: mov     r8, rax
0x73ce: mov     eax, 0x3c
0x73d3: sub     eax, dword ptr [r8]
0x73d6: mov     edi, eax
0x73d8: call    0x4aa0
0x73dd: mov     dword ptr [rip + 0x234fd], 0
0x73e7: jmp     0x565a
0x565a
0x73ec: lea     rdi, [rip + 0x1955d]
0x73f3: xor     eax, eax
0x73f5: call    0x8b00
0x73fa: mov     rdi, qword ptr [rsp + 0x48]
0x73ff: xor     ebp, ebp
0x7401: call    0xc4b0
0x7406: xor     edi, edi
0x7408: call    0xd1a0
0x740d: mov     rdi, qword ptr [rsp + 0x48]
0x7412: call    0xd610
0x7417: mov     rdi, qword ptr [rsp + 0x48]
0x741c: xor     esi, esi
0x741e: call    0x193f0

```



```

0x7423: xor     edi, edi
0x7425: call    0xdbb0
0x742a: xor     edi, edi
0x742c: call    0xd1a0
0x7431: mov     rdi, qword ptr [rip + 0x23478]
0x7438: call    0xf480
0x743d: mov     rdi, qword ptr [rip + 0x23464]
0x7444: call    0xf480
0x7449: xor     eax, eax
0x744b: call    0xa480
0x7450: mov     rdi, qword ptr [rip + 0x23471]
0x7457: call    0xbb10
0x745c: xor     eax, eax
0x745e: mov     dword ptr [rip + 0x23474], 1
0x7468: call    0xc8c0
0x746d: mov     r14, rax
0x7470: jmp     0x565a
0x565a
0x7475: call    0x49d0
0x747a: nop     word ptr [rax + rax]
0x7480: endbr64
0x7484: xor     ebp, ebp
0x7486: mov     r9, rdx
0x7489: pop     rsi
0x748a: mov     rdx, rsp
0x748d: and     rsp, 0xfffffffffffffff0
0x7491: push    rax
0x7492: push    rsp
0x7493: xor     r8d, r8d
0x7496: xor     ecx, ecx
0x7498: lea     rdi, [rip - 0x256f]
0x749f: call    qword ptr [rip + 0x21b23]
0x74a5: hlt
0x74a6: nop     word ptr cs:[rax + rax]
0x74b0: lea     rdi, [rip + 0x23379]
0x74b7: lea     rax, [rip + 0x23372]
0x74be: cmp     rax, rdi
0x74c1: je      0x74d8
0x74c3: mov     rax, qword ptr [rip + 0x21b06]
0x74ca: test    rax, rax
0x74cd: je      0x74d8
0x74cf: jmp     rax
rax
->->->                                indirect JMP
0x67c0: test    rax, rax
0x67c3: je      0x57dc
0x67c9: mov     rdi, qword ptr [rip + 0x240e0]
0x67d0: call    0xf480

```

```

0x67d5: mov     rdi, qword ptr [rip + 0x240cc]
0x67dc: call    0xf480
0x67e1: cmp     qword ptr [r13 + 8], 0
0x67e6: je      0x70b7
0x67ec: lea     rbp, [rsp + 0x60]
0x67f1: mov     rcx, qword ptr [rip + 0x240b0]
0x67f8: xor     eax, eax
0x67fa: mov     rdi, r13
0x67fd: mov     rdx, qword ptr [rip + 0x240ac]
0x6804: mov     rsi, rbp
0x6807: mov     qword ptr [rsp + 0x60], 0
0x6810: call    0x8b50
0x6815: test    eax, eax
0x6817: je      0x6c1f
0x681d: jle     0x5730
0x6823: mov     rdi, qword ptr [r13]
0x6827: mov     dword ptr [rsp + 0xc], eax
0x682b: call    0x49b0
0x6830: mov     rbp, qword ptr [rsp + 0x60]
0x6835: mov     r12, rax
0x6838: mov     rdi, rbp
0x683b: call    0x49b0
0x6840: lea     rdi, [r12 + rax + 0x21]
0x6845: call    0x1d670
0x684a: mov     r8, qword ptr [r13]
0x684e: mov     r9, rbp
0x6851: or      rdx, 0xffffffffffffffff
0x6855: lea     rcx, [rip + 0x19cd9]
0x685c: mov     r12, rax
0x685f: mov     rdi, rax
0x6862: mov     esi, 1
0x6867: xor     eax, eax
0x6869: call    0x4f10
0x686e: mov     edx, 1
0x6873: mov     esi, 1
0x6878: mov     rdi, r12
0x687b: call    0xe480
0x6880: xor     eax, eax
0x6882: call    0xa820
0x6887: mov     rdi, r12
0x688a: call    0x18d10
0x688f: cmp     byte ptr [r13 + 0x1a], 0
0x6894: jne     0x6dc9
0x689a: mov     rdi, rbp
0x689d: call    0x185d0
0x68a2: test    eax, eax
0x68a4: je      0x6dbc
0x68aa: movsx   esi, byte ptr [r13 + 0x1a]

```

```

0x68af: mov     rdi, rbp
0x68b2: call    0x193f0
0x68b7: mov     rdi, rbp
0x68ba: mov     r12d, eax
0x68bd: call    0x18d10
0x68c2: cmp     byte ptr [r13 + 0x1a], 0
0x68c7: je      0x6ccf
0x68cd: mov     eax, r12d
0x68d0: and     eax, 0x7f
0x68d3: add     eax, 1
0x68d6: sub     al, 1
0x68d8: jg      0x6f1d
0x68de: test    r12b, 0x7f
0x68e2: jne     0x6955
0x68e4: and     r12d, 0xff00
0x68eb: jne     0x719d
0x68f1: mov     rdi, qword ptr [r13 + 0x10]
0x68f5: test    rdi, rdi
0x68f8: je      0x6943
0x68fa: xor     eax, eax
0x68fc: call    0x19940
0x6901: mov     r8d, 1
0x6907: mov     esi, 0x21
0x690c: mov     rdx, qword ptr [rip + 0x23f95]
0x6913: mov     rdi, qword ptr [rip + 0x23f96]
0x691a: mov     rcx, rax
0x691d: mov     r12, rax
0x6920: call    0x13e30
0x6925: mov     rdi, qword ptr [rip + 0x23f84]
0x692c: call    0x11100
0x6931: mov     rdi, rax
0x6934: xor     eax, eax
0x6936: call    0x89d0
0x693b: mov     rdi, r12
0x693e: call    0x18d10
0x6943: cmp     dword ptr [rsp + 0xc], 2
0x6948: je      0x71b4
0x694e: cmp     dword ptr [rsp + 0xc], 3
0x6953: je      0x695c
0x6955: xor     ebp, ebp
0x6957: jmp     0x565a
0x565a
0x695c: mov     rdi, qword ptr [rip + 0x23f45]
0x6963: xor     ebp, ebp
0x6965: call    0x110b0
0x696a: jmp     0x565a
0x565a
0x696f: lea     eax, [r12 + 0x22]

```

```

0x6974: cmp      eax, 0x29
0x6977: ja         0x65e3
0x697d: movsxd    rax, dword ptr [rbp + rax*4]
0x6982: add       rax, rbp
0x6985: jmp       rax
rax
                                ->->->                                indirect JMP
0x57f5: test      ebp, ebp
0x57f7: je        0x5730
0x57fd: mov       rdi, r14
0x5800: call      0xc7e0
0x5805: jmp       0x5813
0x5813
0x5807: mov       esi, 3
0x580c: mov       edi, ebp
0x580e: call      0x7570
0x5813: mov       eax, dword ptr [rsp + 0x28]
0x5817: lea       edx, [rax - 1]
0x581a: mov       dword ptr [rsp + 0x28], edx
0x581e: test      eax, eax
0x5820: jne       0x5807
0x5822: jmp       0x5725
0x5725
0x5827: mov       esi, 0xffffffff
0x582c: mov       rdi, qword ptr [rip + 0x2507d]
0x5833: sub       esi, ebp
0x5835: xor       ecx, ecx
0x5837: xor       edx, edx
0x5839: mov       r8d, 1
0x583f: call      0x13e30
0x5844: jmp       0x5730
0x5730
0x5849: mov       esi, 0xffffffff
0x584e: jmp       0x582c
0x582c
0x5850: mov       rdi, qword ptr [rip + 0x25059]
0x5857: xor       esi, esi
0x5859: call      0x14d90
0x585e: mov       rax, qword ptr [rip + 0x2504b]
0x5865: mov       esi, 1
0x586a: mov       rdi, qword ptr [rip + 0x25037]
0x5871: mov       qword ptr [rip + 0x25028], rax
0x5878: mov       qword ptr [rip + 0x25029], rax
0x587f: xor       eax, eax
0x5881: cmp       dword ptr [rip + 0x25058], 0
0x5888: sete      al
0x588b: mov       qword ptr [rip + 0x2501e], rdi
0x5892: mov       dword ptr [rip + 0x25048], eax

```

```

0x5898: call    0x14d90
0x589d: mov     rdi, qword ptr [rip + 0x2500c]
0x58a4: call    0xf570
0x58a9: mov     rdi, r14
0x58ac: call    0xc7e0
0x58b1: xor     eax, eax
0x58b3: call    0x9640
0x58b8: xor     eax, eax
0x58ba: call    0xc8c0
0x58bf: mov     rdi, qword ptr [rip + 0x24fea]
0x58c6: mov     r14, rax
0x58c9: call    0x11100
0x58ce: mov     rdi, rax
0x58d1: call    0xc4b0
0x58d6: mov     rdi, qword ptr [rip + 0x24fd3]
0x58dd: call    0xf480
0x58e2: mov     rdi, qword ptr [rip + 0x24fbf]
0x58e9: call    0xf480
0x58ee: xor     eax, eax
0x58f0: call    0xa480
0x58f5: mov     rdi, qword ptr [rip + 0x24fac]
0x58fc: call    0xf580
0x5901: xor     eax, eax
0x5903: mov     edi, 1
0x5908: mov     dword ptr [rip + 0x24346], 0
0x5912: call    0x7890
0x5917: mov     rdi, qword ptr [rip + 0x24f92]
0x591e: xor     ecx, ecx
0x5920: xor     edx, edx
0x5922: mov     r8d, 1
0x5928: mov     esi, 0xe
0x592d: call    0x13e30
0x5932: mov     rdi, qword ptr [rip + 0x24f6f]
0x5939: xor     ecx, ecx
0x593b: xor     edx, edx
0x593d: mov     r8d, 1
0x5943: mov     esi, 0xe
0x5948: call    0x13e30
0x594d: mov     rdi, qword ptr [rip + 0x24f5c]
0x5954: call    0x11010
0x5959: jmp     0x5730
0x5730:
0x595e: xor     eax, eax
0x5960: call    0x8960
0x5965: mov     rdi, qword ptr [rip + 0x24f44]
0x596c: call    0x11100
0x5971: mov     rdi, rax
0x5974: xor     eax, eax

```

```

0x5976: call    0x89d0
0x597b: jmp     0x5730
0x5730
0x5980: mov     rdi, qword ptr [rip + 0x24f29]
0x5987: call    0xf480
0x598c: mov     rdi, qword ptr [rip + 0x24f15]
0x5993: call    0xf480
0x5998: xor     eax, eax
0x599a: call    0xa480
0x599f: mov     rdi, qword ptr [rip + 0x24f02]
0x59a6: call    0xf570
0x59ab: xor     eax, eax
0x59ad: mov     edi, 1
0x59b2: mov     dword ptr [rip + 0x2429c], 1
0x59bc: call    0x7890
0x59c1: cmp     dword ptr [rip + 0x24fd0], 0xb
0x59c8: jle     0x703f
0x59ce: mov     rdi, qword ptr [rip + 0x24edb]
0x59d5: xor     ecx, ecx
0x59d7: xor     edx, edx
0x59d9: mov     esi, 0xb
0x59de: mov     r8d, 1
0x59e4: call    0x13e30
0x59e9: mov     rdi, qword ptr [rip + 0x24eb8]
0x59f0: xor     ecx, ecx
0x59f2: xor     edx, edx
0x59f4: mov     r8d, 1
0x59fa: mov     esi, 0xb
0x59ff: call    0x13e30
0x5a04: mov     rdi, qword ptr [rip + 0x24ea5]
0x5a0b: call    0x11010
0x5a10: mov     rdi, qword ptr [rip + 0x24e91]
0x5a17: call    0x11010
0x5a1c: jmp     0x5730
0x5730
0x5a21: xor     edi, edi
0x5a23: call    0xca20
0x5a28: lea     rsi, [rsp + 0x38]
0x5a2d: xor     ecx, ecx
0x5a2f: xor     edx, edx
0x5a31: mov     qword ptr [rsp + 0x38], 0
0x5a3a: lea     rdi, [rip + 0x1ad44]
0x5a41: call    0x83e0
0x5a46: mov     rax, qword ptr [rsp + 0x38]
0x5a4b: test    rax, rax
0x5a4e: je      0x5a59
0x5a50: cmp     byte ptr [rax], 0
0x5a53: jne     0x7114

```

```

0x5a59: mov     edi, 1
0x5a5e: call    0xca20
0x5a63: jmp     0x5730
0x5730:
0x5a68: mov     dword ptr [rsp + 0x30], 0
0x5a70: lea     rcx, [rsp + 0x30]
0x5a75: mov     rdx, qword ptr [rip + 0x24e2c]
0x5a7c: mov     esi, 0x2b
0x5a81: mov     rdi, qword ptr [rip + 0x24e28]
0x5a88: mov     r8d, 1
0x5a8e: call    0x13e30
0x5a93: jmp     0x5730
0x5730:
0x5a98: mov     dword ptr [rsp + 0x2c], 0
0x5aa0: lea     rcx, [rsp + 0x2c]
0x5aa5: mov     rdx, qword ptr [rip + 0x24dfc]
0x5aac: mov     esi, 0x2c
0x5ab1: mov     rdi, qword ptr [rip + 0x24df8]
0x5ab8: mov     r8d, 1
0x5abe: call    0x13e30
0x5ac3: jmp     0x5730
0x5730:
0x5ac8: mov     dword ptr [rsp + 0x30], 1
0x5ad0: jmp     0x5a70
0x5a70:
0x5ad2: mov     dword ptr [rsp + 0x2c], 1
0x5ada: jmp     0x5aa0
0x5aa0:
0x5adc: mov     rdx, qword ptr [rip + 0x24dc5]
0x5ae3: mov     rdi, qword ptr [rip + 0x24dc6]
0x5aea: xor     ecx, ecx
0x5aec: mov     esi, 0x2d
0x5af1: mov     r8d, 1
0x5af7: call    0x13e30
0x5afc: mov     rdi, r14
0x5aff: call    0xc970
0x5b04: xor     eax, eax
0x5b06: call    0x9640
0x5b0b: xor     eax, eax
0x5b0d: call    0xc8c0
0x5b12: mov     rdi, qword ptr [rip + 0x24d97]
0x5b19: mov     r14, rax
0x5b1c: call    0x11100
0x5b21: mov     rdi, rax
0x5b24: call    0xc4b0
0x5b29: jmp     0x5730
0x5730:
0x5b2e: mov     rdx, qword ptr [rip + 0x24d73]

```

```

0x5b35: mov     rdi, qword ptr [rip + 0x24d74]
0x5b3c: xor     ecx, ecx
0x5b3e: mov     esi, 0x2e
0x5b43: mov     r8d, 1
0x5b49: call    0x13e30
0x5b4e: jmp     0x5730
0x5730:
0x5b53: lea     r12, [rsp + 0x58]
0x5b58: mov     rcx, qword ptr [r13 + 0x28]
0x5b5c: lea     rdx, [rip + 0x1b230]
0x5b63: mov     rsi, r12
0x5b66: lea     rdi, [rip + 0x1ac46]
0x5b6d: call    0x83e0
0x5b72: test    rax, rax
0x5b75: je      0x5730
0x5b7b: mov     rcx, qword ptr [rsp + 0x58]
0x5b80: cmp     byte ptr [rcx], 0
0x5b83: je      0x5730
0x5b89: mov     rdx, qword ptr [rip + 0x24d18]
0x5b90: mov     r8d, 1
0x5b96: mov     esi, 0x2f
0x5b9b: mov     rdi, qword ptr [rip + 0x24d0e]
0x5ba2: call    0x13e30
0x5ba7: mov     rdi, qword ptr [rsp + 0x58]
0x5bac: call    0x18d10
0x5bb1: mov     qword ptr [rsp + 0x58], 0
0x5bba: jmp     0x5730
0x5730:
0x5bbf: mov     r8d, dword ptr [rsp + 0x28]
0x5bc4: mov     rcx, qword ptr [rsp + 0x58]
0x5bc9: mov     esi, 0x30
0x5bce: mov     rdx, qword ptr [rip + 0x24cd3]
0x5bd5: mov     rdi, qword ptr [rip + 0x24cd4]
0x5bdc: call    0x13e30
0x5be1: jmp     0x5730
0x5730:
0x5be6: mov     r8d, dword ptr [rsp + 0x28]
0x5beb: mov     rcx, qword ptr [rsp + 0x58]
0x5bf0: mov     esi, 0x31
0x5bf5: mov     rdx, qword ptr [rip + 0x24cac]
0x5bfc: mov     rdi, qword ptr [rip + 0x24cad]
0x5c03: call    0x13e30
0x5c08: jmp     0x5730
0x5730:
0x5c0d: mov     rdx, qword ptr [rip + 0x24c94]
0x5c14: mov     rdi, qword ptr [rip + 0x24c95]
0x5c1b: xor     ecx, ecx
0x5c1d: mov     esi, 0x32

```



```

0x5c22: mov     r8d, 1
0x5c28: call    0x13e30
0x5c2d: jmp     0x5730
0x5730:
0x5c32: mov     rdx, qword ptr [rip + 0x24c6f]
0x5c39: mov     rdi, qword ptr [rip + 0x24c70]
0x5c40: xor     ecx, ecx
0x5c42: mov     esi, 0x33
0x5c47: mov     r8d, 1
0x5c4d: call    0x13e30
0x5c52: jmp     0x5730
0x5730:
0x5c57: mov     r8d, dword ptr [rsp + 0x28]
0x5c5c: mov     rdx, qword ptr [rip + 0x24c45]
0x5c63: xor     ecx, ecx
0x5c65: mov     esi, 0x1b
0x5c6a: mov     rdi, qword ptr [rip + 0x24c3f]
0x5c71: call    0x13e30
0x5c76: jmp     0x5730
0x5730:
0x5c7b: mov     r8d, dword ptr [rsp + 0x28]
0x5c80: mov     rdx, qword ptr [rip + 0x24c21]
0x5c87: xor     ecx, ecx
0x5c89: mov     esi, 0x1c
0x5c8e: mov     rdi, qword ptr [rip + 0x24c1b]
0x5c95: call    0x13e30
0x5c9a: jmp     0x5730
0x5730:
0x5c9f: cmp     dword ptr [rip + 0x23fb2], 0
0x5ca6: je      0x5850
0x5cac: test    byte ptr [rsp + 0x28], 1
0x5cb1: je      0x5730
0x5cb7: mov     rdi, qword ptr [rip + 0x24bf2]
0x5cbe: xor     esi, esi
0x5cc0: call    0x14d90
0x5cc5: mov     rax, qword ptr [rip + 0x24be4]
0x5ccc: mov     esi, 1
0x5cd1: mov     rdi, qword ptr [rip + 0x24bd0]
0x5cd8: mov     qword ptr [rip + 0x24bc1], rax
0x5cdf: mov     qword ptr [rip + 0x24bc2], rax
0x5ce6: xor     eax, eax
0x5ce8: cmp     dword ptr [rip + 0x24bf1], 0
0x5cef: sete    al
0x5cf2: mov     qword ptr [rip + 0x24bb7], rdi
0x5cf9: mov     dword ptr [rip + 0x24be1], eax
0x5cff: call    0x14d90
0x5d04: mov     rdi, r14
0x5d07: call    0xc7e0

```

```

0x5d0c: jmp      0x5b04
0x5b04:
0x5d11: lea      r12, [rsp + 0x58]
0x5d16: mov      rcx, qword ptr [r13 + 0x28]
0x5d1a: xor      edx, edx
0x5d1c: lea      rdi, [rip + 0x1aa29]
0x5d23: mov      rsi, r12
0x5d26: call     0x83e0
0x5d2b: test     rax, rax
0x5d2e: je       0x5730
0x5d34: mov      rdi, qword ptr [rsp + 0x58]
0x5d39: cmp      byte ptr [rdi], 0
0x5d3c: je       0x5730
0x5d42: xor      eax, eax
0x5d44: call     0x19940
0x5d49: mov      r8d, 1
0x5d4f: mov      esi, 0x21
0x5d54: mov      rdx, qword ptr [rip + 0x24b4d]
0x5d5b: mov      rdi, qword ptr [rip + 0x24b4e]
0x5d62: mov      rcx, rax
0x5d65: mov      rbp, rax
0x5d68: call     0x13e30
0x5d6d: mov      rdi, qword ptr [rip + 0x24b3c]
0x5d74: call     0x11100
0x5d79: mov      rdi, rax
0x5d7c: xor      eax, eax
0x5d7e: call     0x89d0
0x5d83: mov      rdi, rbp
0x5d86: call     0x18d10
0x5d8b: mov      rdi, qword ptr [rsp + 0x58]
0x5d90: call     0x18d10
0x5d95: mov      qword ptr [rsp + 0x58], 0
0x5d9e: jmp      0x5afc
0x5afc:
0x5da3: lea      r12, [rsp + 0x58]
0x5da8: mov      rcx, qword ptr [r13 + 0x28]
0x5dac: xor      edx, edx
0x5dae: lea      rdi, [rip + 0x1ac3b]
0x5db5: mov      rsi, r12
0x5db8: call     0x83e0
0x5dbd: test     rax, rax
0x5dc0: je       0x5730
0x5dc6: mov      rcx, qword ptr [rsp + 0x58]
0x5dcb: cmp      byte ptr [rcx], 0
0x5dce: je       0x5730
0x5dd4: mov      rdx, qword ptr [rip + 0x24acd]
0x5ddb: mov      r8d, 1
0x5de1: mov      esi, 0x24

```

```

0x5de6: jmp      0x5b9b
0x5b9b:
0x5deb: lea       r12, [rsp + 0x58]
0x5df0: mov       rcx, qword ptr [r13 + 0x28]
0x5df4: xor       edx, edx
0x5df6: lea       rdi, [rip + 0x1ac23]
0x5dfd: mov       rsi, r12
0x5e00: call      0x83e0
0x5e05: test      rax, rax
0x5e08: je        0x5730
0x5e0e: mov       rcx, qword ptr [rsp + 0x58]
0x5e13: cmp       byte ptr [rcx], 0
0x5e16: je        0x5730
0x5e1c: mov       rdx, qword ptr [rip + 0x24a85]
0x5e23: mov       r8d, 1
0x5e29: mov       esi, 0x25
0x5e2e: jmp      0x5b9b
0x5b9b:
0x5e33: mov       rdx, qword ptr [rip + 0x24a6e]
0x5e3a: mov       r8d, 1
0x5e40: mov       esi, 0x21
0x5e45: mov       rdi, qword ptr [rip + 0x24a64]
0x5e4c: mov       rcx, qword ptr [rdx + 0x18]
0x5e50: call      0x13e30
0x5e55: mov       rdi, qword ptr [rip + 0x24a54]
0x5e5c: call      0x11100
0x5e61: mov       rdi, rax
0x5e64: xor       eax, eax
0x5e66: call      0x89d0
0x5e6b: mov       rdi, r14
0x5e6e: call      0xc7e0
0x5e73: jmp      0x5b04
0x5b04:
0x5e78: mov       rdx, qword ptr [rip + 0x24a31]
0x5e7f: mov       r8d, 1
0x5e85: mov       esi, 0x21
0x5e8a: mov       rdi, qword ptr [rip + 0x24a17]
0x5e91: mov       rcx, qword ptr [rdx + 0x18]
0x5e95: call      0x13e30
0x5e9a: mov       rdi, qword ptr [rip + 0x24a07]
0x5ea1: call      0x11100
0x5ea6: mov       rdi, rax
0x5ea9: xor       eax, eax
0x5eab: call      0x89d0
0x5eb0: jmp      0x5730
0x5730:
0x5eb5: mov       rax, qword ptr [rip + 0x249ec]
0x5ebc: mov       rdi, qword ptr [rax + 0x78]

```

```

0x5ec0: mov     r12, qword ptr [rax + 0x18]
0x5ec4: add     rdi, 5
0x5ec8: call    0x1d670
0x5ecd: mov     rbp, rax
0x5ed0: mov     rdi, qword ptr [rsp + 0x50]
0x5ed5: call    0x49b0
0x5eda: mov     rdi, r14
0x5edd: mov     r13, rax
0x5ee0: call    0xc7e0
0x5ee5: test    r13, r13
0x5ee8: je      0x5f1a
0x5eea: mov     r13, qword ptr [rsp + 0x50]
0x5eef: xor     eax, eax
0x5ef1: call    0xca00
0x5ef6: cmp     byte ptr [r13 + rax - 1], 0x2f
0x5efc: je      0x6f46
0x5f02: mov     r13, qword ptr [rsp + 0x50]
0x5f07: xor     eax, eax
0x5f09: call    0xca00
0x5f0e: cmp     byte ptr [r13 + rax - 1], 0x20
0x5f14: je      0x6f46
0x5f1a: mov     rdi, r12
0x5f1d: call    0x49b0
0x5f22: mov     rdi, r12
0x5f25: mov     rsi, rax
0x5f28: call    0x18530
0x5f2d: mov     r8, r12
0x5f30: lea     rcx, [rip + 0x1a809]
0x5f37: test    eax, eax
0x5f39: jne     0x5f42
0x5f3b: lea     rcx, [rip + 0x1a805]
0x5f42: or      rdx, 0xffffffffffffffff
0x5f46: mov     esi, 1
0x5f4b: mov     rdi, rbp
0x5f4e: xor     eax, eax
0x5f50: call    0x4f10
0x5f55: mov     rdi, rbp
0x5f58: call    0x49b0
0x5f5d: mov     rdi, rbp
0x5f60: mov     rsi, rax
0x5f63: call    0x184d0
0x5f68: mov     rdi, rbp
0x5f6b: call    0xd610
0x5f70: mov     rdi, rbp
0x5f73: call    0x18d10
0x5f78: xor     eax, eax
0x5f7a: call    0xc8c0
0x5f7f: mov     r14, rax

```

```

0x5f82: jmp      0x5730
0x5730
0x5f87: mov      rdi, r14
0x5f8a: lea      r12, [rip + 0x1b488]
0x5f91: lea      rbp, [rip + 0x1b2fc]
0x5f98: call     0xc7e0
0x5f9d: mov      rdi, qword ptr [rip + 0x2490c]
0x5fa4: call     0x100c0
0x5fa9: jmp      0x6029
0x6029
0x5fab: mov      rdx, qword ptr [rip + 0x248fe]
0x5fb2: cdqe
0x5fb4: imul     rax, rax, 0x50
0x5fb8: add      rax, qword ptr [rdx + 0x38]
0x5fbc: mov      r14, qword ptr [rax]
0x5fbf: mov      rdi, r14
0x5fc2: call     0x49b0
0x5fc7: lea      rdi, [rax + 5]
0x5fcb: call     0x1d670
0x5fd0: mov      rdi, r14
0x5fd3: mov      r13, rax
0x5fd6: call     0x49b0
0x5fdb: mov      rdi, r14
0x5fde: mov      rsi, rax
0x5fe1: call     0x18530
0x5fe6: mov      r8, r14
0x5fe9: mov      rcx, rbp
0x5fec: test     eax, eax
0x5fee: jne      0x5ff3
0x5ff0: mov      rcx, r12
0x5ff3: or       rdx, 0xffffffffffffffff
0x5ff7: mov      esi, 1
0x5ffc: mov      rdi, r13
0x5fff: xor     eax, eax
0x6001: call     0x4f10
0x6006: mov      rdi, r13
0x6009: call     0x49b0
0x600e: mov      rdi, r13
0x6011: mov      rsi, rax
0x6014: call     0x184d0
0x6019: mov      rdi, r13
0x601c: call     0xd610
0x6021: mov      rdi, r13
0x6024: call     0x18d10
0x6029: mov      rdi, qword ptr [rip + 0x24880]
0x6030: call     0x100e0
0x6035: cmp      eax, -1
0x6038: jne      0x5fab

```

```

0x603e: lea     rdi, [rip + 0x1a700]
0x6045: call    0xd610
0x604a: xor     eax, eax
0x604c: call    0xc8c0
0x6051: mov     r14, rax
0x6054: jmp     0x5730
0x5730
0x6059: mov     rdi, r14
0x605c: call    0xc7e0
0x6061: mov     rdi, r15
0x6064: call    0xdb10
0x6069: mov     rbp, qword ptr [rsp + 0x50]
0x606e: mov     rdi, rbp
0x6071: call    0x49b0
0x6076: sub     eax, 1
0x6079: movsxd  rdx, eax
0x607c: jmp     0x6097
0x6097
0x607e: lea     rcx, [rbp + rdx]
0x6083: sub     rdx, 1
0x6087: cmp     byte ptr [rcx], 0x20
0x608a: jne     0x609b
0x608c: mov     byte ptr [rcx], 0
0x608f: mov     rbp, qword ptr [rsp + 0x50]
0x6094: sub     eax, 1
0x6097: test    eax, eax
0x6099: jns     0x607e
0x609b: movzx   eax, byte ptr [rbp]
0x609f: cmp     al, 0x2b
0x60a1: je      0x71c7
0x60a7: jg      0x6edf
0x60ad: test    al, al
0x60af: je      0x71f6
0x60b5: cmp     al, 0x2a
0x60b7: jne     0x60c3
0x60b9: cmp     byte ptr [rbp + 1], 0
0x60bd: je      0x72fd
0x60c3: xor     eax, eax
0x60c5: lea     rsi, [rsp + 0x48]
0x60ca: mov     rdi, rbp
0x60cd: call    0x1b5b0
0x60d2: test    eax, eax
0x60d4: js      0x7258
0x60da: mov     rdi, qword ptr [rsp + 0x48]
0x60df: call    0x18620
0x60e4: test    eax, eax
0x60e6: je      0x72c0
0x60ec: xor     eax, eax

```

```

0x60ee: call    0xc8c0
0x60f3: mov     edx, 0x1c
0x60f8: xor     esi, esi
0x60fa: lea     rdi, [rip + 0x1a631]
0x6101: call    0x8070
0x6106: jmp     0x5723
0x5723
0x610b: lea     r12, [rsp + 0x58]
0x6110: mov     rcx, qword ptr [r13 + 0x28]
0x6114: xor     edx, edx
0x6116: lea     rdi, [rip + 0x1a63b]
0x611d: mov     rsi, r12
0x6120: call    0x83e0
0x6125: test    rax, rax
0x6128: je      0x5730
0x612e: mov     rcx, qword ptr [rsp + 0x58]
0x6133: cmp     byte ptr [rcx], 0
0x6136: je      0x5730
0x613c: mov     rdx, qword ptr [rip + 0x24765]
0x6143: mov     r8d, 1
0x6149: mov     esi, 0x26
0x614e: jmp     0x5b9b
0x5b9b
0x6153: xor     ecx, ecx
0x6155: xor     edx, edx
0x6157: xor     esi, esi
0x6159: xor     r13d, r13d
0x615c: lea     rdi, [rip + 0x1a603]
0x6163: lea     rbp, [rsp + 0x60]
0x6168: call    0x86c0
0x616d: mov     rdx, qword ptr [rip + 0x24734]
0x6174: mov     rdi, qword ptr [rip + 0x24735]
0x617b: xor     ecx, ecx
0x617d: mov     r8d, 1
0x6183: mov     esi, 0x27
0x6188: lea     r12, [rsp + 0x58]
0x618d: call    0x13e30
0x6192: mov     dword ptr [rsp + 0xc], 0
0x619a: xor     edi, edi
0x619c: mov     rcx, rbp
0x619f: mov     edx, 1
0x61a4: mov     rsi, r12
0x61a7: call    0x86c0
0x61ac: test    rax, rax
0x61af: je      0x6c72
0x61b5: mov     rdi, qword ptr [rsp + 0x58]
0x61ba: cmp     byte ptr [rdi], 0
0x61bd: jne     0x61d1

```

```

0x61bf: mov     rax, qword ptr [rsp]
0x61c3: test    rax, rax
0x61c6: je      0x61d1
0x61c8: cmp     byte ptr [rax], 0
0x61cb: jne     0x6b0b
0x61d1: cmp     dword ptr [rsp + 0x60], 2
0x61d6: je      0x6bf1
0x61dc: mov     qword ptr [rsp + 0x68], rdi
0x61e1: mov     rdx, qword ptr [rip + 0x246c0]
0x61e8: mov     r8d, 1
0x61ee: mov     rcx, rbp
0x61f1: mov     rdi, qword ptr [rip + 0x246b8]
0x61f8: mov     esi, 0x28
0x61fd: call    0x13e30
0x6202: cmp     dword ptr [rsp + 0x60], -1
0x6207: je      0x6d8e
0x620d: test    r13d, r13d
0x6210: jne     0x6b64
0x6216: mov     rdi, qword ptr [rsp + 0x58]
0x621b: call    0x49b0
0x6220: cmp     rax, qword ptr [rsp + 0x70]
0x6225: jbe     0x619a
0x622b: xor     eax, eax
0x622d: call    0xce70
0x6232: xor     eax, eax
0x6234: call    0xd870
0x6239: xor     eax, eax
0x623b: call    0xd740
0x6240: xor     eax, eax
0x6242: call    0xa820
0x6247: jmp     0x619a
0x619a:
0x624c: xor     ecx, ecx
0x624e: xor     edx, edx
0x6250: xor     esi, esi
0x6252: xor     r13d, r13d
0x6255: lea     rdi, [rip + 0x1a51e]
0x625c: lea     rbp, [rsp + 0x60]
0x6261: call    0x86c0
0x6266: mov     rdx, qword ptr [rip + 0x2463b]
0x626d: mov     rdi, qword ptr [rip + 0x2463c]
0x6274: xor     ecx, ecx
0x6276: mov     r8d, 1
0x627c: mov     esi, 0x27
0x6281: lea     r12, [rsp + 0x58]
0x6286: call    0x13e30
0x628b: mov     dword ptr [rsp + 0xc], 0
0x6293: xor     edi, edi

```



```

0x6295: mov     rcx, rbp
0x6298: mov     edx, 2
0x629d: mov     rsi, r12
0x62a0: call    0x86c0
0x62a5: test    rax, rax
0x62a8: je      0x6c72
0x62ae: mov     rdi, qword ptr [rsp + 0x58]
0x62b3: cmp     byte ptr [rdi], 0
0x62b6: jne     0x62ca
0x62b8: mov     rax, qword ptr [rsp]
0x62bc: test    rax, rax
0x62bf: je      0x62ca
0x62c1: cmp     byte ptr [rax], 0
0x62c4: jne     0x6a74
0x62ca: cmp     dword ptr [rsp + 0x60], 2
0x62cf: je      0x6bbc
0x62d5: mov     qword ptr [rsp + 0x68], rdi
0x62da: mov     rdx, qword ptr [rip + 0x245c7]
0x62e1: mov     r8d, 1
0x62e7: mov     rcx, rbp
0x62ea: mov     rdi, qword ptr [rip + 0x245bf]
0x62f1: mov     esi, 0x29
0x62f6: call    0x13e30
0x62fb: cmp     dword ptr [rsp + 0x60], -1
0x6300: je      0x6da8
0x6306: test    r13d, r13d
0x6309: jne     0x6acd
0x630f: mov     rdi, qword ptr [rsp + 0x58]
0x6314: call    0x49b0
0x6319: cmp     rax, qword ptr [rsp + 0x70]
0x631e: jbe     0x6293
0x6324: xor     eax, eax
0x6326: call    0xce70
0x632b: xor     eax, eax
0x632d: call    0xd870
0x6332: xor     eax, eax
0x6334: call    0xd740
0x6339: xor     eax, eax
0x633b: call    0xa820
0x6340: jmp     0x6293
0x6293:
0x6345: mov     rsi, qword ptr [rip + 0x2455c]
0x634c: mov     rdi, qword ptr [rip + 0x2455d]
0x6353: xor     eax, eax
0x6355: call    0x8a80
0x635a: jmp     0x5afc
0x5afc:
0x635f: mov     rsi, qword ptr [rip + 0x24542]

```

```

0x6366: mov     rdi, qword ptr [rip + 0x24543]
0x636d: xor     eax, eax
0x636f: call    0x8a30
0x6374: jmp     0x5afc
0x5afc
0x6379: mov     rdi, qword ptr [rip + 0x24530]
0x6380: call    0xf490
0x6385: mov     rdi, rax
0x6388: mov     r12, rax
0x638b: call    0x49b0
0x6390: lea     rdi, [rax + 5]
0x6394: call    0x1d670
0x6399: mov     rbp, rax
0x639c: jmp     0x5ed0
0x5ed0
0x63a1: mov     rdx, qword ptr [rip + 0x24500]
0x63a8: mov     rdi, qword ptr [rip + 0x24501]
0x63af: xor     ecx, ecx
0x63b1: mov     esi, 0x1f
0x63b6: mov     r8d, 1
0x63bc: call    0x13e30
0x63c1: jmp     0x5730
0x5730
0x63c6: mov     rdx, qword ptr [rip + 0x244db]
0x63cd: mov     rdi, qword ptr [rip + 0x244dc]
0x63d4: xor     ecx, ecx
0x63d6: mov     esi, 0x20
0x63db: mov     r8d, 1
0x63e1: call    0x13e30
0x63e6: jmp     0x5730
0x5730
0x63eb: mov     rdx, qword ptr [rip + 0x244b6]
0x63f2: mov     rdi, qword ptr [rip + 0x244b7]
0x63f9: xor     ecx, ecx
0x63fb: mov     esi, 0x19
0x6400: mov     r8d, 1
0x6406: call    0x13e30
0x640b: jmp     0x5730
0x5730
0x6410: mov     rdx, qword ptr [rip + 0x24491]
0x6417: mov     rdi, qword ptr [rip + 0x24492]
0x641e: xor     ecx, ecx
0x6420: mov     esi, 3
0x6425: mov     r8d, 1
0x642b: call    0x13e30
0x6430: jmp     0x5730
0x5730
0x6435: cmp     dword ptr [rip + 0x244cc], 0

```

```

0x643c: jne      0x701d
0x6442: mov      r15d, dword ptr [rsp + 0x10]
0x6447: mov      rdi, qword ptr [rip + 0x24462]
0x644e: call     0x11100
0x6453: mov      rdi, qword ptr [rip + 0x24466]
0x645a: mov      rbp, rax
0x645d: call     0xf3e0
0x6462: mov      rdi, qword ptr [rip + 0x2444f]
0x6469: call     0xf3e0
0x646e: xor      edi, edi
0x6470: call     0x9b00
0x6475: xor      eax, eax
0x6477: call     0xa6c0
0x647c: cmp      qword ptr [rip + 0x24f84], 0
0x6484: je       0x648d
0x6486: xor      eax, eax
0x6488: call     0xc820
0x648d: xor      eax, eax
0x648f: call     0xde40
0x6494: xor      eax, eax
0x6496: call     0x19580
0x649b: mov      rdi, qword ptr [rip + 0x24426]
0x64a2: call     0xbbc0
0x64a7: test     r15d, r15d
0x64aa: je       0x507e
0x64b0: mov      rdi, rbp
0x64b3: xor      r15d, r15d
0x64b6: call     0x49b0
0x64bb: mov      rsi, rbp
0x64be: mov      edi, 3
0x64c3: mov      rdx, rax
0x64c6: call     0x4790
0x64cb: jmp      0x507e
0x507e
0x64d0: mov      rdi, r14
0x64d3: xor      ebp, ebp
0x64d5: call     0xc7e0
0x64da: jmp      0x64f0
0x64f0
0x64dc: mov      edi, 1
0x64e1: add      ebp, 1
0x64e4: call     0xdbb0
0x64e9: xor      eax, eax
0x64eb: call     0xa820
0x64f0: cmp      dword ptr [rsp + 0x28], ebp
0x64f4: jg       0x64dc
0x64f6: jmp      0x5723
0x5723

```

```

0x64fb: test    byte ptr [rsp + 0x28], 1
0x6500: je      0x5730
0x6506: xor     edi, edi
0x6508: lea     rbp, [rip + 0x19f49]
0x650f: lea     r13, [rip + 0x22b0a]
0x6516: call    0x4aa0
0x651b: mov     rdi, qword ptr [rip + 0x243a6]
0x6522: call    0xbb40
0x6527: mov     rdi, qword ptr [rip + 0x2373a]
0x652e: xor     edx, edx
0x6530: xor     esi, esi
0x6532: call    0xe480
0x6537: xor     eax, eax
0x6539: call    0xa820
0x653e: mov     rdi, r14
0x6541: call    0xc970
0x6546: xor     eax, eax
0x6548: call    0xc8c0
0x654d: mov     r14, rax
0x6550: xor     eax, eax
0x6552: call    0xd870
0x6557: xor     eax, eax
0x6559: call    0xd740
0x655e: xor     eax, eax
0x6560: call    0xa820
0x6565: mov     rdi, r15
0x6568: call    0xdb10
0x656d: mov     dword ptr [rip + 0x2436d], 1
0x6577: jmp     0x658e
0x658e:
0x6579: nop     dword ptr [rax]
0x6580: mov     rdi, qword ptr [rip + 0x236e1]
0x6587: xor     eax, eax
0x6589: call    0x7d10
0x658e: mov     rdi, rbx
0x6591: call    0xb180
0x6596: test    rax, rax
0x6599: je      0x6580
0x659b: mov     rdx, qword ptr [rax]
0x659e: mov     rax, qword ptr [rax + 0x10]
0x65a2: movzx   r12d, byte ptr [rdx]
0x65a6: test    rax, rax
0x65a9: je      0x696f
0x65af: cmp     byte ptr [rax + 0x1b], 0
0x65b3: je      0x696f
0x65b9: mov     rax, qword ptr [rax]
0x65bc: mov     esi, 0x23
0x65c1: sub     rax, r13

```

```

0x65c4: cqp
0x65c6: idiv    rsi
0x65c9: mov     r12d, eax
0x65cc: not     r12d
0x65cf: cmp     eax, 6
0x65d2: jne     0x6e1f
0x65d8: test    byte ptr [rsp + 0x28], 1
0x65dd: jne     0x7383
0x65e3: mov     rdi, r14
0x65e6: call    0xc7e0
0x65eb: jmp     0x65fa
0x65fa
0x65ed: mov     esi, 3
0x65f2: mov     edi, r12d
0x65f5: call    0x7570
0x65fa: mov     eax, dword ptr [rsp + 0x28]
0x65fe: lea     edx, [rax - 1]
0x6601: mov     dword ptr [rsp + 0x28], edx
0x6605: test    eax, eax
0x6607: jne     0x65ed
0x6609: call    0xc8c0
0x660e: mov     r14, rax
0x6611: xor     eax, eax
0x6613: call    0xde0
0x6618: jmp     0x653e
0x653e
0x661d: xor     eax, eax
0x661f: call    0x96c0
0x6624: mov     rdi, qword ptr [rip + 0x24285]
0x662b: call    0x11100
0x6630: mov     rdi, rax
0x6633: call    0xc4b0
0x6638: xor     edi, edi
0x663a: xor     eax, eax
0x663c: call    0x7bd0
0x6641: jmp     0x5730
0x5730
0x6646: test    byte ptr [rsp + 0x28], 1
0x664b: je      0x5730
0x6651: cmp     dword ptr [rip + 0x23600], 0
0x6658: je      0x5730
0x665e: mov     rdi, qword ptr [rip + 0x2424b]
0x6665: call    0xf480
0x666a: mov     rdi, qword ptr [rip + 0x24237]
0x6671: call    0xf480
0x6676: mov     rdx, qword ptr [rip + 0x2422b]
0x667d: mov     rdi, qword ptr [rip + 0x2422c]
0x6684: xor     ecx, ecx

```

```

0x6686: mov     r8d, 1
0x668c: mov     esi, 0x23
0x6691: jmp     0x59ff
0x59ff
0x6696: mov     rdi, r14
0x6699: xor     ebp, ebp
0x669b: call    0xc7e0
0x66a0: jmp     0x66b6
0x66b6
0x66a2: mov     edi, 2
0x66a7: add     ebp, 1
0x66aa: call    0xdbb0
0x66af: xor     eax, eax
0x66b1: call    0xa820
0x66b6: cmp     dword ptr [rsp + 0x28], ebp
0x66ba: jg      0x66a2
0x66bc: jmp     0x5723
0x5723
0x66c1: mov     rdi, r14
0x66c4: call    0xc7e0
0x66c9: xor     edi, edi
0x66cb: call    0xd1a0
0x66d0: xor     eax, eax
0x66d2: call    0xc8c0
0x66d7: mov     r14, rax
0x66da: jmp     0x5730
0x5730
0x66df: mov     rdx, qword ptr [rip + 0x241c2]
0x66e6: mov     rdi, qword ptr [rip + 0x241c3]
0x66ed: xor     ecx, ecx
0x66ef: mov     esi, 2
0x66f4: mov     r8d, 1
0x66fa: call    0x13e30
0x66ff: jmp     0x5730
0x5730
0x6704: mov     rdx, qword ptr [rip + 0x2419d]
0x670b: mov     rdi, qword ptr [rip + 0x2419e]
0x6712: xor     ecx, ecx
0x6714: mov     esi, 0x1a
0x6719: mov     r8d, 1
0x671f: call    0x13e30
0x6724: jmp     0x5730
0x5730
0x6729: xor     ebp, ebp
0x672b: jmp     0x6750
0x6750
0x672d: mov     r8d, 1
0x6733: xor     ecx, ecx

```

```

0x6735: mov     esi, 4
0x673a: add     ebp, 1
0x673d: mov     rdx, qword ptr [rip + 0x24164]
0x6744: mov     rdi, qword ptr [rip + 0x24165]
0x674b: call    0x13e30
0x6750: cmp     dword ptr [rsp + 0x28], ebp
0x6754: jg      0x672d
0x6756: jmp     0x5730
0x5730
0x675b: xor     ebp, ebp
0x675d: jmp     0x6782
0x6782
0x675f: mov     r8d, 1
0x6765: xor     ecx, ecx
0x6767: mov     esi, 0x1d
0x676c: add     ebp, 1
0x676f: mov     rdx, qword ptr [rip + 0x24132]
0x6776: mov     rdi, qword ptr [rip + 0x24133]
0x677d: call    0x13e30
0x6782: cmp     dword ptr [rsp + 0x28], ebp
0x6786: jg      0x675f
0x6788: jmp     0x5730
0x5730
0x678d: xor     ebp, ebp
0x678f: jmp     0x67b4
0x67b4
0x6791: mov     r8d, 1
0x6797: xor     ecx, ecx
0x6799: mov     esi, 0x1e
0x679e: add     ebp, 1
0x67a1: mov     rdx, qword ptr [rip + 0x24100]
0x67a8: mov     rdi, qword ptr [rip + 0x24101]
0x67af: call    0x13e30
0x67b4: cmp     dword ptr [rsp + 0x28], ebp
0x67b8: jg      0x6791
0x67ba: jmp     0x5730
0x5730
0x67bf: nop
0x67c0: test    rax, rax
0x67c3: je      0x57dc
0x67c9: mov     rdi, qword ptr [rip + 0x240e0]
0x67d0: call    0xf480
0x67d5: mov     rdi, qword ptr [rip + 0x240cc]
0x67dc: call    0xf480
0x67e1: cmp     qword ptr [r13 + 8], 0
0x67e6: je      0x70b7
0x67ec: lea     rbp, [rsp + 0x60]
0x67f1: mov     rcx, qword ptr [rip + 0x240b0]

```

```

0x67f8: xor     eax, eax
0x67fa: mov     rdi, r13
0x67fd: mov     rdx, qword ptr [rip + 0x240ac]
0x6804: mov     rsi, rbp
0x6807: mov     qword ptr [rsp + 0x60], 0
0x6810: call    0x8b50
0x6815: test    eax, eax
0x6817: je      0x6c1f
0x681d: jle     0x5730
0x6823: mov     rdi, qword ptr [r13]
0x6827: mov     dword ptr [rsp + 0xc], eax
0x682b: call    0x49b0
0x6830: mov     rbp, qword ptr [rsp + 0x60]
0x6835: mov     r12, rax
0x6838: mov     rdi, rbp
0x683b: call    0x49b0
0x6840: lea     rdi, [r12 + rax + 0x21]
0x6845: call    0x1d670
0x684a: mov     r8, qword ptr [r13]
0x684e: mov     r9, rbp
0x6851: or      rdx, 0xffffffffffffffff
0x6855: lea     rcx, [rip + 0x19cd9]
0x685c: mov     r12, rax
0x685f: mov     rdi, rax
0x6862: mov     esi, 1
0x6867: xor     eax, eax
0x6869: call    0x4f10
0x686e: mov     edx, 1
0x6873: mov     esi, 1
0x6878: mov     rdi, r12
0x687b: call    0xe480
0x6880: xor     eax, eax
0x6882: call    0xa820
0x6887: mov     rdi, r12
0x688a: call    0x18d10
0x688f: cmp     byte ptr [r13 + 0x1a], 0
0x6894: jne     0x6dc9
0x689a: mov     rdi, rbp
0x689d: call    0x185d0
0x68a2: test    eax, eax
0x68a4: je      0x6dbc
0x68aa: movsx   esi, byte ptr [r13 + 0x1a]
0x68af: mov     rdi, rbp
0x68b2: call    0x193f0
0x68b7: mov     rdi, rbp
0x68ba: mov     r12d, eax
0x68bd: call    0x18d10
0x68c2: cmp     byte ptr [r13 + 0x1a], 0

```



```

0x68c7: je      0x6ccf
0x68cd: mov     eax, r12d
0x68d0: and      eax, 0x7f
0x68d3: add      eax, 1
0x68d6: sub      al, 1
0x68d8: jg       0x6f1d
0x68de: test     r12b, 0x7f
0x68e2: jne      0x6955
0x68e4: and      r12d, 0xff00
0x68eb: jne      0x719d
0x68f1: mov      rdi, qword ptr [r13 + 0x10]
0x68f5: test     rdi, rdi
0x68f8: je       0x6943
0x68fa: xor      eax, eax
0x68fc: call     0x19940
0x6901: mov      r8d, 1
0x6907: mov      esi, 0x21
0x690c: mov      rdx, qword ptr [rip + 0x23f95]
0x6913: mov      rdi, qword ptr [rip + 0x23f96]
0x691a: mov      rcx, rax
0x691d: mov      r12, rax
0x6920: call     0x13e30
0x6925: mov      rdi, qword ptr [rip + 0x23f84]
0x692c: call     0x11100
0x6931: mov      rdi, rax
0x6934: xor      eax, eax
0x6936: call     0x89d0
0x693b: mov      rdi, r12
0x693e: call     0x18d10
0x6943: cmp      dword ptr [rsp + 0xc], 2
0x6948: je       0x71b4
0x694e: cmp      dword ptr [rsp + 0xc], 3
0x6953: je       0x695c
0x6955: xor      ebp, ebp
0x6957: jmp      0x565a
0x565a
0x695c: mov      rdi, qword ptr [rip + 0x23f45]
0x6963: xor      ebp, ebp
0x6965: call     0x110b0
0x696a: jmp      0x565a
0x565a
0x696f: lea      eax, [r12 + 0x22]
0x6974: cmp      eax, 0x29
0x6977: ja       0x65e3
0x697d: movsxd   rax, dword ptr [rbp + rax*4]
0x6982: add      rax, rbp
0x6985: jmp      rax
rax

```

```

                                ->->->                                indirect JMP
0xa1a0: endbr64
0xa1a4: mov     eax, dword ptr [rip + 0x211ae]
0xa1aa: mov     r8, rdi
0xa1ad: mov     edi, dword ptr [rip + 0x207e5]
0xa1b3: push    r13
0xa1b5: push    r12
0xa1b7: xor     r12d, r12d
0xa1ba: lea     edx, [rax + rsi]
0xa1bd: push    rbx
0xa1be: mov     dword ptr [rip + 0x21194], edx
0xa1c4: cmp     edi, eax
0xa1c6: jle     0xa21c
0xa1c8: mov     ecx, dword ptr [rip + 0x21186]
0xa1ce: cmp     ecx, dword ptr [rip + 0x207c8]
0xa1d4: jge     0xa21c
0xa1d6: mov     r12d, edi
0xa1d9: sub     r12d, eax
0xa1dc: cmp     edx, edi
0xa1de: cmovle r12d, esi
0xa1e2: imul    edi, ecx
0xa1e5: mov     rsi, r8
0xa1e8: movsxd  r13, r12d
0xa1eb: lea     ebx, [rdi + rax]
0xa1ee: mov     rdi, qword ptr [rip + 0x21143]
0xa1f5: mov     rdx, r13
0xa1f8: movsxd  rbx, ebx
0xa1fb: add     rdi, rbx
0xa1fe: call    0x4bb0
0xa203: add     rbx, qword ptr [rip + 0x21126]
0xa20a: movzx   esi, byte ptr [rip + 0x21140]
0xa211: mov     rdx, r13
0xa214: mov     rdi, rbx
0xa217: call    0x4a50
0xa21c: mov     eax, r12d
0xa21f: pop     rbx
0xa220: pop     r12
0xa222: pop     r13
0xa224: ret
0xa225: nop     word ptr cs:[rax + rax]
0xa230: endbr64
0xa234: push    r13
0xa236: push    r12
0xa238: xor     r12d, r12d
0xa23b: push    rbx
0xa23c: sub     rsp, 0x10
0xa240: mov     rax, qword ptr fs:[0x28]
0xa249: mov     qword ptr [rsp + 8], rax

```

```

0xa24e: xor     eax, eax
0xa250: mov     eax, dword ptr [rip + 0x21102]
0xa256: mov     byte ptr [rsp + 7], dil
0xa25b: mov     edi, dword ptr [rip + 0x20737]
0xa261: lea     edx, [rax + 1]
0xa264: mov     dword ptr [rip + 0x210ee], edx
0xa26a: cmp     eax, edi
0xa26c: jge     0xa2c2
0xa26e: mov     ecx, dword ptr [rip + 0x210e0]
0xa274: cmp     ecx, dword ptr [rip + 0x20722]
0xa27a: jge     0xa2c2
0xa27c: cmp     edx, edi
0xa27e: jle     0xa2e0
0xa280: mov     r12d, edi
0xa283: sub     r12d, eax
0xa286: movsxd  r13, r12d
0xa289: imul    edi, ecx
0xa28c: lea     rsi, [rsp + 7]
0xa291: mov     rdx, r13
0xa294: lea     ebx, [rdi + rax]
0xa297: mov     rdi, qword ptr [rip + 0x2109a]
0xa29e: movsxd  rbx, ebx
0xa2a1: add     rdi, rbx
0xa2a4: call    0x4bb0
0xa2a9: add     rbx, qword ptr [rip + 0x21080]
0xa2b0: movzx   esi, byte ptr [rip + 0x2109a]
0xa2b7: mov     rdx, r13
0xa2ba: mov     rdi, rbx
0xa2bd: call    0x4a50
0xa2c2: mov     rax, qword ptr [rsp + 8]
0xa2c7: sub     rax, qword ptr fs:[0x28]
0xa2d0: jne     0xa2ee
0xa2d2: add     rsp, 0x10
0xa2d6: mov     eax, r12d
0xa2d9: pop     rbx
0xa2da: pop     r12
0xa2dc: pop     r13
0xa2de: ret
0xa2df: nop
0xa2e0: mov     r13d, 1
0xa2e6: mov     r12d, 1
0xa2ec: jmp     0xa289
0xa289
0xa2ee: call    0x49d0
0xa2f3: nop     word ptr cs:[rax + rax]
0xa2fe: nop
0xa300: endbr64
0xa304: push    rbp

```

```

0xa305: lea     rdx, [rip - 0x89c]
0xa30c: push    rbx
0xa30d: sub     rsp, 8
0xa311: mov     esi, dword ptr [rip + 0x20685]
0xa317: mov     rdi, qword ptr [rip + 0x1faca]
0xa31e: call    0x4820
0xa323: mov     edx, dword ptr [rip + 0x20bdf]
0xa329: test    edx, edx
0xa32b: jle     0xa365
0xa32d: xor     ebp, ebp
0xa32f: lea     rbx, [rip + 0x20bea]
0xa336: jmp     0xa34c
0xa34c:
0xa338: nop     dword ptr [rax + rax]
0xa340: mov     edx, dword ptr [rip + 0x20bc2]
0xa346: add     ebp, eax
0xa348: cmp     edx, ebp
0xa34a: jle     0xa365
0xa34c: sub     edx, ebp
0xa34e: movsxd  rsi, ebp
0xa351: mov     edi, 1
0xa356: add     rsi, rbx
0xa359: movsxd  rdx, edx
0xa35c: call    0x18da0
0xa361: test    eax, eax
0xa363: jns     0xa340
0xa365: mov     edx, dword ptr [rip + 0x20631]
0xa36b: xor     esi, esi
0xa36d: imul    edx, dword ptr [rip + 0x20624]
0xa374: mov     dword ptr [rip + 0x20b8a], 0
0xa37e: mov     rdi, qword ptr [rip + 0x20fb3]
0xa385: movsxd  rdx, edx
0xa388: call    0x4a50
0xa38d: mov     edx, dword ptr [rip + 0x20609]
0xa393: imul    edx, dword ptr [rip + 0x205fe]
0xa39a: xor     esi, esi
0xa39c: mov     rdi, qword ptr [rip + 0x20f8d]
0xa3a3: movsxd  rdx, edx
0xa3a6: call    0x4a50
0xa3ab: mov     edx, dword ptr [rip + 0x205eb]
0xa3b1: imul    edx, dword ptr [rip + 0x205e0]
0xa3b8: xor     esi, esi
0xa3ba: mov     rdi, qword ptr [rip + 0x20f67]
0xa3c1: movsxd  rdx, edx
0xa3c4: call    0x4a50
0xa3c9: mov     edx, dword ptr [rip + 0x205cd]
0xa3cf: imul    edx, dword ptr [rip + 0x205c2]
0xa3d6: xor     esi, esi

```

```

0xa3d8: mov     rdi, qword ptr [rip + 0x20f41]
0xa3df: movsxd  rdx, edx
0xa3e2: call    0x4a50
0xa3e7: mov     dword ptr [rip + 0x20f67], 0
0xa3f1: mov     dword ptr [rip + 0x20f59], 0
0xa3fb: add     rsp, 8
0xa3ff: pop     rbx
0xa400: pop     rbp
0xa401: ret
0xa402: nop     word ptr cs:[rax + rax]
0xa40d: nop     dword ptr [rax]
0xa410: endbr64
0xa414: sub     rsp, 8
0xa418: mov     edx, dword ptr [rip + 0x2057e]
0xa41e: imul    edx, dword ptr [rip + 0x20573]
0xa425: mov     esi, 0x20
0xa42a: mov     rdi, qword ptr [rip + 0x20f07]
0xa431: movsxd  rdx, edx
0xa434: call    0x4a50
0xa439: mov     edx, dword ptr [rip + 0x2055d]
0xa43f: imul    edx, dword ptr [rip + 0x20552]
0xa446: movzx   esi, byte ptr [rip + 0x20f04]
0xa44d: mov     rdi, qword ptr [rip + 0x20edc]
0xa454: movsxd  rdx, edx
0xa457: call    0x4a50
0xa45c: mov     edx, dword ptr [rip + 0x2053a]
0xa462: imul    edx, dword ptr [rip + 0x2052f]
0xa469: xor     esi, esi
0xa46b: mov     rdi, qword ptr [rip + 0x20eb6]
0xa472: add     rsp, 8
0xa476: movsxd  rdx, edx
0xa479: jmp     0x4a50
0x4a50
0xa47e: nop
0xa480: endbr64
0xa484: mov     edx, dword ptr [rip + 0x20512]
0xa48a: imul    edx, dword ptr [rip + 0x20507]
0xa491: xor     esi, esi
0xa493: mov     rdi, qword ptr [rip + 0x20e8e]
0xa49a: movsxd  rdx, edx
0xa49d: jmp     0x4a50
0x4a50
0xa4a2: nop     word ptr cs:[rax + rax]
0xa4ad: nop     dword ptr [rax]
0xa4b0: endbr64
0xa4b4: mov     dword ptr [rip + 0x20e9a], edi
0xa4ba: mov     dword ptr [rip + 0x20e98], esi
0xa4c0: ret

```

```

0xa4c1: nop      word ptr cs:[rax + rax]
0xa4cc: nop      dword ptr [rax]
0xa4d0: endbr64
0xa4d4: mov      eax, dword ptr [rip + 0x20e7a]
0xa4da: mov      dword ptr [rdi], eax
0xa4dc: mov      eax, dword ptr [rip + 0x20e76]
0xa4e2: mov      dword ptr [rsi], eax
0xa4e4: ret
0xa4e5: nop      word ptr cs:[rax + rax]
0xa4f0: endbr64
0xa4f4: movzx    eax, byte ptr [rip + 0x20e56]
0xa4fb: and      edi, 7
0xa4fe: and      eax, 0xffffffff8
0xa501: or       eax, edi
0xa503: mov      byte ptr [rip + 0x20e48], al
0xa509: ret
0xa50a: nop      word ptr [rax + rax]
0xa510: endbr64
0xa514: movzx    eax, byte ptr [rip + 0x20e36]
0xa51b: shl      edi, 3
0xa51e: and      edi, 0x38
0xa521: and      eax, 0xffffffffc7
0xa524: or       eax, edi
0xa526: mov      byte ptr [rip + 0x20e25], al
0xa52c: ret
0xa52d: nop      dword ptr [rax]
0xa530: endbr64
0xa534: movzx    eax, byte ptr [rip + 0x20e16]
0xa53b: shl      edi, 6
0xa53e: and      edi, 0x40
0xa541: and      eax, 0xffffffffbf
0xa544: or       eax, edi
0xa546: mov      byte ptr [rip + 0x20e05], al
0xa54c: ret
0xa54d: nop      dword ptr [rax]
0xa550: endbr64
0xa554: movzx    eax, byte ptr [rip + 0x20df6]
0xa55b: shl      edi, 7
0xa55e: and      eax, 0x7f
0xa561: or       eax, edi
0xa563: mov      byte ptr [rip + 0x20de8], al
0xa569: ret
0xa56a: nop      word ptr [rax + rax]
0xa570: endbr64
0xa574: shl      edi, 6
0xa577: and      esi, 7
0xa57a: shl      edx, 3
0xa57d: mov      eax, edi

```

```

0xa57f: movzx    edi, byte ptr [rip + 0x20dcb]
0xa586: and      edx, 0x38
0xa589: and      eax, 0x40
0xa58c: and      edi, 0xffffffffb8
0xa58f: or       edi, eax
0xa591: or       edi, esi
0xa593: and      edi, 0xffffffffc7
0xa596: or       edi, edx
0xa598: mov      byte ptr [rip + 0x20db2], dil
0xa59f: ret
0xa5a0: endbr64
0xa5a4: push     rbp
0xa5a5: push     rbx
0xa5a6: sub      rsp, 8
0xa5aa: movsxd   rax, dword ptr [rip + 0x20957]
0xa5b1: cmp      eax, 0x400
0xa5b6: je       0xa610
0xa5b8: lea      edx, [rax + 1]
0xa5bb: lea      rbx, [rip + 0x2095e]
0xa5c2: mov      dword ptr [rip + 0x20940], edx
0xa5c8: mov      byte ptr [rbx + rax], 7
0xa5cc: test     edx, edx
0xa5ce: jle      0xa5fd
0xa5d0: xor      ebp, ebp
0xa5d2: jmp      0xa5e4
0xa5e4:
0xa5d4: nop      dword ptr [rax]
0xa5d8: mov      edx, dword ptr [rip + 0x2092a]
0xa5de: add      ebp, eax
0xa5e0: cmp      edx, ebp
0xa5e2: jle      0xa5fd
0xa5e4: sub      edx, ebp
0xa5e6: movsxd   rsi, ebp
0xa5e9: mov      edi, 1
0xa5ee: add      rsi, rbx
0xa5f1: movsxd   rdx, edx
0xa5f4: call     0x18da0
0xa5f9: test     eax, eax
0xa5fb: jns      0xa5d8
0xa5fd: mov      dword ptr [rip + 0x20901], 0
0xa607: add      rsp, 8
0xa60b: pop      rbx
0xa60c: pop      rbp
0xa60d: ret
0xa60e: nop
0xa610: xor      ebp, ebp
0xa612: lea      rbx, [rip + 0x20907]
0xa619: jmp      0xa62c

```

```

0xa62c
0xa61b: nop      dword ptr [rax + rax]
0xa620: add      ebp, eax
0xa622: mov      eax, dword ptr [rip + 0x208e0]
0xa628: cmp      eax, ebp
0xa62a: jle      0xa645
0xa62c: sub      eax, ebp
0xa62e: movsxd   rsi, ebp
0xa631: mov      edi, 1
0xa636: add      rsi, rbx
0xa639: movsxd   rdx, eax
0xa63c: call     0x18da0
0xa641: test     eax, eax
0xa643: jns      0xa620
0xa645: mov      dword ptr [rip + 0x208b9], 1
0xa64f: mov      edx, 1
0xa654: mov      byte ptr [rip + 0x208c5], 7
0xa65b: jmp      0xa5d0
0xa5d0
0xa660: endbr64
0xa664: test     edi, edi
0xa666: je       0xa688
0xa668: mov      rdi, qword ptr [rip + 0x1f759]
0xa66f: test     rdi, rdi
0xa672: je       0xa694
0xa674: lea      rdx, [rip - 0xc0b]
0xa67b: mov      esi, 1
0xa680: jmp      0x4820
0x4820
0xa685: nop      dword ptr [rax]
0xa688: mov      rdi, qword ptr [rip + 0x1f719]
0xa68f: test     rdi, rdi
0xa692: jne      0xa674
0xa694: ret
0xa695: nop      word ptr cs:[rax + rax]
0xa6a0: endbr64
0xa6a4: movzx    eax, byte ptr [rip + 0x20ca6]
0xa6ab: mov      byte ptr [rdi], al
0xa6ad: ret
0xa6ae: nop
0xa6b0: endbr64
0xa6b4: movzx    eax, byte ptr [rdi]
0xa6b7: mov      byte ptr [rip + 0x20c94], al
0xa6bd: ret
0xa6be: nop
0xa6c0: endbr64
0xa6c4: push     rbp
0xa6c5: push     rbx

```



```

0xa6c6: sub     rsp, 8
0xa6ca: cmp     dword ptr [rip + 0x1f58f], 1
0xa6d1: je      0xa720
0xa6d3: mov     rdi, qword ptr [rip + 0x1f66e]
0xa6da: test    rdi, rdi
0xa6dd: je      0xa6f0
0xa6df: lea     rdx, [rip - 0xc76]
0xa6e6: mov     esi, 1
0xa6eb: call    0x4820
0xa6f0: mov     dword ptr [rip + 0x20c52], 0
0xa6fa: mov     dword ptr [rip + 0x20c44], 0
0xa704: mov     dword ptr [rip + 0x20c36], 0
0xa70e: mov     dword ptr [rip + 0x20c28], 0
0xa718: add     rsp, 8
0xa71c: pop     rbx
0xa71d: pop     rbp
0xa71e: ret
0xa71f: nop
0xa720: mov     edx, dword ptr [rip + 0x207e2]
0xa726: cmp     edx, 0x3fb
0xa72c: jg      0xa750
0xa72e: movsxd  rax, edx
0xa731: lea     rcx, [rip + 0x207e8]
0xa738: add     edx, 4
0xa73b: add     rax, rcx
0xa73e: mov     ecx, dword ptr [rip + 0x1f5db]
0xa744: mov     dword ptr [rip + 0x207be], edx
0xa74a: mov     dword ptr [rax], ecx
0xa74c: jmp     0xa6d3
0xa6d3:
0xa74e: nop
0xa750: xor     ebx, ebx
0xa752: lea     rbp, [rip + 0x207c7]
0xa759: jmp     0xa76c
0xa76c:
0xa75b: nop     dword ptr [rax + rax]
0xa760: mov     edx, dword ptr [rip + 0x207a2]
0xa766: add     ebx, eax
0xa768: cmp     edx, ebx
0xa76a: jle     0xa785
0xa76c: sub     edx, ebx
0xa76e: movsxd  rsi, ebx
0xa771: mov     edi, 1
0xa776: add     rsi, rbp
0xa779: movsxd  rdx, edx
0xa77c: call    0x18da0
0xa781: test    eax, eax
0xa783: jns     0xa760

```

```

0xa785: mov     edx, 4
0xa78a: lea     rax, [rip + 0x2078f]
0xa791: jmp     0xa73e
0xa73e
0xa793: nop     word ptr cs:[rax + rax]
0xa79e: nop
0xa7a0: push    r12
0xa7a2: mov     r12d, edi
0xa7a5: push    rbp
0xa7a6: mov     ebp, esi
0xa7a8: sub     rsp, 8
0xa7ac: mov     eax, dword ptr [rip + 0x1f71e]
0xa7b2: test    eax, eax
0xa7b4: je      0xa800
0xa7b6: mov     eax, ebp
0xa7b8: mov     esi, dword ptr [rip + 0x201da]
0xa7be: mov     edx, dword ptr [rip + 0x201d8]
0xa7c4: or      eax, r12d
0xa7c7: mov     rdi, qword ptr [rip + 0x1f63a]
0xa7ce: js      0xa7d4
0xa7d0: cmp     ebp, esi
0xa7d2: jnl     0xa810
0xa7d4: sub     edx, 1
0xa7d7: sub     esi, 1
0xa7da: call    0x4960
0xa7df: add     rsp, 8
0xa7e3: mov     esi, 1
0xa7e8: lea     rdx, [rip - 0xd7f]
0xa7ef: pop     rbp
0xa7f0: mov     rdi, rax
0xa7f3: pop     r12
0xa7f5: jmp     0x4820
0x4820
0xa7fa: nop     word ptr [rax + rax]
0xa800: xor     eax, eax
0xa802: call    0xa6c0
0xa807: jmp     0xa7b6
0xa7b6
0xa809: nop     dword ptr [rax]
0xa810: cmp     r12d, edx
0xa813: jge     0xa7d4
0xa815: mov     edx, r12d
0xa818: mov     esi, ebp
0xa81a: jmp     0xa7da
0xa7da
0xa81c: nop     dword ptr [rax]
0xa820: endbr64
0xa824: push    r15

```

```

0xa826: push    r14
0xa828: push    r13
0xa82a: push    r12
0xa82c: push    rbp
0xa82d: push    rbx
0xa82e: sub     rsp, 0x38
0xa832: mov     ebx, dword ptr [rip + 0x20160]
0xa838: imul    ebx, dword ptr [rip + 0x2015d]
0xa83f: mov     rax, qword ptr fs:[0x28]
0xa848: mov     qword ptr [rsp + 0x28], rax
0xa84d: xor     eax, eax
0xa84f: mov     rdi, qword ptr [rip + 0x1f552]
0xa856: cmp     dword ptr [rip + 0x20aff], 1
0xa85d: sbb     ebx, 0
0xa860: test    rdi, rdi
0xa863: je      0xa876
0xa865: lea     rdx, [rip - 0xdfc]
0xa86c: mov     esi, 1
0xa871: call    0x4820
0xa876: test    ebx, ebx
0xa878: jle     0xab70
0xa87e: movsxd  rax, ebx
0xa881: mov     dword ptr [rsp + 4], 0xffffffff
0xa889: xor     ebp, ebp
0xa88b: mov     r13d, 0xffffffff
0xa891: mov     qword ptr [rsp + 8], rax
0xa896: lea     r12, [rip + 0x20683]
0xa89d: jmp     0xa9ce
0xa9ce:
0xa8a2: nop     word ptr [rax + rax]
0xa8a8: cdq
0xa8a9: idiv    dword ptr [rip + 0x200e9]
0xa8af: mov     r14d, edx
0xa8b2: cmp     edx, r13d
0xa8b5: jne     0xaa20
0xa8bb: cmp     eax, dword ptr [rsp + 4]
0xa8bf: jne     0xaa20
0xa8c5: mov     rax, qword ptr [rip + 0x20a64]
0xa8cc: movzx   ebx, byte ptr [rax + rbp]
0xa8d0: mov     edi, ebx
0xa8d2: mov     r14d, ebx
0xa8d5: sar     edi, 6
0xa8d8: and     r14d, 7
0xa8dc: and     edi, 1
0xa8df: call    0x9870
0xa8e4: mov     eax, ebx
0xa8e6: and     eax, 7
0xa8e9: cmp     dword ptr [rip + 0x20a5c], 1

```

```

0xa8f0: je      0xaa40
0xa8f6: cmp     dword ptr [rip + 0x1f363], 1
0xa8fd: je      0xaa5f
0xa903: xor     edi, edi
0xa905: cmp     eax, 7
0xa908: setne   dil
0xa90c: call    0x9790
0xa911: movzx   eax, byte ptr [rip + 0x20a38]
0xa918: mov     dword ptr [rip + 0x20a2a], 1
0xa922: and     eax, 0xffffffff8
0xa925: or      eax, r14d
0xa928: mov     byte ptr [rip + 0x20a22], al
0xa92e: sar     ebx, 3
0xa931: and     ebx, 7
0xa934: cmp     dword ptr [rip + 0x20a0d], 1
0xa93b: je      0xaab0
0xa941: cmp     dword ptr [rip + 0x1f318], 1
0xa948: je      0xaad8
0xa94e: xor     edi, edi
0xa950: test    ebx, ebx
0xa952: setne   dil
0xa956: call    0x9790
0xa95b: mov     edx, dword ptr [rip + 0x205a7]
0xa961: movzx   eax, byte ptr [rip + 0x209e8]
0xa968: shl     ebx, 3
0xa96b: mov     dword ptr [rip + 0x209d3], 1
0xa975: and     eax, 0xffffffffc7
0xa978: or      ebx, eax
0xa97a: mov     byte ptr [rip + 0x209d0], bl
0xa980: mov     rax, qword ptr [rip + 0x209b1]
0xa987: movzx   ebx, byte ptr [rax + rbp]
0xa98b: cmp     edx, 0x400
0xa991: je      0xab30
0xa997: lea     eax, [rdx + 1]
0xa99a: movsxd  rdx, edx
0xa99d: add     r13d, 1
0xa9a1: cmp     dword ptr [rip + 0x1fff0], r13d
0xa9a8: mov     dword ptr [rip + 0x2055a], eax
0xa9ae: mov     byte ptr [r12 + rdx], bl
0xa9b2: jne     0xa9bf
0xa9b4: mov     dword ptr [rsp + 4], 0xffffffff
0xa9bc: xor     r13d, r13d
0xa9bf: add     rbp, 1
0xa9c3: cmp     rbp, qword ptr [rsp + 8]
0xa9c8: je      0xab70
0xa9ce: mov     rdx, qword ptr [rip + 0x20953]
0xa9d5: mov     rcx, qword ptr [rip + 0x2095c]
0xa9dc: mov     eax, ebp

```

```

0xa9de: movzx    ebx, byte ptr [rdx + rbp]
0xa9e2: cmp      byte ptr [rcx + rbp], bl
0xa9e5: jne      0xa8a8
0xa9eb: mov      rdx, qword ptr [rip + 0x2092e]
0xa9f2: mov      rcx, qword ptr [rip + 0x20937]
0xa9f9: movzx    edi, byte ptr [rdx + rbp]
0xa9fd: cmp      byte ptr [rcx + rbp], dil
0xaa01: je       0xa9bf
0xaa03: cdq
0xaa04: idiv     dword ptr [rip + 0x1ff8e]
0xaa0a: mov      r14d, edx
0xaa0d: cmp      edx, r13d
0xaa10: je       0xa8bb
0xaa16: nop      word ptr cs:[rax + rax]
0xaa20: mov      esi, r14d
0xaa23: mov      edi, eax
0xaa25: mov      dword ptr [rsp + 4], eax
0xaa29: mov      r13d, r14d
0xaa2c: call     0xa7a0
0xaa31: jmp      0xa8c5
0xa8c5
0xaa36: nop      word ptr cs:[rax + rax]
0xaa40: movzx    edx, byte ptr [rip + 0x20909]
0xaa47: and      edx, 7
0xaa4a: cmp      eax, edx
0xaa4c: je       0xa92e
0xaa52: cmp      dword ptr [rip + 0x1f207], 1
0xaa59: jne      0xa903
0xaa5f: mov      eax, dword ptr [rip + 0x164b4]
0xaa65: mov      edx, dword ptr [rip + 0x2049d]
0xaa6b: mov      dword ptr [rsp + 0x10], eax
0xaa6f: movzx    eax, byte ptr [rip + 0x164a7]
0xaa76: mov      byte ptr [rsp + 0x14], al
0xaa7a: lea      eax, [r14 + 0x30]
0xaa7e: mov      byte ptr [rsp + 0x13], al
0xaa82: cmp      edx, 0x3fa
0xaa88: jg       0xac18
0xaa8e: movsxd   rax, edx
0xaa91: add      edx, 5
0xaa94: add      rax, r12
0xaa97: mov      ecx, dword ptr [rsp + 0x10]
0xaa9b: mov      dword ptr [rip + 0x20467], edx
0xaaa1: mov      dword ptr [rax], ecx
0xaaa3: movzx    ecx, byte ptr [rsp + 0x14]
0xaaa8: mov      byte ptr [rax + 4], cl
0xaaaab: jmp      0xa911
0xa911
0xaab0: movzx    eax, byte ptr [rip + 0x20899]

```

```

0xaab7: mov     edx, dword ptr [rip + 0x2044b]
0xaabd: sar     eax, 3
0xaac0: and     eax, 7
0xaac3: cmp     ebx, eax
0xaac5: je      0xa980
0xaacb: cmp     dword ptr [rip + 0x1f18e], 1
0xaad2: jne     0xa94e
0xaad8: mov     eax, dword ptr [rip + 0x16436]
0xaade: mov     edx, dword ptr [rip + 0x20424]
0xaae4: mov     dword ptr [rsp + 0x10], eax
0xaae8: movzx   eax, byte ptr [rip + 0x16429]
0xaaef: mov     byte ptr [rsp + 0x14], al
0xaaf3: lea     eax, [rbx + 0x30]
0xaaf6: mov     byte ptr [rsp + 0x13], al
0xaafa: cmp     edx, 0x3fa
0xab00: jg      0xac58
0xab06: movsxd  rax, edx
0xab09: add     edx, 5
0xab0c: add     rax, r12
0xab0f: mov     ecx, dword ptr [rsp + 0x10]
0xab13: mov     dword ptr [rip + 0x203ef], edx
0xab19: mov     dword ptr [rax], ecx
0xab1b: movzx   ecx, byte ptr [rsp + 0x14]
0xab20: mov     byte ptr [rax + 4], cl
0xab23: jmp     0xa961
0xa961
0xab28: nop     dword ptr [rax + rax]
0xab30: xor     r14d, r14d
0xab33: jmp     0xab46
0xab46
0xab35: nop     dword ptr [rax]
0xab38: mov     edx, dword ptr [rip + 0x203ca]
0xab3e: add     r14d, eax
0xab41: cmp     edx, r14d
0xab44: jle     0xab60
0xab46: sub     edx, r14d
0xab49: movsxd  rsi, r14d
0xab4c: mov     edi, 1
0xab51: add     rsi, r12
0xab54: movsxd  rdx, edx
0xab57: call    0x18da0
0xab5c: test    eax, eax
0xab5e: jns     0xab38
0xab60: mov     eax, 1
0xab65: xor     edx, edx
0xab67: jmp     0xa99a
0xa99a
0xab6c: nop     dword ptr [rax]

```

```

0xab70: mov     edi, dword ptr [rip + 0x207de]
0xab76: mov     esi, dword ptr [rip + 0x207dc]
0xab7c: call    0xa7a0
0xab81: mov     rdi, qword ptr [rip + 0x1f240]
0xab88: test    rdi, rdi
0xab8b: je      0xab9e
0xab8d: lea     rdx, [rip - 0x1124]
0xab94: mov     esi, 1
0xab99: call    0x4820
0xab9e: mov     edx, dword ptr [rip + 0x20364]
0xaba4: test    edx, edx
0xaba6: jne     0xac98
0xabac: mov     edx, dword ptr [rip + 0x1fde6]
0xabab: imul    edx, dword ptr [rip + 0x1fde3]
0xabab: mov     rsi, qword ptr [rip + 0x20778]
0xabc0: mov     rdi, qword ptr [rip + 0x20761]
0xabc7: movsxd  rdx, edx
0xabca: call    0x4bb0
0xabcf: mov     edx, dword ptr [rip + 0x1fdc3]
0xabd5: imul    edx, dword ptr [rip + 0x1fdc0]
0xabdc: mov     rdi, qword ptr [rip + 0x2073d]
0xabe3: mov     rsi, qword ptr [rip + 0x20746]
0xabea: movsxd  rdx, edx
0xabed: mov     rax, qword ptr [rsp + 0x28]
0xabf2: sub     rax, qword ptr fs:[0x28]
0xabfb: jne     0xace4
0xac01: add     rsp, 0x38
0xac05: pop     rbx
0xac06: pop     rbp
0xac07: pop     r12
0xac09: pop     r13
0xac0b: pop     r14
0xac0d: pop     r15
0xac0f: jmp     0x4bb0
0x4bb0:
0xac14: nop     dword ptr [rax]
0xac18: xor     r15d, r15d
0xac1b: jmp     0xac2e
0xac2e:
0xac1d: nop     dword ptr [rax]
0xac20: mov     edx, dword ptr [rip + 0x202e2]
0xac26: add     r15d, eax
0xac29: cmp     edx, r15d
0xac2c: jle     0xac48
0xac2e: sub     edx, r15d
0xac31: movsxd  rsi, r15d
0xac34: mov     edi, 1
0xac39: add     rsi, r12

```

```

0xac3c: movsxd  rdx, edx
0xac3f: call     0x18da0
0xac44: test     eax, eax
0xac46: jns      0xac20
0xac48: mov      edx, 5
0xac4d: mov      rax, r12
0xac50: jmp      0xaa97
0xaa97:
0xac55: nop      dword ptr [rax]
0xac58: xor      r14d, r14d
0xac5b: jmp      0xac6e
0xac6e:
0xac5d: nop      dword ptr [rax]
0xac60: mov      edx, dword ptr [rip + 0x202a2]
0xac66: add      r14d, eax
0xac69: cmp      edx, r14d
0xac6c: jle      0xac88
0xac6e: sub      edx, r14d
0xac71: movsxd   rsi, r14d
0xac74: mov      edi, 1
0xac79: add      rsi, r12
0xac7c: movsxd   rdx, edx
0xac7f: call     0x18da0
0xac84: test     eax, eax
0xac86: jns      0xac60
0xac88: mov      edx, 5
0xac8d: mov      rax, r12
0xac90: jmp      0xab0f
0xab0f:
0xac95: nop      dword ptr [rax]
0xac98: mov      ebx, 0
0xac9d: lea      rbp, [rip + 0x2027c]
0xaca4: jg       0xacbc
0xaca6: jmp      0xacd5
0xacd5:
0xaca8: nop      dword ptr [rax + rax]
0xacb0: mov      edx, dword ptr [rip + 0x20252]
0xacb6: add      ebx, eax
0xacb8: cmp      edx, ebx
0xacba: jle      0xacd5
0xacbc: sub      edx, ebx
0xacbe: movsxd   rsi, ebx
0xacc1: mov      edi, 1
0xacc6: add      rsi, rbp
0xacc9: movsxd   rdx, edx
0xaccc: call     0x18da0
0xacd1: test     eax, eax
0xacd3: jns      0xacb0

```



```

0xacd5: mov     dword ptr [rip + 0x20229], 0
0xacdf: jmp     0xabac
0xabac:
0xace4: call    0x49d0
0xace9: nop     dword ptr [rax]
0xacf0: endbr64
0xacf4: push    r12
0xacf6: xor     eax, eax
0xacf8: mov     r12, rdi
0xacfb: push    rbx
0xacfc: mov     ebx, esi
0xacfe: sub     rsp, 8
0xad02: call    0xa820
0xad07: mov     rdx, qword ptr [rip + 0x1fc1a]
0xad0e: test    rdx, rdx
0xad11: je      0xad17
0xad13: xor     eax, eax
0xad15: call    rdx
0xad17: movsxd  rdx, ebx
0xad1a: mov     rsi, r12
0xad1d: xor     edi, edi
0xad1f: call    0x18d30
0xad24: mov     rdx, qword ptr [rip + 0x1fbf5]
0xad2b: mov     r12d, eax
0xad2e: test    rdx, rdx
0xad31: je      0xad37
0xad33: xor     eax, eax
0xad35: call    rdx
0xad37: add     rsp, 8
0xad3b: mov     eax, r12d
0xad3e: pop     rbx
0xad3f: pop     r12
0xad41: ret
0xad42: nop     word ptr cs:[rax + rax]
0xad4d: nop     dword ptr [rax]
0xad50: push    r12
0xad52: mov     edi, 1
0xad57: push    rbp
0xad58: lea     rbp, [rip + 0x1fda1]
0xad5f: push    rbx
0xad60: sub     rsp, 0x50
0xad64: mov     rax, qword ptr fs:[0x28]
0xad6d: mov     qword ptr [rsp + 0x48], rax
0xad72: xor     eax, eax
0xad74: mov     r12, rsp
0xad77: mov     rsi, r12
0xad7a: call    0x4d60
0xad7f: mov     edx, 0xff

```

```

0xad84: mov     esi, 1
0xad89: mov     edi, 1
0xad8e: mov     word ptr [rsp + 0x11], dx
0xad93: mov     rdx, r12
0xad96: call    0x4d70
0xad9b: mov     edi, 1
0xada0: mov     dword ptr [rip + 0x1ef7a], 0xffffffff
0xadaa: call    0x18b70
0xadaf: mov     dword ptr [rip + 0x20147], 0
0xadb9: nop     dword ptr [rax]
0xadc0: xor     eax, eax
0xadc2: call    0xa820
0xadc7: mov     rdx, qword ptr [rip + 0x1fb5a]
0xadce: test    rdx, rdx
0xadd1: je      0xadd7
0xadd3: xor     eax, eax
0xadd5: call    rdx
0xadd7: mov     edx, 0x400
0xaddc: mov     rsi, rbp
0xaddf: xor     edi, edi
0xade1: call    0x18d30
0xade6: mov     rdx, qword ptr [rip + 0x1fb33]
0xaded: mov     ebx, eax
0xadeb: test    rdx, rdx
0xadf2: je      0xadf8
0xadf4: xor     eax, eax
0xadf6: call    rdx
0xadf8: mov     dword ptr [rip + 0x20106], ebx
0xadfe: test    ebx, ebx
0xae00: js      0xadc0
0xae02: xor     edi, edi
0xae04: call    0x18b70
0xae09: mov     rsi, r12
0xae0c: mov     edi, 1
0xae11: call    0x4d60
0xae16: mov     eax, 7
0xae1b: mov     rdx, r12
0xae1e: mov     esi, 1
0xae23: mov     edi, 1
0xae28: mov     word ptr [rsp + 0x11], ax
0xae2d: call    0x4d70
0xae32: mov     eax, dword ptr [rip + 0x200cc]
0xae38: mov     dword ptr [rip + 0x1eee2], 7
0xae42: test    eax, eax
0xae44: je      0xae9f
0xae46: movsxd  rsi, dword ptr [rip + 0x200b3]
0xae4d: movsx   r8d, byte ptr [rbp + rsi]
0xae53: mov     rdx, rsi

```

```

0xae56: mov     ecx, r8d
0xae59: test    r8b, r8b
0xae5c: js      0xae90
0xae5e: sub     eax, 1
0xae61: add     edx, 1
0xae64: mov     dword ptr [rip + 0x2009a], eax
0xae6a: mov     dword ptr [rip + 0x20090], edx
0xae70: mov     rax, qword ptr [rsp + 0x48]
0xae75: sub     rax, qword ptr fs:[0x28]
0xae7e: jne     0xaea7
0xae80: add     rsp, 0x50
0xae84: mov     eax, r8d
0xae87: pop     rbx
0xae88: pop     rbp
0xae89: pop     r12
0xae8b: ret
0xae8c: nop     dword ptr [rax]
0xae90: and     ecx, 0x7f
0xae93: mov     r8d, 0x1b
0xae99: mov     byte ptr [rbp + rsi], cl
0xae9d: jmp     0xae70
0xae70:
0xae9f: mov     r8d, 0xffffffff
0xaea5: jmp     0xae70
0xae70:
0xaea7: call    0x49d0
0xaeac: nop     dword ptr [rax]
0xaeb0: endbr64
0xaeb4: xor     eax, eax
0xaeb6: sub     rsp, 8
0xaeba: call    0x18ab0
0xaebf: mov     eax, dword ptr [rip + 0x2003f]
0xaec5: test    eax, eax
0xaec7: jne     0xaed8
0xaec9: xor     eax, eax
0xaecb: add     rsp, 8
0xaecf: jmp     0xad50
0xad50:
0xaed4: nop     dword ptr [rax]
0xaed8: movsxd  rsi, dword ptr [rip + 0x20021]
0xaedf: lea     rdi, [rip + 0x1fc1a]
0xae6: movsx  r8d, byte ptr [rdi + rsi]
0xaeeb: mov     rdx, rsi
0xaeee: mov     ecx, r8d
0xae71: test    r8b, r8b
0xae74: js      0xaf10
0xae76: sub     eax, 1
0xae79: add     edx, 1

```

```

0xae0c: mov     dword ptr [rip + 0x20002], eax
0xae0e: mov     dword ptr [rip + 0x1fff8], edx
0xae10: mov     eax, r8d
0xae12: add     rsp, 8
0xae14: ret
0xae16: and     ecx, 0x7f
0xae18: mov     r8d, 0x1b
0xae1a: mov     byte ptr [rdi + rsi], cl
0xae1c: jmp     0xae08
0xae1e: nop
0xae20: endbr64
0xae22: push    r13
0xae24: mov     r13, rsi
0xae26: push    r12
0xae28: mov     r12, rdx
0xae2a: push    rbp
0xae2c: mov     rbp, rdi
0xae2e: mov     edi, 0x18
0xae30: push    rbx
0xae32: sub     rsp, 8
0xae34: call    0x1d670
0xae36: mov     rdi, r13
0xae38: mov     rbx, rax
0xae3a: call    0x1d7a0
0xae3c: mov     qword ptr [rbx + 0x10], r12
0xae3e: mov     qword ptr [rbx], rax
0xae40: mov     rax, qword ptr [rbp]
0xae42: mov     qword ptr [rbx + 8], rax
0xae44: mov     qword ptr [rbp], rbx
0xae46: add     rsp, 8
0xae48: pop     rbx
0xae4a: pop     rbp
0xae4c: pop     r12
0xae4e: pop     r13
0xae50: ret
0xae52: nop     word ptr [rax + rax]
0xae54: endbr64
0xae56: cmp     byte ptr [rdi], 0
0xae58: je      0xb030
0xae5a: push    r13
0xae5c: push    r12
0xae5e: mov     r12, rsi
0xae60: push    rbp
0xae62: mov     rbp, rdi
0xae64: push    rbx
0xae66: sub     rsp, 8
0xae68: mov     r13, qword ptr [rip + 0x1fa0c]

```

```

0xaf94: test    r13, r13
0xaf97: je        0xb010
0xaf99: mov     rbx, qword ptr [r13]
0xaf9d: mov     rsi, qword ptr [rbx]
0xafa0: call    0x4b30
0xafa5: test    eax, eax
0xafa7: jle     0xb010
0xafa9: nop     dword ptr [rax]
0xafb0: lea     r13, [rbx + 8]
0xafb4: mov     rbx, qword ptr [rbx + 8]
0xafb8: mov     qword ptr [rip + 0x1f9e1], r13
0xafbf: test    rbx, rbx
0xafc2: je      0xafd3
0xafc4: mov     rsi, qword ptr [rbx]
0xafc7: mov     rdi, rbp
0xafca: call    0x4b30
0xafcf: test    eax, eax
0xafd1: jg      0xafb0
0xafd3: mov     edi, 0x18
0xafd8: call    0x1d670
0xafdd: mov     rdi, rbp
0xafe0: mov     rbx, rax
0xafe3: call    0x1d7a0
0xafe8: mov     qword ptr [rbx + 0x10], r12
0xafec: mov     qword ptr [rbx], rax
0xafef: mov     rax, qword ptr [r13]
0xaff3: mov     qword ptr [rbx + 8], rax
0xaff7: mov     qword ptr [r13], rbx
0xaffb: add     rsp, 8
0xafff: pop     rbx
0xb000: pop     rbp
0xb001: pop     r12
0xb003: pop     r13
0xb005: ret
0xb006: nop     word ptr cs:[rax + rax]
0xb010: mov     rbx, qword ptr [rip + 0x1f969]
0xb017: lea     r13, [rip + 0x1f962]
0xb01e: mov     qword ptr [rip + 0x1f97b], r13
0xb025: test    rbx, rbx
0xb028: jne     0xafc4
0xb02a: jmp     0xafd3
0xafd3:
0xb02c: nop     dword ptr [rax]
0xb030: ret
0xb031: nop     word ptr cs:[rax + rax]
0xb03c: nop     dword ptr [rax]
0xb040: endbr64
0xb044: mov     rax, qword ptr [rip + 0x1f935]

```

```

0xb04b: mov      qword ptr [rip + 0x1f926], rax
0xb052: ret
0xb053: nop      word ptr cs:[rax + rax]
0xb05e: nop
0xb060: endbr64
0xb064: cmp      qword ptr [rip + 0x1f90c], 0
0xb06c: je       0xb073
0xb06e: jmp      0x99e0
0x99e0
0xb073: xor      eax, eax
0xb075: ret
0xb076: nop      word ptr cs:[rax + rax]
0xb080: endbr64
0xb084: mov      eax, dword ptr [rip + 0x1f912]
0xb08a: push     r12
0xb08c: mov      r12, rdi
0xb08f: push     rbp
0xb090: push     rbx
0xb091: sub      eax, 1
0xb094: movzx    ebx, byte ptr [rip + 0x202b6]
0xb09b: mov      dword ptr [rip + 0x202b3], eax
0xb0a1: mov      dword ptr [rip + 0x202ad], 0
0xb0ab: mov      eax, ebx
0xb0ad: and      eax, 0xffffffffc0
0xb0b0: or       eax, 0x38
0xb0b3: mov      byte ptr [rip + 0x20298], al
0xb0b9: mov      eax, dword ptr [rip + 0x1f8d9]
0xb0bf: lea      edi, [rax + 1]
0xb0c2: movsxd   rdi, edi
0xb0c5: call     0x1d670
0xb0ca: movsxd   rdx, dword ptr [rip + 0x1f8c7]
0xb0d1: mov      esi, 0x20
0xb0d6: mov      rdi, rax
0xb0d9: mov      rbp, rax
0xb0dc: call     0x4a50
0xb0e1: movsxd   rax, dword ptr [rip + 0x1f8b0]
0xb0e8: mov      rdi, rbp
0xb0eb: mov      byte ptr [rbp + rax], 0
0xb0f0: mov      esi, dword ptr [rip + 0x1f8a2]
0xb0f6: call     0xa1a0
0xb0fb: mov      rdi, rbp
0xb0fe: lea      rbp, [rip + 0x1f8bb]
0xb105: call     0x18d10
0xb10a: mov      eax, dword ptr [rip + 0x1f88c]
0xb110: mov      rdi, r12
0xb113: mov      dword ptr [rip + 0x2023b], 0
0xb11d: sub      eax, 1
0xb120: mov      dword ptr [rip + 0x2022e], eax

```

```

0xb126: call    0x9fe0
0xb12b: mov     esi, 0x13
0xb130: lea     rdi, [rip + 0x15ad5]
0xb137: call    0xa1a0
0xb13c: mov     rdi, rbp
0xb13f: call    0x49b0
0xb144: mov     rdi, rbp
0xb147: mov     esi, eax
0xb149: call    0xa1a0
0xb14e: mov     esi, 1
0xb153: lea     rdi, [rip + 0x155eb]
0xb15a: call    0xa1a0
0xb15f: xor     eax, eax
0xb161: call    0xa820
0xb166: mov     byte ptr [rip + 0x201e5], bl
0xb16c: pop     rbx
0xb16d: pop     rbp
0xb16e: pop     r12
0xb170: ret
0xb171: nop     word ptr cs:[rax + rax]
0xb17c: nop     dword ptr [rax]
0xb180: endbr64
0xb184: push    r15
0xb186: push    r14
0xb188: mov     r14, rdi
0xb18b: push    r13
0xb18d: push    r12
0xb18f: push    rbp
0xb190: lea     rbp, [rip + 0x1f969]
0xb197: push    rbx
0xb198: sub     rsp, 0x28
0xb19c: jmp     0xb1ae
0xb1ae:
0xb19e: nop
0xb1a0: xor     eax, eax
0xb1a2: call    0xad50
0xb1a7: mov     ebx, eax
0xb1a9: cmp     ebx, -1
0xb1ac: jne     0xb1ef
0xb1ae: xor     eax, eax
0xb1b0: call    0x18ab0
0xb1b5: mov     eax, dword ptr [rip + 0x1fd49]
0xb1bb: test    eax, eax
0xb1bd: je      0xb1a0
0xb1bf: movsxd  rsi, dword ptr [rip + 0x1fd3a]
0xb1c6: movsx   ebx, byte ptr [rbp + rsi]
0xb1cb: mov     rdx, rsi
0xb1ce: mov     ecx, ebx

```

```

0xb1d0: test    bl, bl
0xb1d2: js      0xb4a0
0xb1d8: sub     eax, 1
0xb1db: add     edx, 1
0xb1de: mov     dword ptr [rip + 0x1fd20], eax
0xb1e4: mov     dword ptr [rip + 0x1fd16], edx
0xb1ea: cmp     ebx, -1
0xb1ed: je      0xb1ae
0xb1ef: test    r14, r14
0xb1f2: je      0xb1fb
0xb1f4: mov     dword ptr [r14], 1
0xb1fb: mov     eax, dword ptr [rip + 0x201fb]
0xb201: test    ebx, ebx
0xb203: jne     0xb445
0xb209: mov     ebx, 0xff
0xb20e: test    eax, eax
0xb210: jne     0xb240
0xb212: xor     r12d, r12d
0xb215: mov     ebp, 0x1fe
0xb21a: mov     ebx, 0xff
0xb21f: call    0x4f00
0xb224: mov     rax, qword ptr [rax]
0xb227: movzx   eax, word ptr [rax + rbp]
0xb22b: test    ah, 0x40
0xb22e: jne     0xb47d
0xb234: test    r12b, r12b
0xb237: jne     0xb47d
0xb23d: nop     dword ptr [rax]
0xb240: mov     rax, qword ptr [rip + 0x1f739]
0xb247: xor     r12d, r12d
0xb24a: lea     r13, [rip + 0x1f8af]
0xb251: mov     dword ptr [rip + 0x1f751], 0
0xb25b: mov     dword ptr [rip + 0x1f743], 0
0xb265: mov     qword ptr [rip + 0x1f70c], rax
0xb26c: test    ebx, ebx
0xb26e: mov     eax, 0xffffffff
0xb273: movsxd  rbp, r12d
0xb276: cmovne  eax, ebx
0xb279: mov     rbx, qword ptr [rip + 0x200e0]
0xb280: mov     byte ptr [rbx + rbp], al
0xb283: add     rbp, 1
0xb287: mov     byte ptr [rbx + rbp], 0
0xb28b: cmp     qword ptr [rip + 0x1f6e5], 0
0xb293: je      0xb2b3
0xb295: mov     rdi, rbx
0xb298: call    0x99e0
0xb29d: mov     r15, rax
0xb2a0: cmp     rax, -1

```



```

0xb2a4: je      0xb33a
0xb2aa: test    rax, rax
0xb2ad: jne     0xb38f
0xb2b3: mov     ecx, dword ptr [rip + 0x1fc4b]
0xb2b9: test    ecx, ecx
0xb2bb: jne     0xb2ee
0xb2bd: mov     edx, dword ptr [rip + 0x1f6e5]
0xb2c3: test    edx, edx
0xb2c5: je      0xb380
0xb2cb: mov     rdi, rbx
0xb2ce: call    0xb080
0xb2d3: mov     dword ptr [rip + 0x1f6cf], 1
0xb2dd: jmp     0xb2ee
0xb2ee
0xb2df: nop
0xb2e0: xor     eax, eax
0xb2e2: call    0xad50
0xb2e7: mov     ebx, eax
0xb2e9: cmp     ebx, -1
0xb2ec: jne     0xb32c
0xb2ee: xor     eax, eax
0xb2f0: call    0x18ab0
0xb2f5: mov     eax, dword ptr [rip + 0x1fc09]
0xb2fb: test    eax, eax
0xb2fd: je      0xb2e0
0xb2ff: movsxd  rsi, dword ptr [rip + 0x1fbfa]
0xb306: movsx   ebx, byte ptr [r13 + rsi]
0xb30c: mov     rdx, rsi
0xb30f: mov     ecx, ebx
0xb311: test    bl, bl
0xb313: js      0xb360
0xb315: sub     eax, 1
0xb318: add     edx, 1
0xb31b: mov     dword ptr [rip + 0x1fbe3], eax
0xb321: mov     dword ptr [rip + 0x1fbd9], edx
0xb327: cmp     ebx, -1
0xb32a: je      0xb2ee
0xb32c: add     r12d, 1
0xb330: cmp     r12d, 0xf
0xb334: jne     0xb26c
0xb33a: mov     edi, 1
0xb33f: xor     r15d, r15d
0xb342: call    0x4aa0
0xb347: mov     dword ptr [rip + 0x1f65b], 0
0xb351: jmp     0xb433
0xb433
0xb356: nop     word ptr cs:[rax + rax]
0xb360: and     ecx, 0x7f

```

```

0xb363: add    r12d, 1
0xb367: mov    byte ptr [r13 + rsi], cl
0xb36c: cmp    r12d, 0xf
0xb370: je     0xb33a
0xb372: mov    eax, 0x1b
0xb377: jmp    0xb279
0xb279:
0xb37c: nop    dword ptr [rax]
0xb380: mov    edi, 1
0xb385: call   0x4aa0
0xb38a: jmp    0xb2d3
0xb2d3:
0xb38f: test   r14, r14
0xb392: je     0xb41f
0xb398: mov    ecx, dword ptr [rip + 0x1fb66]
0xb39e: cmp    r12d, ecx
0xb3a1: jge    0xb41f
0xb3a3: movsxd rax, dword ptr [rip + 0x1fb56]
0xb3aa: lea    r9d, [r12 + 1]
0xb3af: lea    rdx, [rip + 0x1f74a]
0xb3b6: movsxd rdi, r9d
0xb3b9: mov    qword ptr [rsp + 0x18], rdi
0xb3be: lea    r8d, [r9 + rax]
0xb3c2: lea    r13, [rax + rdx]
0xb3c6: jmp    0xb3f1
0xb3f1:
0xb3c8: nop    dword ptr [rax + rax]
0xb3d0: sub    ecx, r9d
0xb3d3: add    dword ptr [r14], 1
0xb3d7: add    r13, qword ptr [rsp + 0x18]
0xb3dc: mov    dword ptr [rip + 0x1fb1d], r8d
0xb3e3: add    r8d, r9d
0xb3e6: mov    dword ptr [rip + 0x1fb18], ecx
0xb3ec: cmp    r12d, ecx
0xb3ef: jge    0xb41f
0xb3f1: mov    rdx, rbp
0xb3f4: mov    rsi, r13
0xb3f7: mov    rdi, rbx
0xb3fa: mov    dword ptr [rsp + 0x14], r8d
0xb3ff: mov    dword ptr [rsp + 0x10], r9d
0xb404: mov    dword ptr [rsp + 0xc], ecx
0xb408: call   0x4af0
0xb40d: mov    ecx, dword ptr [rsp + 0xc]
0xb411: mov    r9d, dword ptr [rsp + 0x10]
0xb416: test   eax, eax
0xb418: mov    r8d, dword ptr [rsp + 0x14]
0xb41d: je     0xb3d0
0xb41f: mov    edi, 1

```

```

0xb424: call    0x4aa0
0xb429: mov     dword ptr [rip + 0x1f579], 0
0xb433: add     rsp, 0x28
0xb437: mov     rax, r15
0xb43a: pop     rbx
0xb43b: pop     rbp
0xb43c: pop     r12
0xb43e: pop     r13
0xb440: pop     r14
0xb442: pop     r15
0xb444: ret
0xb445: test    eax, eax
0xb447: jne     0xb240
0xb44d: cmp     ebx, 0xa
0xb450: sete    bpl
0xb454: cmp     ebx, 0xd
0xb457: sete    al
0xb45a: or      ebp, eax
0xb45c: call    0x4f00
0xb461: mov     rax, qword ptr [rax]
0xb464: test    bpl, bpl
0xb467: je      0xb4c6
0xb469: mov     ebx, 0xa
0xb46e: test    byte ptr [rax + 0x15], 0x40
0xb472: je      0xb240
0xb478: mov     ebx, 0xa
0xb47d: mov     rax, qword ptr [rip + 0x1f4dc]
0xb484: lea     r15, [rip + 0x1f4d5]
0xb48b: mov     byte ptr [rax], bl
0xb48d: mov     rax, qword ptr [rip + 0x1f4cc]
0xb494: mov     byte ptr [rax + 1], 0
0xb498: jmp     0xb433
0xb433:
0xb49a: nop     word ptr [rax + rax]
0xb4a0: and     ecx, 0x7f
0xb4a3: mov     byte ptr [rbp + rsi], cl
0xb4a7: test    r14, r14
0xb4aa: je      0xb4b3
0xb4ac: mov     dword ptr [r14], 1
0xb4b3: mov     eax, dword ptr [rip + 0x1ff43]
0xb4b9: mov     ebx, 0x1b
0xb4be: test    eax, eax
0xb4c0: jne     0xb240
0xb4c6: movsxd  rbp, ebx
0xb4c9: add     rbp, rbp
0xb4cc: cmp     ebx, 7
0xb4cf: sete    r12b
0xb4d3: jmp     0xb21f

```

```

0xb21f
0xb4d8: nop      dword ptr [rax + rax]
0xb4e0: endbr64
0xb4e4: mov     eax, dword ptr [rip + 0x1f4c2]
0xb4ea: test    eax, eax
0xb4ec: jne     0xb4f0
0xb4ee: ret
0xb4ef: nop
0xb4f0: sub     rsp, 8
0xb4f4: mov     rdi, qword ptr [rip + 0x1fe65]
0xb4fb: call    0xb080
0xb500: mov     dword ptr [rip + 0x1f49e], 1
0xb50a: add     rsp, 8
0xb50e: ret
0xb50f: nop
0xb510: endbr64
0xb514: mov     rax, qword ptr [rip + 0x1fe45]
0xb51b: ret
0xb51c: nop     dword ptr [rax]
0xb520: endbr64
0xb524: push    r15
0xb526: mov     esi, 0x5413
0xb52b: mov     edi, 1
0xb530: push    r14
0xb532: push    r13
0xb534: push    r12
0xb536: push    rbp
0xb537: push    rbx
0xb538: sub     rsp, 0x58
0xb53c: mov     rax, qword ptr fs:[0x28]
0xb545: mov     qword ptr [rsp + 0x48], rax
0xb54a: xor     eax, eax
0xb54c: lea     rdx, [rsp + 0x18]
0xb551: mov     dword ptr [rsp + 0x10], 0
0xb559: mov     dword ptr [rsp + 0x14], 0
0xb561: call    0x4a70
0xb566: test    eax, eax
0xb568: jne     0xb5a0
0xb56a: mov     ebx, eax
0xb56c: movzx   eax, word ptr [rsp + 0x1a]
0xb571: test    ax, ax
0xb574: je      0xb790
0xb57a: movzx   ebp, word ptr [rsp + 0x18]
0xb57f: test    bp, bp
0xb582: je      0xb790
0xb588: movzx   ebx, ax
0xb58b: lea     r12d, [rbp - 1]
0xb58f: lea     eax, [rbx - 1]

```

```

0xb592: mov     dword ptr [rsp + 0xc], eax
0xb596: jmp     0xb5b2
0xb5b2:
0xb598: nop     dword ptr [rax + rax]
0xb5a0: mov     dword ptr [rsp + 0xc], 0xffffffff
0xb5a8: mov     r12d, 0xffffffff
0xb5ae: xor     ebx, ebx
0xb5b0: xor     ebp, ebp
0xb5b2: lea     r13, [rip + 0x15667]
0xb5b9: lea     r15, [rip + 0x15a13]
0xb5c0: mov     rdi, r13
0xb5c3: call    0x47e0
0xb5c8: mov     rdi, rax
0xb5cb: test    rax, rax
0xb5ce: je      0xb5df
0xb5d0: lea     rdx, [rsp + 0x14]
0xb5d5: mov     rsi, r15
0xb5d8: xor     eax, eax
0xb5da: call    0x4c60
0xb5df: lea     r14, [rip + 0x15642]
0xb5e6: mov     rdi, r14
0xb5e9: call    0x47e0
0xb5ee: mov     rdi, rax
0xb5f1: test    rax, rax
0xb5f4: je      0xb605
0xb5f6: lea     rdx, [rsp + 0x10]
0xb5fb: mov     rsi, r15
0xb5fe: xor     eax, eax
0xb600: call    0x4c60
0xb605: cmp     dword ptr [rsp + 0xc], 0x3ff
0xb60d: mov     eax, dword ptr [rip + 0x1e8fd]
0xb613: jbe     0xb628
0xb615: mov     ebx, dword ptr [rsp + 0x14]
0xb619: lea     edx, [rbx - 1]
0xb61c: cmp     edx, 0x3ff
0xb622: ja      0xb7c0
0xb628: mov     dword ptr [rip + 0x1f36a], ebx
0xb62e: cmp     r12d, 0x3ff
0xb635: jbe     0xb64a
0xb637: mov     ebp, dword ptr [rsp + 0x10]
0xb63b: lea     edx, [rbp - 1]
0xb63e: cmp     edx, 0x3ff
0xb644: ja      0xb7a8
0xb64a: lea     r12, [rsp + 0x20]
0xb64f: mov     r8d, ebp
0xb652: mov     rcx, r15
0xb655: mov     edx, 0x20
0xb65a: mov     esi, 1

```

```

0xb65f: mov     rdi, r12
0xb662: xor     eax, eax
0xb664: mov     dword ptr [rip + 0x1f332], ebp
0xb66a: call    0x4f10
0xb66f: mov     rsi, r12
0xb672: mov     rdi, r14
0xb675: call    0x18380
0xb67a: mov     rcx, r15
0xb67d: mov     edx, 0x20
0xb682: mov     rdi, r12
0xb685: mov     r8d, dword ptr [rip + 0x1f30c]
0xb68c: mov     esi, 1
0xb691: xor     eax, eax
0xb693: call    0x4f10
0xb698: mov     rdi, r13
0xb69b: mov     rsi, r12
0xb69e: call    0x18380
0xb6a3: mov     rdi, qword ptr [rip + 0x1fc8e]
0xb6aa: test    rdi, rdi
0xb6ad: je      0xb6b4
0xb6af: call    0x18d10
0xb6b4: mov     rdi, qword ptr [rip + 0x1fc75]
0xb6bb: test    rdi, rdi
0xb6be: je      0xb6c5
0xb6c0: call    0x18d10
0xb6c5: mov     rdi, qword ptr [rip + 0x1fc5c]
0xb6cc: test    rdi, rdi
0xb6cf: je      0xb6d6
0xb6d1: call    0x18d10
0xb6d6: mov     rdi, qword ptr [rip + 0x1fc43]
0xb6dd: test    rdi, rdi
0xb6e0: je      0xb6e7
0xb6e2: call    0x18d10
0xb6e7: mov     edi, dword ptr [rip + 0x1f2ab]
0xb6ed: imul    edi, dword ptr [rip + 0x1f2a8]
0xb6f4: mov     esi, 1
0xb6f9: movsxd  rdi, edi
0xb6fc: call    0x1d740
0xb701: mov     edi, dword ptr [rip + 0x1f291]
0xb707: mov     esi, 1
0xb70c: imul    edi, dword ptr [rip + 0x1f289]
0xb713: mov     qword ptr [rip + 0x1fc1e], rax
0xb71a: movsxd  rdi, edi
0xb71d: call    0x1d740
0xb722: mov     edi, dword ptr [rip + 0x1f270]
0xb728: mov     esi, 1
0xb72d: imul    edi, dword ptr [rip + 0x1f268]
0xb734: mov     qword ptr [rip + 0x1fbf5], rax

```

```

0xb73b: movsxd  rdi, edi
0xb73e: call  0x1d740
0xb743: mov    edi, dword ptr [rip + 0x1f24f]
0xb749: mov    esi, 1
0xb74e: imul   edi, dword ptr [rip + 0x1f247]
0xb755: mov    qword ptr [rip + 0x1fbcc], rax
0xb75c: movsxd  rdi, edi
0xb75f: call  0x1d740
0xb764: mov    qword ptr [rip + 0x1fbb5], rax
0xb76b: mov    rax, qword ptr [rsp + 0x48]
0xb770: sub    rax, qword ptr fs:[0x28]
0xb779: jne    0xb7dc
0xb77b: add    rsp, 0x58
0xb77f: pop    rbx
0xb780: pop    rbp
0xb781: pop    r12
0xb783: pop    r13
0xb785: pop    r14
0xb787: pop    r15
0xb789: ret
0xb78a: nop    word ptr [rax + rax]
0xb790: mov    dword ptr [rsp + 0xc], 0xffffffff
0xb798: xor    ebp, ebp
0xb79a: mov    r12d, 0xffffffff
0xb7a0: jmp    0xb5b2
0xb5b2
0xb7a5: nop    dword ptr [rax]
0xb7a8: lea    edx, [rax - 1]
0xb7ab: mov    ebp, 0x18
0xb7b0: cmp    edx, 0x400
0xb7b6: cmovb  ebp, eax
0xb7b9: jmp    0xb64a
0xb64a
0xb7be: nop
0xb7c0: mov    ebx, dword ptr [rip + 0x1e72a]
0xb7c6: lea    edx, [rbx - 1]
0xb7c9: cmp    edx, 0x3ff
0xb7cf: mov    edx, 0x50
0xb7d4: cmova  ebx, edx
0xb7d7: jmp    0xb628
0xb628
0xb7dc: call  0x49d0
0xb7e1: nop    word ptr cs:[rax + rax]
0xb7ec: nop    dword ptr [rax]
0xb7f0: endbr64
0xb7f4: push  r15
0xb7f6: push  r14
0xb7f8: push  r13

```

```

0xb7fa: push    r12
0xb7fc: mov     r12, rdi
0xb7ff: mov     edi, 0x5858
0xb804: push    rbp
0xb805: push    rbx
0xb806: mov     ebx, esi
0xb808: sub     rsp, 0x38
0xb80c: mov     rax, qword ptr fs:[0x28]
0xb815: mov     qword ptr [rsp + 0x28], rax
0xb81a: movabs  rax, 0x7363762f7665642f
0xb824: mov     word ptr [rsp + 8], di
0xb829: mov     byte ptr [rsp + 0xa], 0
0xb82e: mov     dword ptr [rsp + 0x18], 0x585861
0xb836: mov     qword ptr [rsp], rax
0xb83a: mov     qword ptr [rsp + 0x10], rax
0xb83f: cmp     esi, 1
0xb842: je      0xb978
0xb848: mov     esi, dword ptr [rip + 0x1fbb2]
0xb84e: mov     r13, qword ptr [rip + 0x1d783]
0xb855: mov     r14d, 1
0xb85b: test    esi, esi
0xb85d: jne     0xb90c
0xb863: mov     rax, qword ptr [rip + 0x1f126]
0xb86a: lea     r15, [rsp + 0x10]
0xb86f: mov     esi, r14d
0xb872: mov     dword ptr [rip + 0x1fb84], 0
0xb87c: mov     rdi, r15
0xb87f: movzx   edx, byte ptr [rax + 8]
0xb883: mov     byte ptr [rsp + 0x19], dl
0xb887: movzx   eax, byte ptr [rax + 9]
0xb88b: mov     byte ptr [rsp + 0x1a], al
0xb88f: call    0x4930
0xb894: mov     ebp, eax
0xb896: cmp     eax, -1
0xb899: je      0xb8f0
0xb89b: mov     eax, dword ptr [rip + 0x1f0fb]
0xb8a1: imul    eax, dword ptr [rip + 0x1f0f0]
0xb8a8: mov     edi, ebp
0xb8aa: mov     rsi, r12
0xb8ad: lea     edx, [rax + rax + 4]
0xb8b1: xor     eax, eax
0xb8b3: call    r13
0xb8b6: mov     edi, ebp
0xb8b8: call    0x4ab0
0xb8bd: cmp     ebx, 2
0xb8c0: je      0xb990
0xb8c6: mov     eax, 1
0xb8cb: mov     rdx, qword ptr [rsp + 0x28]

```



```

0xb8d0: sub    rdx, qword ptr fs:[0x28]
0xb8d9: jne     0xbb07
0xb8df: add     rsp, 0x38
0xb8e3: pop     rbx
0xb8e4: pop     rbp
0xb8e5: pop     r12
0xb8e7: pop     r13
0xb8e9: pop     r14
0xb8eb: pop     r15
0xb8ed: ret
0xb8ee: nop
0xb8f0: mov     ecx, 0x30
0xb8f5: mov     esi, r14d
0xb8f8: mov     rdi, r15
0xb8fb: mov     word ptr [rsp + 0x19], cx
0xb900: call    0x4930
0xb905: mov     ebp, eax
0xb907: cmp     eax, -1
0xb90a: jne     0xb89b
0xb90c: mov     dword ptr [rip + 0x1faea], 1
0xb916: cmp     ebx, 2
0xb919: je      0xba30
0xb91f: mov     rax, qword ptr [rip + 0x1f06a]
0xb926: mov     r15, rsp
0xb929: mov     esi, r14d
0xb92c: mov     rdi, r15
0xb92f: movzx   edx, byte ptr [rax + 8]
0xb933: movzx   eax, byte ptr [rax + 9]
0xb937: mov     byte ptr [rsp + 8], dl
0xb93b: mov     byte ptr [rsp + 9], al
0xb93f: call    0x4930
0xb944: mov     ebp, eax
0xb946: cmp     eax, -1
0xb949: je      0xba00
0xb94f: mov     edx, dword ptr [rip + 0x1f043]
0xb955: mov     edi, ebp
0xb957: mov     rsi, r12
0xb95a: xor     eax, eax
0xb95c: imul    edx, dword ptr [rip + 0x1f039]
0xb963: add     edx, 4
0xb966: call    r13
0xb969: mov     edi, ebp
0xb96b: call    0x4ab0
0xb970: jmp     0xb8c6
0xb8c6:
0xb975: nop     dword ptr [rax]
0xb978: mov     r13, qword ptr [rip + 0x1d661]
0xb97f: xor     r14d, r14d

```

```

0xb982: jmp      0xb863
0xb863
0xb987: nop      word ptr [rax + rax]
0xb990: movsx     esi, byte ptr [r12 + 2]
0xb996: movsx     edi, byte ptr [r12 + 3]
0xb99c: call     0xa7a0
0xb9a1: mov      edx, dword ptr [rip + 0x1f561]
0xb9a7: test     edx, edx
0xb9a9: jle      0xb9e5
0xb9ab: xor      ebp, ebp
0xb9ad: lea      rbx, [rip + 0x1f56c]
0xb9b4: jmp      0xb9cc
0xb9cc
0xb9b6: nop      word ptr cs:[rax + rax]
0xb9c0: mov      edx, dword ptr [rip + 0x1f542]
0xb9c6: add      ebp, eax
0xb9c8: cmp      edx, ebp
0xb9ca: jle      0xb9e5
0xb9cc: sub      edx, ebp
0xb9ce: movsxd   rsi, ebp
0xb9d1: mov      edi, 1
0xb9d6: add      rsi, rbx
0xb9d9: movsxd   rdx, edx
0xb9dc: call     0x18da0
0xb9e1: test     eax, eax
0xb9e3: jns      0xb9c0
0xb9e5: mov      dword ptr [rip + 0x1f519], 0
0xb9ef: mov      eax, 1
0xb9f4: jmp      0xb8cb
0xb8cb
0xb9f9: nop      dword ptr [rax]
0xba00: mov      edx, 0x30
0xba05: mov      esi, r14d
0xba08: mov      rdi, r15
0xba0b: mov      word ptr [rsp + 8], dx
0xba10: call     0x4930
0xba15: mov      ebp, eax
0xba17: cmp      eax, -1
0xba1a: jne      0xb94f
0xba20: xor      eax, eax
0xba22: jmp      0xb8cb
0xb8cb
0xba27: nop      word ptr [rax + rax]
0xba30: xor      eax, eax
0xba32: mov      r15, rsp
0xba35: call     0xa300
0xba3a: mov      rax, qword ptr [rip + 0x1ef4f]
0xba41: mov      esi, r14d

```

```

0xba44: mov     rdi, r15
0xba47: movzx   edx, byte ptr [rax + 8]
0xba4b: movzx   eax, byte ptr [rax + 9]
0xba4f: mov     byte ptr [rsp + 8], dl
0xba53: mov     byte ptr [rsp + 9], al
0xba57: call    0x4930
0xba5c: mov     ebp, eax
0xba5e: cmp     eax, -1
0xba61: je      0xbae0
0xba63: mov     edx, dword ptr [rip + 0x1ef2f]
0xba69: mov     rsi, r12
0xba6c: mov     edi, ebp
0xba6e: xor     eax, eax
0xba70: imul    edx, dword ptr [rip + 0x1ef25]
0xba77: add     edx, 4
0xba7a: call    r13
0xba7d: mov     edi, ebp
0xba7f: call    0x4ab0
0xba84: mov     eax, dword ptr [rip + 0x1ef12]
0xba8a: xor     esi, esi
0xba8c: lea     edi, [rax - 1]
0xba8f: call    0xa7a0
0xba94: mov     edx, dword ptr [rip + 0x1f46e]
0xba9a: test    edx, edx
0xba9c: jle     0xb9e5
0xbaa2: xor     ebp, ebp
0xbaa4: lea     rbx, [rip + 0x1f475]
0xbaab: jmp     0xbac0
0xbac0:
0xbaad: nop     dword ptr [rax]
0xbab0: mov     edx, dword ptr [rip + 0x1f452]
0xbab6: add     ebp, eax
0xbab8: cmp     edx, ebp
0xbaba: jle     0xb9e5
0xbac0: sub     edx, ebp
0xbac2: movsxd  rsi, ebp
0xbac5: mov     edi, 1
0xbaca: add     rsi, rbx
0xbacd: movsxd  rdx, edx
0xbad0: call    0x18da0
0xbad5: test    eax, eax
0xbad7: jns     0xbab0
0xbad9: jmp     0xb9e5
0xb9e5:
0xbade: nop
0xbae0: mov     eax, 0x30
0xbae5: mov     esi, r14d
0xbae8: mov     rdi, r15

```

```

0xbaeb: mov     word ptr [rsp + 8], ax
0xbaf0: call    0x4930
0xbaf5: mov     ebp, eax
0xbaf7: cmp     eax, -1
0xbafa: jne     0xba63
0xbb00: xor     eax, eax
0xbb02: jmp     0xb8cb
0xb8cb
0xbb07: call    0x49d0
0xbb0c: nop     dword ptr [rax]
0xbb10: endbr64
0xbb14: mov     eax, dword ptr [rip + 0x1ee6e]
0xbb1a: test    eax, eax
0xbb1c: jne     0xbb20
0xbb1e: ret
0xbb1f: nop
0xbb20: sub     rsp, 8
0xbb24: mov     esi, 1
0xbb29: xor     eax, eax
0xbb2b: call    0xb7f0
0xbb30: mov     dword ptr [rip + 0x1f8ce], eax
0xbb36: add     rsp, 8
0xbb3a: ret
0xbb3b: nop     dword ptr [rax + rax]
0xbb40: endbr64
0xbb44: push    rbp
0xbb45: xor     eax, eax
0xbb47: mov     rbp, rdi
0xbb4a: call    0xa6c0
0xbb4f: mov     edx, dword ptr [rip + 0x1ee33]
0xbb55: test    edx, edx
0xbb57: je      0xbb94
0xbb59: mov     eax, dword ptr [rip + 0x1f8a5]
0xbb5f: test    eax, eax
0xbb61: je      0xbb94
0xbb63: mov     edx, dword ptr [rip + 0x1ee33]
0xbb69: imul    edx, dword ptr [rip + 0x1ee28]
0xbb70: xor     esi, esi
0xbb72: mov     rdi, qword ptr [rip + 0x1f7af]
0xbb79: movsxd  rdx, edx
0xbb7c: call    0x4a50
0xbb81: xor     eax, eax
0xbb83: mov     esi, 2
0xbb88: mov     rdi, rbp
0xbb8b: call    0xb7f0
0xbb90: test    eax, eax
0xbb92: jne     0xbba0
0xbb94: xor     eax, eax

```

```

0xbb96: pop      rbp
0xbb97: jmp      0xa300
0xa300:
0xbb9c: nop      dword ptr [rax]
0xbba0: mov      edx, dword ptr [rip + 0x1edf6]
0xbba6: imul     edx, dword ptr [rip + 0x1edeb]
0xbbad: xor      esi, esi
0xbbae: mov      rdi, qword ptr [rip + 0x1f782]
0xbbb6: pop      rbp
0xbbb7: movsxd   rdx, edx
0xbbba: jmp      0x4a50
0x4a50:
0xbbbf: nop
0xbbc0: endbr64
0xbbc4: push     rbp
0xbbc5: mov      rbp, rdi
0xbbc8: push     rbx
0xbbc9: sub      rsp, 8
0xbbcd: cmp      dword ptr [rip + 0x1ed84], 1
0xbbdd: je       0xbcd0
0xbbda: xor      eax, eax
0xbbdc: call     0xa6c0
0xbbe1: test     rbp, rbp
0xbbe4: je       0xbbf4
0xbbe6: mov      eax, dword ptr [rip + 0x1ed9c]
0xbbec: test     eax, eax
0xbbee: jne      0xbcc0
0xbbf4: mov      eax, dword ptr [rip + 0x1eda2]
0xbbfa: mov      rdi, qword ptr [rip + 0x1e347]
0xbc01: lea      rdx, [rip - 0x2198]
0xbc08: lea      esi, [rax - 1]
0xbc0b: call     0x4820
0xbc10: mov      edx, dword ptr [rip + 0x1f2f2]
0xbc16: test     edx, edx
0xbc18: jle      0xbc4d
0xbc1a: xor      ebp, ebp
0xbc1c: lea      rbx, [rip + 0x1f2fd]
0xbc23: jmp      0xbc34
0xbc34:
0xbc25: nop      dword ptr [rax]
0xbc28: mov      edx, dword ptr [rip + 0x1f2da]
0xbc2e: add      ebp, eax
0xbc30: cmp      edx, ebp
0xbc32: jle      0xbc4d
0xbc34: sub      edx, ebp
0xbc36: movsxd   rsi, ebp
0xbc39: mov      edi, 1
0xbc3e: add      rsi, rbx

```

```

0xbc41: movsxd  rdx, edx
0xbc44: call     0x18da0
0xbc49: test     eax, eax
0xbc4b: jns      0xbc28
0xbc4d: mov      edi, dword ptr [rip + 0x1ed49]
0xbc53: xor      esi, esi
0xbc55: mov      dword ptr [rip + 0x1f2a9], 0
0xbc5f: call     0xa7a0
0xbc64: mov      edx, dword ptr [rip + 0x1f29e]
0xbc6a: test     edx, edx
0xbc6c: jle      0xbca5
0xbc6e: xor      ebp, ebp
0xbc70: lea      rbx, [rip + 0x1f2a9]
0xbc77: jmp      0xbc8c
0xbc8c:
0xbc79: nop      dword ptr [rax]
0xbc80: mov      edx, dword ptr [rip + 0x1f282]
0xbc86: add      ebp, eax
0xbc88: cmp      edx, ebp
0xbc8a: jle      0xbca5
0xbc8c: sub      edx, ebp
0xbc8e: movsxd   rsi, ebp
0xbc91: mov      edi, 1
0xbc96: add      rsi, rbx
0xbc99: movsxd   rdx, edx
0xbc9c: call     0x18da0
0xbca1: test     eax, eax
0xbca3: jns      0xbc80
0xbca5: mov      dword ptr [rip + 0x1f259], 0
0xbcaf: add      rsp, 8
0xbcb3: mov      edi, 0xa
0xbcb8: pop      rbx
0xbcb9: pop      rbp
0xbcba: jmp      0x4810
0x4810:
0xbcbf: nop
0xbcc0: mov      rdi, rbp
0xbcc3: call     0xbb40
0xbcc8: jmp      0xbbf4
0xbbf4:
0xbccd: nop      dword ptr [rax]
0xbcd0: lea      rdx, [rip + 0x1f6e9]
0xbcd7: mov      esi, 1
0xbcdc: mov      edi, 1
0xbce1: call     0x4d70
0xbce6: mov      dword ptr [rip + 0x1ec68], 0
0xbcf0: jmp      0xbbda
0xbbda:

```

```

0xbc5f: nop      word ptr cs:[rax + rax]
0xbd00: endbr64
0xbd04: push     r13
0xbd06: lea      r13, [rip + 0x1cc33]
0xbd0d: push     r12
0xbd0f: push     rbp
0xbd10: mov      rbp, rdi
0xbd13: lea      rdi, [rip + 0x14f14]
0xbd1a: push     rbx
0xbd1b: xor      ebx, ebx
0xbd1d: sub      rsp, 8
0xbd21: jmp      0xbd2d
0xbd2d
0xbd23: nop      dword ptr [rax + rax]
0xbd28: mov      rdi, qword ptr [r13 + rbx*8]
0xbd2d: mov      rsi, rbp
0xbd30: mov      r12d, ebx
0xbd33: call     0x4b30
0xbd38: test     eax, eax
0xbd3a: je       0xbd60
0xbd3c: add      rbx, 1
0xbd40: cmp      rbx, 0xa
0xbd44: jne      0xbd28
0xbd46: add      rsp, 8
0xbd4a: mov      r12d, 0xffffffff
0xbd50: pop      rbx
0xbd51: mov      eax, r12d
0xbd54: pop      rbp
0xbd55: pop      r12
0xbd57: pop      r13
0xbd59: ret
0xbd5a: nop      word ptr [rax + rax]
0xbd60: lea      eax, [rbx - 8]
0xbd63: cmp      ebx, 7
0xbd66: cmovg    r12d, eax
0xbd6a: add      rsp, 8
0xbd6e: pop      rbx
0xbd6f: pop      rbp
0xbd70: mov      eax, r12d
0xbd73: pop      r12
0xbd75: pop      r13
0xbd77: ret
0xbd78: nop      dword ptr [rax + rax]
0xbd80: endbr64
0xbd84: push     r12
0xbd86: mov      r12, rdi
0xbd89: push     rbp
0xbd8a: lea      rbp, [rip + 0x1e1e7]

```

```

0xbd91: push    rbx
0xbd92: mov     ebx, 0x11
0xbd97: jmp     0xbdac
0xbdac:
0xbd99: nop     dword ptr [rax]
0xbda0: add     ebx, 1
0xbda3: add     rbp, 0x20
0xbda7: cmp     ebx, 0x26
0xbdaa: je      0xbdd8
0xbdac: mov     rdi, qword ptr [rbp]
0xbdb0: mov     rsi, r12
0xbdb3: call    0x4b30
0xbdb8: test    eax, eax
0xbdba: jne     0xbda0
0xbdbc: movsxd  rbx, ebx
0xbdbf: lea     rax, [rip + 0x1df7a]
0xbdc6: shl     rbx, 5
0xbdca: mov     rax, qword ptr [rax + rbx + 8]
0xbdcf: pop     rbx
0xbdd0: pop     rbp
0xbdd1: pop     r12
0xbdd3: ret
0xbdd4: nop     dword ptr [rax]
0xbdd8: pop     rbx
0xbdd9: xor     eax, eax
0xbddb: pop     rbp
0xbddc: pop     r12
0xbdde: ret
0xbddf: nop
0xbde0: endbr64
0xbde4: push    r14
0xbde6: lea     rdi, [rip + 0x1479d]
0xbded: push    r13
0xbdef: push    r12
0xbdf1: push    rbp
0xbdf2: push    rbx
0xbdf3: sub     rsp, 0xb0
0xbdfa: mov     rax, qword ptr fs:[0x28]
0xbe03: mov     qword ptr [rsp + 0xa8], rax
0xbe0b: xor     eax, eax
0xbe0d: call    0x47e0
0xbe12: lea     rsi, [rsp + 0x10]
0xbe17: mov     edi, 1
0xbe1c: mov     rbp, rax
0xbe1f: call    0x4eb0
0xbe24: mov     rax, qword ptr [rsp + 0x38]
0xbe29: shr     rax, 8
0xbe2d: cmp     rax, 4

```



```

0xbe31: sete    al
0xbe34: movzx   eax, al
0xbe37: mov     dword ptr [rip + 0x1eb4b], eax
0xbe3d: test    rbp, rbp
0xbe40: je      0xc36d
0xbe46: mov     rdi, rbp
0xbe49: call    0x49b0
0xbe4e: cmp     rax, 0x3f
0xbe52: ja      0xc308
0xbe58: lea     r12, [rip + 0x14ff1]
0xbe5f: lea     rbx, [rip + 0x14d6f]
0xbe66: lea     r14, [rip + 0x14e02]
0xbe6d: lea     r13, [rip + 0x14e07]
0xbe74: nop     dword ptr [rax]
0xbe78: xor     edi, edi
0xbe7a: mov     rsi, rbp
0xbe7d: call    0x4e90
0xbe82: cmp     eax, -1
0xbe85: je      0xbf38
0xbe8b: test    eax, eax
0xbe8d: jne     0xbff8
0xbe93: mov     rdi, qword ptr [rip + 0x1e9e6]
0xbe9a: mov     esi, 1
0xbe9f: mov     r8, rbp
0xbea2: mov     rdx, r12
0xbea5: mov     rcx, qword ptr [rip + 0x1ea3c]
0xbeac: lea     r9, [rip + 0x1de5d]
0xeb3: call    0x4e60
0xeb8: mov     rsi, rbx
0xebb: mov     rdi, rbp
0xeb3e: call    0x4b30
0xebc3: test    eax, eax
0xebc5: je      0xbf98
0xebcb: mov     rdi, rbp
0xebce: call    0x9940
0xbed3: test    eax, eax
0xbed5: jne     0xbfc8
0xbedb: mov     rsi, r14
0xbede: mov     rdi, rbp
0xee1: call    0x4b30
0xee6: test    eax, eax
0xee8: je      0xbf01
0xeea: lea     rsi, [rip + 0x14d84]
0xef1: mov     rdi, rbp
0xef4: call    0x4b30
0xef9: test    eax, eax
0xefb: jne     0xc3b4
0xbf01: mov     rdi, qword ptr [rip + 0x1e978]

```

```

0xbf08: mov     rcx, qword ptr [rip + 0x1e9d9]
0xbf0f: mov     esi, 1
0xbf14: xor     eax, eax
0xbf16: mov     rdx, r13
0xbf19: lea     rbp, [rip + 0x14cba]
0xbf20: call    0x4e60
0xbf25: xor     edi, edi
0xbf27: mov     rsi, rbp
0xbf2a: call    0x4e90
0xbf2f: cmp     eax, -1
0xbf32: jne     0xbe8b
0xbf38: mov     rcx, qword ptr [rip + 0x1e9a9]
0xbf3f: mov     rdi, qword ptr [rip + 0x1e93a]
0xbf46: xor     eax, eax
0xbf48: lea     r8, [rip + 0x1ddc1]
0xbf4f: lea     rdx, [rip + 0x14ea2]
0xbf56: mov     esi, 1
0xbf5b: call    0x4e60
0xbf60: mov     rdi, qword ptr [rip + 0x1e919]
0xbf67: xor     eax, eax
0xbf69: mov     rcx, qword ptr [rip + 0x1e978]
0xbf70: lea     r8, [rip + 0x1dd89]
0xbf77: lea     rdx, [rip + 0x14ea2]
0xbf7e: mov     esi, 1
0xbf83: call    0x4e60
0xbf88: mov     edi, 1
0xbf8d: call    0x4e20
0xbf92: nop     word ptr [rax + rax]
0xbf98: mov     rcx, qword ptr [rip + 0x1e949]
0xbf9f: mov     rdi, qword ptr [rip + 0x1e8da]
0xbfa6: lea     rdx, [rip + 0x14c99]
0xbfad: mov     esi, 1
0xbfb2: lea     rbp, [rip + 0x14c21]
0xbfb9: call    0x4e60
0xbfbe: jmp     0xbe78
0xbe78:
0xbfc3: nop     dword ptr [rax + rax]
0xbfc8: mov     rcx, qword ptr [rip + 0x1e919]
0xbfcf: mov     rdi, qword ptr [rip + 0x1e8aa]
0xbfd6: lea     rdx, [rip + 0x14c7d]
0xbfdd: xor     eax, eax
0xbfdf: mov     esi, 1
0xbfe4: lea     rbp, [rip + 0x14bf6]
0xbfeb: call    0x4e60
0xbff0: jmp     0xbe78
0xbe78:
0xbff5: nop     dword ptr [rax]
0xbff8: mov     rdi, rbp

```

```

0xbffb: lea      r13, [rsp + 8]
0xc000: call     0x1d7a0
0xc005: mov      edi, 0x800
0xc00a: mov      qword ptr [rip + 0x1e93f], rax
0xc011: call     0x1d670
0xc016: mov      rdi, qword ptr [rip + 0x1de03]
0xc01d: mov      rsi, r13
0xc020: mov      qword ptr [rsp + 8], rax
0xc025: call     0x4ca0
0xc02a: xor      edx, edx
0xc02c: test     rax, rax
0xc02f: je       0xc034
0xc031: movzx    edx, byte ptr [rax]
0xc034: mov      rdi, qword ptr [rip + 0x1de25]
0xc03b: mov      rsi, r13
0xc03e: mov      byte ptr [rip + 0x1e904], dl
0xc044: call     0x4ca0
0xc049: mov      rdi, qword ptr [rip + 0x1ddf0]
0xc050: mov      rsi, r13
0xc053: mov      qword ptr [rip + 0x1e8e6], rax
0xc05a: call     0x4ca0
0xc05f: cmp      qword ptr [rip + 0x1e8d9], 0
0xc067: mov      qword ptr [rip + 0x1e8ca], rax
0xc06e: je       0xc2c0
0xc074: test     rax, rax
0xc077: je       0xc2c0
0xc07d: mov      rdi, qword ptr [rip + 0x1dcbc]
0xc084: mov      rsi, r13
0xc087: call     0x4ca0
0xc08c: mov      rdi, qword ptr [rip + 0x1dced]
0xc093: mov      rsi, r13
0xc096: mov      qword ptr [rip + 0x1dcab], rax
0xc09d: call     0x4ca0
0xc0a2: mov      rdi, qword ptr [rip + 0x1dcb7]
0xc0a9: mov      rsi, r13
0xc0ac: mov      qword ptr [rip + 0x1dcd5], rax
0xc0b3: call     0x4ca0
0xc0b8: cmp      qword ptr [rip + 0x1dc88], 0
0xc0c0: mov      qword ptr [rip + 0x1dca1], rax
0xc0c7: je       0xc352
0xc0cd: mov      rdi, qword ptr [rip + 0x1ddac]
0xc0d4: mov      rsi, r13
0xc0d7: call     0x4ca0
0xc0dc: xor      edx, edx
0xc0de: mov      qword ptr [rip + 0x1dda3], rax
0xc0e5: test     rax, rax
0xc0e8: je       0xc0ff
0xc0ea: mov      rdi, qword ptr [rip + 0x1ddcf]

```

```

0xc0f1: mov     qword ptr [rip + 0x1dc70], rax
0xc0f8: call    0x4c90
0xc0fd: mov     edx, eax
0xc0ff: mov     rdi, qword ptr [rip + 0x1dd9a]
0xc106: mov     dword ptr [rip + 0x1ddc4], edx
0xc10c: call    0x4c50
0xc111: mov     dword ptr [rip + 0x1dd99], eax
0xc117: test    eax, eax
0xc119: js      0xc13c
0xc11b: mov     qword ptr [rip + 0x1dc62], 0
0xc126: mov     qword ptr [rip + 0x1dc37], 0
0xc131: mov     qword ptr [rip + 0x1dc0c], 0
0xc13c: mov     rdi, qword ptr [rip + 0x1dd9d]
0xc143: call    0x4c50
0xc148: mov     rdi, qword ptr [rip + 0x1ddb1]
0xc14f: mov     dword ptr [rip + 0x1dd9b], eax
0xc155: call    0x4c50
0xc15a: mov     rdi, qword ptr [rip + 0x1dc3f]
0xc161: mov     rsi, r13
0xc164: mov     dword ptr [rip + 0x1dda6], eax
0xc16a: call    0x4ca0
0xc16f: mov     rdi, qword ptr [rip + 0x1dc4a]
0xc176: mov     rsi, r13
0xc179: mov     qword ptr [rip + 0x1dc28], rax
0xc180: call    0x4ca0
0xc185: cmp     qword ptr [rip + 0x1dc1b], 0
0xc18d: mov     qword ptr [rip + 0x1dc34], rax
0xc194: je      0xc2a0
0xc19a: test    rax, rax
0xc19d: je      0xc2a0
0xc1a3: mov     rdi, qword ptr [rip + 0x1dc36]
0xc1aa: mov     rsi, r13
0xc1ad: lea     rbp, [rip + 0x1ddac]
0xc1b4: lea     rbx, [rbp - 0x220]
0xc1bb: call    0x4ca0
0xc1c0: mov     rdi, qword ptr [rip + 0x1dc39]
0xc1c7: mov     rsi, r13
0xc1ca: lea     r12, [rbx + 0x4c0]
0xc1d1: mov     qword ptr [rip + 0x1dc10], rax
0xc1d8: call    0x4ca0
0xc1dd: mov     rdi, qword ptr [rip + 0x1dd3c]
0xc1e4: mov     rsi, r13
0xc1e7: mov     qword ptr [rip + 0x1dc1a], rax
0xc1ee: call    0x4ca0
0xc1f3: mov     rdi, qword ptr [rip + 0x1dd46]
0xc1fa: mov     rsi, r13
0xc1fd: mov     qword ptr [rip + 0x1dd24], rax
0xc204: call    0x4ca0

```

```

0xc209: mov     qword ptr [rip + 0x1dd38], rax
0xc210: mov     rdi, qword ptr [rbp]
0xc214: mov     rsi, r13
0xc217: add     rbp, 0x20
0xc21b: call    0x4ca0
0xc220: mov     qword ptr [rbp - 0x18], rax
0xc224: cmp     rbp, r12
0xc227: jne     0xc210
0xc229: xor     ebp, ebp
0xc22b: lea     r13, [rip + 0x161b3]
0xc232: lea     r14, [rip + 0x14c57]
0xc239: jmp     0xc249
0xc249
0xc23b: nop     dword ptr [rax + rax]
0xc240: add     rbx, 0x20
0xc244: cmp     rbx, r12
0xc247: je      0xc268
0xc249: cmp     qword ptr [rbx + 8], 0
0xc24e: jne     0xc240
0xc250: mov     eax, dword ptr [rbx + 0x14]
0xc253: test    eax, eax
0xc255: jne     0xc2e0
0xc25b: mov     qword ptr [rbx + 8], r13
0xc25f: add     rbx, 0x20
0xc263: cmp     rbx, r12
0xc266: jne     0xc249
0xc268: test    ebp, ebp
0xc26a: jne     0xc390
0xc270: mov     rax, qword ptr [rsp + 0xa8]
0xc278: sub     rax, qword ptr fs:[0x28]
0xc281: jne     0xc3be
0xc287: add     rsp, 0xb0
0xc28e: pop     rbx
0xc28f: pop     rbp
0xc290: pop     r12
0xc292: pop     r13
0xc294: pop     r14
0xc296: ret
0xc297: nop     word ptr [rax + rax]
0xc2a0: mov     qword ptr [rip + 0x1dafd], 0
0xc2ab: mov     qword ptr [rip + 0x1db12], 0
0xc2b6: jmp     0xc1a3
0xc1a3
0xc2bb: nop     dword ptr [rax + rax]
0xc2c0: mov     qword ptr [rip + 0x1e66d], 0
0xc2cb: mov     qword ptr [rip + 0x1e66a], 0
0xc2d6: jmp     0xc07d
0xc07d

```

```

0xc2db: nop        dword ptr [rax + rax]
0xc2e0: mov         r8, qword ptr [rbx]
0xc2e3: mov         rcx, qword ptr [rip + 0x1e5fe]
0xc2ea: mov         rdx, r14
0xc2ed: xor         eax, eax
0xc2ef: mov         rdi, qword ptr [rip + 0x1e58a]
0xc2f6: mov         esi, 1
0xc2fb: add         ebp, 1
0xc2fe: call        0x4e60
0xc303: jmp         0xc240
0xc240:
0xc308: mov         rcx, qword ptr [rip + 0x1e5d9]
0xc30f: mov         rdi, qword ptr [rip + 0x1e56a]
0xc316: lea         rdx, [rip + 0x14aab]
0xc31d: xor         eax, eax
0xc31f: mov         esi, 1
0xc324: call        0x4e60
0xc329: mov         rcx, qword ptr [rip + 0x1e550]
0xc330: mov         edx, 0x11
0xc335: mov         esi, 1
0xc33a: lea         rdi, [rip + 0x148f3]
0xc341: lea         rbp, [rip + 0x1d9d1]
0xc348: call        0x4e40
0xc34d: jmp         0xbe58
0xbe58:
0xc352: mov         qword ptr [rip + 0x1da2b], 0
0xc35d: mov         qword ptr [rip + 0x1da00], 0
0xc368: jmp         0xc0cd
0xc0cd:
0xc36d: mov         rcx, qword ptr [rip + 0x1e574]
0xc374: mov         rdi, qword ptr [rip + 0x1e505]
0xc37b: lea         rdx, [rip + 0x14a16]
0xc382: xor         eax, eax
0xc384: mov         esi, 1
0xc389: call        0x4e60
0xc38e: jmp         0xc329
0xc329:
0xc390: mov         rcx, qword ptr [rip + 0x1e551]
0xc397: mov         rdi, qword ptr [rip + 0x1e4e2]
0xc39e: mov         r8d, ebp
0xc3a1: xor         eax, eax
0xc3a3: lea         rdx, [rip + 0x14b16]
0xc3aa: mov         esi, 1
0xc3af: call        0x4e60
0xc3b4: mov         edi, 1
0xc3b9: call        0x4e20
0xc3be: call        0x49d0
0xc3c3: nop        word ptr cs:[rax + rax]

```

```

0xc3ce: nop
0xc3d0: endbr64
0xc3d4: push    rbx
0xc3d5: mov     ebx, edi
0xc3d7: xor     edi, edi
0xc3d9: call    0x48f0
0xc3de: test    eax, eax
0xc3e0: je      0xc453
0xc3e2: mov     edi, 1
0xc3e7: call    0x48f0
0xc3ec: test    eax, eax
0xc3ee: je      0xc453
0xc3f0: mov     edi, 1
0xc3f5: call    0x4cd0
0xc3fa: mov     qword ptr [rip + 0x1e58f], rax
0xc401: test    rax, rax
0xc404: je      0xc47e
0xc406: lea     rsi, [rip + 0x1efb3]
0xc40d: mov     edi, 1
0xc412: call    0x4d60
0xc417: mov     edi, 0x40
0xc41c: call    0x1d670
0xc421: mov     dword ptr [rip + 0x1efd5], ebx
0xc427: pop     rbx
0xc428: mov     qword ptr [rip + 0x1ef31], rax
0xc42f: mov     qword ptr [rip + 0x1e52a], rax
0xc436: xor     eax, eax
0xc438: mov     qword ptr [rip + 0x1e52d], 0
0xc443: mov     qword ptr [rip + 0x1e51a], 0
0xc44e: jmp     0xbde0
0xbde0
0xc453: mov     rdi, qword ptr [rip + 0x1e426]
0xc45a: mov     rcx, qword ptr [rip + 0x1e487]
0xc461: lea     rdx, [rip + 0x14a88]
0xc468: xor     eax, eax
0xc46a: mov     esi, 1
0xc46f: call    0x4e60
0xc474: mov     edi, 1
0xc479: call    0x4e20
0xc47e: mov     rdi, qword ptr [rip + 0x1e3fb]
0xc485: mov     rcx, qword ptr [rip + 0x1e45c]
0xc48c: mov     esi, 1
0xc491: lea     rdx, [rip + 0x147f8]
0xc498: call    0x4e60
0xc49d: mov     edi, 1
0xc4a2: call    0x4e20
0xc4a7: nop     word ptr [rax + rax]
0xc4b0: endbr64

```

```

0xc4b4: push    r12
0xc4b6: push    rbp
0xc4b7: mov     rbp, rdi
0xc4ba: mov     rdi, qword ptr [rip + 0x1e48f]
0xc4c1: push    rbx
0xc4c2: call    0x9940
0xc4c7: test    eax, eax
0xc4c9: jne     0xc4d0
0xc4cb: pop     rbx
0xc4cc: pop     rbp
0xc4cd: pop     r12
0xc4cf: ret
0xc4d0: mov     rdi, rbp
0xc4d3: call    0x49b0
0xc4d8: lea     rdi, [rax + 0x81]
0xc4df: mov     rbx, rax
0xc4e2: call    0x1d670
0xc4e7: mov     rdi, rbp
0xc4ea: mov     r12, rax
0xc4ed: call    0x1d7a0
0xc4f2: mov     rsi, rbx
0xc4f5: add     ebx, 0x7f
0xc4f8: mov     rbp, rax
0xc4fb: mov     rdi, rax
0xc4fe: call    0x184d0
0xc503: push    7
0xc505: lea     rcx, [rip + 0x147a2]
0xc50c: xor     eax, eax
0xc50e: push    rbp
0xc50f: mov     rdx, -1
0xc516: mov     esi, 1
0xc51b: mov     rdi, r12
0xc51e: lea     r9, [rip + 0x14040]
0xc525: mov     r8d, 0x1b
0xc52b: call    0x4f10
0xc530: mov     eax, dword ptr [rip + 0x1e462]
0xc536: pop     rdx
0xc537: pop     rcx
0xc538: add     eax, eax
0xc53a: cmp     ebx, eax
0xc53c: jl      0xc555
0xc53e: cdqe
0xc540: mov     byte ptr [r12 + rax], 7
0xc545: mov     eax, dword ptr [rip + 0x1e44d]
0xc54b: add     eax, eax
0xc54d: cdqe
0xc54f: mov     byte ptr [r12 + rax + 1], 0
0xc555: mov     rdi, r12

```



```

0xc558: call    0x49b0
0xc55d: mov     rsi, r12
0xc560: mov     edi, 1
0xc565: mov     rdx, rax
0xc568: call    0x18da0
0xc56d: mov     rdi, rbp
0xc570: call    0x18d10
0xc575: mov     rdi, r12
0xc578: call    0x18d10
0xc57d: mov     rdi, qword ptr [rip + 0x1e2bc]
0xc584: pop     rbx
0xc585: pop     rbp
0xc586: pop     r12
0xc588: jmp     0x4c40
0x4c40
0xc58d: nop     dword ptr [rax]
0xc590: endbr64
0xc594: sub     rsp, 8
0xc598: mov     edi, 0x18
0xc59d: call    0x1d670
0xc5a2: mov     qword ptr [rax], 0
0xc5a9: mov     qword ptr [rax + 8], 0
0xc5b1: add     rsp, 8
0xc5b5: ret
0xc5b6: nop     word ptr cs:[rax + rax]
0xc5c0: endbr64
0xc5c4: test    rdi, rdi
0xc5c7: je      0xc5d0
0xc5c9: jmp     0x18d10
0x18d10
0xc5ce: nop
0xc5d0: ret
0xc5d1: nop     word ptr cs:[rax + rax]
0xc5dc: nop     dword ptr [rax]
0xc5e0: endbr64
0xc5e4: mov     dword ptr [rdi], esi
0xc5e6: mov     dword ptr [rdi + 4], edx
0xc5e9: mov     dword ptr [rdi + 8], ecx
0xc5ec: mov     dword ptr [rdi + 0xc], r8d
0xc5f0: ret
0xc5f1: nop     word ptr cs:[rax + rax]
0xc5fc: nop     dword ptr [rax]
0xc600: endbr64
0xc604: mov     rax, rdi
0xc607: mov     rdi, rsi
0xc60a: mov     esi, edx
0xc60c: mov     ecx, dword ptr [rax + 0x10]
0xc60f: lea     r8d, [rcx + rdx]

```

```

0xc613: mov     edx, dword ptr [rax + 0xc]
0xc616: mov     dword ptr [rax + 0x10], r8d
0xc61a: cmp     edx, ecx
0xc61c: jle     0xc640
0xc61e: mov     r9d, dword ptr [rax + 8]
0xc622: cmp     dword ptr [rax + 0x14], r9d
0xc626: jge     0xc640
0xc628: cmp     r8d, edx
0xc62b: jle     0xc631
0xc62d: sub     edx, ecx
0xc62f: mov     esi, edx
0xc631: jmp     0xa1a0
0xa1a0
0xc636: nop     word ptr cs:[rax + rax]
0xc640: xor     eax, eax
0xc642: ret
0xc643: nop     word ptr cs:[rax + rax]
0xc64e: nop
0xc650: endbr64
0xc654: mov     ecx, dword ptr [rdi + 0x10]
0xc657: lea     edx, [rcx + 1]
0xc65a: mov     dword ptr [rdi + 0x10], edx
0xc65d: cmp     edx, dword ptr [rdi + 0xc]
0xc660: jg      0xc66a
0xc662: mov     ecx, dword ptr [rdi + 8]
0xc665: cmp     dword ptr [rdi + 0x14], ecx
0xc668: jl      0xc670
0xc66a: xor     eax, eax
0xc66c: ret
0xc66d: nop     dword ptr [rax]
0xc670: mov     edi, esi
0xc672: jmp     0xa230
0xa230
0xc677: nop     word ptr [rax + rax]
0xc680: endbr64
0xc684: mov     rax, rdi
0xc687: mov     edi, esi
0xc689: mov     dword ptr [rax + 0x10], edx
0xc68c: add     edx, dword ptr [rax]
0xc68e: mov     dword ptr [rax + 0x14], esi
0xc691: add     edi, dword ptr [rax + 4]
0xc694: mov     esi, edx
0xc696: jmp     0xa4b0
0xa4b0
0xc69b: nop     dword ptr [rax + rax]
0xc6a0: endbr64
0xc6a4: mov     eax, dword ptr [rdi]
0xc6a6: ret

```

```

0xc6a7: nop      word ptr [rax + rax]
0xc6b0: endbr64
0xc6b4: mov      eax, dword ptr [rdi + 4]
0xc6b7: ret
0xc6b8: nop      dword ptr [rax + rax]
0xc6c0: endbr64
0xc6c4: mov      eax, dword ptr [rdi + 8]
0xc6c7: ret
0xc6c8: nop      dword ptr [rax + rax]
0xc6d0: endbr64
0xc6d4: mov      eax, dword ptr [rdi + 0xc]
0xc6d7: ret
0xc6d8: nop      dword ptr [rax + rax]
0xc6e0: push     r14
0xc6e2: push     r13
0xc6e4: push     r12
0xc6e6: mov      r12, qword ptr [rip + 0x1ed1b]
0xc6ed: push     rbp
0xc6ee: push     rbx
0xc6ef: mov      r13, qword ptr [r12 + 0x20]
0xc6f4: mov      rbx, qword ptr [r12 + 0x18]
0xc6f9: cmp      r13, rbx
0xc6fc: je       0xc761
0xc6fe: mov      ebp, edi
0xc700: jbe      0xc770
0xc702: test     bpl, 1
0xc706: je       0xc75c
0xc708: mov      rdi, qword ptr [r12 + 0x60]
0xc70d: test     rdi, rdi
0xc710: je       0xc71e
0xc712: call     0x18d10
0xc717: mov      r12, qword ptr [rip + 0x1ecea]
0xc71e: mov      r14, r13
0xc721: sub      r14, rbx
0xc724: lea      rdi, [r14 + 1]
0xc728: call     0x1d670
0xc72d: mov      rdx, r14
0xc730: mov      qword ptr [r12 + 0x60], rax
0xc735: mov      rax, qword ptr [rip + 0x1eccc]
0xc73c: mov      rsi, qword ptr [rax + 0x58]
0xc740: mov      rdi, qword ptr [rax + 0x60]
0xc744: add      rsi, rbx
0xc747: call     0x4bb0
0xc74c: mov      rax, qword ptr [rip + 0x1ecb5]
0xc753: mov      rax, qword ptr [rax + 0x60]
0xc757: mov      byte ptr [rax + r14], 0
0xc75c: and      ebp, 2
0xc75f: jne      0xc788

```

```

0xc761: pop    rbx
0xc762: pop    rbp
0xc763: pop    r12
0xc765: pop    r13
0xc767: pop    r14
0xc769: ret
0xc76a: nop    word ptr [rax + rax]
0xc770: mov    rax, rbx
0xc773: mov    rbx, r13
0xc776: mov    r13, rax
0xc779: test   bpl, 1
0xc77d: je     0xc75c
0xc77f: jmp    0xc708
0xc708
0xc781: nop    dword ptr [rax]
0xc788: mov    rax, qword ptr [rip + 0x1ec79]
0xc78f: mov    rdx, rbx
0xc792: sub    rdx, r13
0xc795: add    qword ptr [rax + 0x48], rdx
0xc799: mov    rdi, qword ptr [rax + 0x58]
0xc79d: add    qword ptr [rax + 0x38], rdx
0xc7a1: mov    qword ptr [rax + 0x18], rbx
0xc7a5: lea    rsi, [rdi + r13]
0xc7a9: add    rdi, rbx
0xc7ac: mov    qword ptr [rax + 0x20], rbx
0xc7b0: call   0x48b0
0xc7b5: mov    rbx, qword ptr [rip + 0x1ec4c]
0xc7bc: mov    rax, qword ptr [rbx + 0x38]
0xc7c0: mov    rdi, qword ptr [rbx + 0x58]
0xc7c4: lea    rsi, [rax + 1]
0xc7c8: mov    qword ptr [rbx + 0x50], rsi
0xc7cc: call   0x1d690
0xc7d1: mov    qword ptr [rbx + 0x58], rax
0xc7d5: pop    rbx
0xc7d6: pop    rbp
0xc7d7: pop    r12
0xc7d9: pop    r13
0xc7db: pop    r14
0xc7dd: ret
0xc7de: nop
0xc7e0: endbr64
0xc7e4: test   rdi, rdi
0xc7e7: je     0xc818
0xc7e9: push   rbp
0xc7ea: mov    rbp, rdi
0xc7ed: mov    rdi, qword ptr [rdi + 0x58]
0xc7f1: test   rdi, rdi
0xc7f4: je     0xc7fb

```

```

0xc7f6: call    0x18d10
0xc7fb: mov     rdi, qword ptr [rbp + 0x60]
0xc7ff: test    rdi, rdi
0xc802: je      0xc809
0xc804: call    0x18d10
0xc809: mov     rdi, rbp
0xc80c: pop     rbp
0xc80d: jmp     0x18d10
0x18d10
0xc812: nop     word ptr [rax + rax]
0xc818: ret
0xc819: nop     dword ptr [rax]
0xc820: endbr64
0xc824: mov     rax, qword ptr [rip + 0x1ebdd]
0xc82b: push    rbp
0xc82c: mov     rdi, qword ptr [rax + 0x68]
0xc830: xor     eax, eax
0xc832: call    0x1aac0
0xc837: mov     rax, qword ptr [rip + 0x1ebca]
0xc83e: mov     rdi, qword ptr [rax]
0xc841: call    0xc5c0
0xc846: mov     rbp, qword ptr [rip + 0x1ebbb]
0xc84d: test    rbp, rbp
0xc850: je      0xc876
0xc852: mov     rdi, qword ptr [rbp + 0x58]
0xc856: test    rdi, rdi
0xc859: je      0xc860
0xc85b: call    0x18d10
0xc860: mov     rdi, qword ptr [rbp + 0x60]
0xc864: test    rdi, rdi
0xc867: je      0xc86e
0xc869: call    0x18d10
0xc86e: mov     rdi, rbp
0xc871: call    0x18d10
0xc876: mov     qword ptr [rip + 0x1eb87], 0
0xc881: pop     rbp
0xc882: ret
0xc883: nop     word ptr cs:[rax + rax]
0xc88e: nop
0xc890: endbr64
0xc894: mov     rax, qword ptr [rip + 0x1eb6d]
0xc89b: movsxd  rcx, edi
0xc89e: mov     edx, esi
0xc8a0: mov     r8, rcx
0xc8a3: mov     qword ptr [rax + 0x28], rcx
0xc8a7: movsxd  rcx, esi
0xc8aa: mov     rdi, qword ptr [rax]
0xc8ad: xor     esi, esi

```

```

0xc8af: mov     qword ptr [rax + 0x30], rcx
0xc8b3: mov     ecx, 1
0xc8b8: jmp     0xc5e0
0xc5e0:
0xc8bd: nop     dword ptr [rax]
0xc8c0: endbr64
0xc8c4: push    r12
0xc8c6: mov     edi, 0x70
0xc8cb: push    rbx
0xc8cc: sub     rsp, 8
0xc8d0: call    0x1d670
0xc8d5: mov     rbx, qword ptr [rip + 0x1eb2c]
0xc8dc: mov     r12, rax
0xc8df: movdqu  xmm0, xmmword ptr [rbx]
0xc8e3: movups  xmmword ptr [rax], xmm0
0xc8e6: movdqu  xmm1, xmmword ptr [rbx + 0x10]
0xc8eb: movups  xmmword ptr [rax + 0x10], xmm1
0xc8ef: movdqu  xmm2, xmmword ptr [rbx + 0x20]
0xc8f4: movups  xmmword ptr [rax + 0x20], xmm2
0xc8f8: movdqu  xmm3, xmmword ptr [rbx + 0x30]
0xc8fd: movups  xmmword ptr [rax + 0x30], xmm3
0xc901: movdqu  xmm4, xmmword ptr [rbx + 0x40]
0xc906: movups  xmmword ptr [rax + 0x40], xmm4
0xc90a: movdqu  xmm5, xmmword ptr [rbx + 0x50]
0xc90f: movups  xmmword ptr [rax + 0x50], xmm5
0xc913: movdqu  xmm6, xmmword ptr [rbx + 0x60]
0xc918: cmp     qword ptr [rax + 0x58], 0
0xc91d: movups  xmmword ptr [rax + 0x60], xmm6
0xc921: je      0xc94a
0xc923: mov     rdi, qword ptr [rax + 0x50]
0xc927: call    0x1d670
0xc92c: mov     rdx, qword ptr [r12 + 0x50]
0xc931: mov     qword ptr [rbx + 0x58], rax
0xc935: mov     rax, qword ptr [rip + 0x1eacc]
0xc93c: mov     rsi, qword ptr [r12 + 0x58]
0xc941: mov     rdi, qword ptr [rax + 0x58]
0xc945: call    0x4bb0
0xc94a: mov     rdi, qword ptr [r12 + 0x60]
0xc94f: test    rdi, rdi
0xc952: je      0xc964
0xc954: mov     rbx, qword ptr [rip + 0x1eaad]
0xc95b: call    0x1d7a0
0xc960: mov     qword ptr [rbx + 0x60], rax
0xc964: add     rsp, 8
0xc968: mov     rax, r12
0xc96b: pop     rbx
0xc96c: pop     r12
0xc96e: ret

```

```

0xc96f: nop
0xc970: endbr64
0xc974: test    rdi, rdi
0xc977: je      0xc9f0
0xc979: push    r13
0xc97b: push    r12
0xc97d: push    rbp
0xc97e: push    rbx
0xc97f: mov     rbx, rdi
0xc982: sub     rsp, 8
0xc986: mov     rbp, qword ptr [rip + 0x1ea7b]
0xc98d: mov     rdi, qword ptr [rbp + 0x58]
0xc991: mov     r13, qword ptr [rbp + 0x28]
0xc995: mov     r12, qword ptr [rbp + 0x30]
0xc999: test    rdi, rdi
0xc99c: je      0xc9a3
0xc99e: call    0x18d10
0xc9a3: mov     rdi, qword ptr [rbp + 0x60]
0xc9a7: test    rdi, rdi
0xc9aa: je      0xc9b1
0xc9ac: call    0x18d10
0xc9b1: mov     rdi, rbp
0xc9b4: call    0x18d10
0xc9b9: movsxd  rax, r13d
0xc9bc: mov     edx, r12d
0xc9bf: movsxd  r12, r12d
0xc9c2: mov     qword ptr [rbx + 0x28], rax
0xc9c6: mov     rdi, qword ptr [rbx]
0xc9c9: mov     r8d, r13d
0xc9cc: mov     ecx, 1
0xc9d1: mov     qword ptr [rbx + 0x30], r12
0xc9d5: xor     esi, esi
0xc9d7: mov     qword ptr [rip + 0x1ea2a], rbx
0xc9de: add     rsp, 8
0xc9e2: pop     rbx
0xc9e3: pop     rbp
0xc9e4: pop     r12
0xc9e6: pop     r13
0xc9e8: jmp     0xc5e0
0xc5e0
0xc9ed: nop     dword ptr [rax]
0xc9f0: ret
0xc9f1: nop     word ptr cs:[rax + rax]
0xc9fc: nop     dword ptr [rax]
0xca00: endbr64
0xca04: mov     rdx, qword ptr [rip + 0x1e9fd]
0xca0b: mov     rax, qword ptr [rdx + 0x18]
0xca0f: sub     rax, qword ptr [rdx + 0x40]

```

```

0xca13: ret
0xca14: nop      word ptr cs:[rax + rax]
0xca1f: nop
0xca20: endbr64
0xca24: mov      rax, qword ptr [rip + 0x1e9dd]
0xca2b: mov      r8d, dword ptr [rax + 8]
0xca2f: mov      dword ptr [rax + 8], edi
0xca32: mov      eax, r8d
0xca35: ret
0xca36: nop      word ptr cs:[rax + rax]
0xca40: endbr64
0xca44: mov      rax, qword ptr [rip + 0x1e9bd]
0xca4b: cmp      qword ptr [rax + 0x48], 0
0xca50: sete     al
0xca53: movzx    eax, al
0xca56: ret
0xca57: nop      word ptr [rax + rax]
0xca60: endbr64
0xca64: mov      rax, qword ptr [rip + 0x1e99d]
0xca6b: mov      rdx, qword ptr [rax + 0x18]
0xca6f: mov      qword ptr [rax + 0x20], rdx
0xca73: ret
0xca74: nop      word ptr cs:[rax + rax]
0xca7f: nop
0xca80: endbr64
0xca84: sub      rsp, 8
0xca88: xor      eax, eax
0xca8a: mov      edi, 3
0xca8f: call     0xc6e0
0xca94: mov      rax, qword ptr [rip + 0x1e96d]
0xca9b: mov      dword ptr [rax + 0x10], 0x18
0xcaa2: add      rsp, 8
0xcaa6: ret
0xcaa7: nop      word ptr [rax + rax]
0xcab0: endbr64
0xcab4: sub      rsp, 8
0xcab8: xor      eax, eax
0xcaba: mov      edi, 1
0xcabf: call     0xc6e0
0xcac4: mov      rax, qword ptr [rip + 0x1e93d]
0xcacb: mov      dword ptr [rax + 0x10], 0x19
0xcad2: add      rsp, 8
0xcad6: ret
0xcad7: nop      word ptr [rax + rax]
0xcae0: endbr64
0xcae4: mov      rax, qword ptr [rip + 0x1e91d]
0xcaeb: mov      rdx, qword ptr [rax + 0x18]
0xcaef: mov      rcx, qword ptr [rax + 0x20]

```



```

0xcaf3: movsxd   rdx, edx
0xcaf6: mov      qword ptr [rax + 0x18], rcx
0xcafa: mov      qword ptr [rax + 0x20], rdx
0xcafe: ret
0xcaff: nop
0xcb00: endbr64
0xcb04: mov      rax, qword ptr [rip + 0x1e8fd]
0xcb0b: mov      rdx, qword ptr [rax + 0x18]
0xcb0f: cmp      rdx, qword ptr [rax + 0x40]
0xcb13: jbe      0xcb24
0xcb15: sub      rdx, 1
0xcb19: mov      dword ptr [rax + 0x10], 1
0xcb20: mov      qword ptr [rax + 0x18], rdx
0xcb24: ret
0xcb25: nop      word ptr cs:[rax + rax]
0xcb30: endbr64
0xcb34: mov      rax, qword ptr [rip + 0x1e8cd]
0xcb3b: mov      rdx, qword ptr [rax + 0x18]
0xcb3f: cmp      rdx, qword ptr [rax + 0x38]
0xcb43: jae      0xcb54
0xcb45: add      rdx, 1
0xcb49: mov      dword ptr [rax + 0x10], 2
0xcb50: mov      qword ptr [rax + 0x18], rdx
0xcb54: ret
0xcb55: nop      word ptr cs:[rax + rax]
0xcb60: endbr64
0xcb64: mov      rdx, qword ptr [rip + 0x1e89d]
0xcb6b: mov      rax, qword ptr [rdx + 0x18]
0xcb6f: mov      rdi, qword ptr [rdx + 0x40]
0xcb73: cmp      rax, rdi
0xcb76: jbe      0xcc20
0xcb7c: mov      r8, qword ptr [rdx + 0x58]
0xcb80: jmp      0xcba9
0xcba9:
0xcb82: nop      word ptr [rax + rax]
0xcb88: lea      ecx, [rsi - 0x30]
0xcb8b: cmp      cl, 9
0xcb8e: jbe      0xcbbf
0xcb90: sub      esi, 0x24
0xcb93: cmp      sil, 1
0xcb97: jbe      0xcbbf
0xcb99: mov      qword ptr [rdx + 0x18], rax
0xcb9d: mov      dword ptr [rdx + 0x10], 1
0xcba4: cmp      rdi, rax
0xcba7: jae      0xcbef
0xcba9: sub      rax, 1
0xcbad: movzx    esi, byte ptr [r8 + rax]
0xcbb2: mov      ecx, esi

```

```

0xcbb4: and     ecx, 0xffffffff
0xcbb7: sub     ecx, 0x41
0xcbb8: cmp     cl, 0x19
0xcbbd: ja      0xcb88
0xcbbf: mov     rax, qword ptr [rdx + 0x18]
0xcbc3: cmp     rdi, rax
0xcbc6: jae     0xcbef
0xcbc8: sub     rax, 1
0xcbcc: movzx   esi, byte ptr [r8 + rax]
0xcbd1: mov     ecx, esi
0xcbd3: and     ecx, 0xffffffff
0xcbd6: sub     ecx, 0x41
0xcbd9: cmp     cl, 0x19
0xcbd9: jbe     0xcc00
0xcbde: lea     ecx, [rsi - 0x30]
0xcbe1: cmp     cl, 9
0xcbe4: jbe     0xcc00
0xcbe6: sub     esi, 0x24
0xcbe9: cmp     sil, 1
0xcbed: jbe     0xcc00
0xcbef: mov     dword ptr [rdx + 0x10], 3
0xcbf6: ret
0xcbf7: nop     word ptr [rax + rax]
0xcc00: mov     qword ptr [rdx + 0x18], rax
0xcc04: mov     dword ptr [rdx + 0x10], 1
0xcc0b: cmp     rdi, rax
0xcc0e: jb      0xcbc8
0xcc10: mov     dword ptr [rdx + 0x10], 3
0xcc17: ret
0xcc18: nop     dword ptr [rax + rax]
0xcc20: ret
0xcc21: nop     word ptr cs:[rax + rax]
0xcc2c: nop     dword ptr [rax]
0xcc30: endbr64
0xcc34: mov     rdx, qword ptr [rip + 0x1e7cd]
0xcc3b: mov     rax, qword ptr [rdx + 0x18]
0xcc3f: mov     rdi, qword ptr [rdx + 0x38]
0xcc43: cmp     rax, rdi
0xcc46: jae     0xcce8
0xcc4c: mov     r8, qword ptr [rdx + 0x58]
0xcc50: jmp     0xcc7d
0xcc7d:
0xcc52: nop     word ptr [rax + rax]
0xcc58: lea     ecx, [rsi - 0x30]
0xcc5b: cmp     cl, 9
0xcc5e: jbe     0xcc8f
0xcc60: sub     esi, 0x24
0xcc63: cmp     sil, 1

```

```

0xcc67: jbe      0xcc8f
0xcc69: add      rax, 1
0xcc6d: mov      dword ptr [rdx + 0x10], 2
0xcc74: mov      qword ptr [rdx + 0x18], rax
0xcc78: cmp      rdi, rax
0xcc7b: je       0xccbb
0xcc7d: movzx    esi, byte ptr [r8 + rax]
0xcc82: mov      ecx, esi
0xcc84: and      ecx, 0xffffffff
0xcc87: sub      ecx, 0x41
0xcc8a: cmp      cl, 0x19
0xcc8d: ja       0xcc58
0xcc8f: mov      rax, qword ptr [rdx + 0x18]
0xcc93: cmp      rdi, rax
0xcc96: jbe      0xccbb
0xcc98: movzx    esi, byte ptr [r8 + rax]
0xcc9d: mov      ecx, esi
0xcc9f: and      ecx, 0xffffffff
0xcca2: sub      ecx, 0x41
0xcca5: cmp      cl, 0x19
0xcca8: jbe      0xccc8
0xccaa: lea      ecx, [rsi - 0x30]
0xccad: cmp      cl, 9
0xccb0: jbe      0xccc8
0xccb2: sub      esi, 0x24
0xccb5: cmp      sil, 1
0xccb9: jbe      0xccc8
0xccbb: mov      dword ptr [rdx + 0x10], 4
0xccc2: ret
0xccc3: nop      dword ptr [rax + rax]
0xccc8: add      rax, 1
0xcccc: mov      dword ptr [rdx + 0x10], 2
0xcd03: mov      qword ptr [rdx + 0x18], rax
0xcd07: cmp      rdi, rax
0xcd0a: jne      0xcc98
0xcd0c: mov      dword ptr [rdx + 0x10], 4
0xcce3: ret
0xcce4: nop      dword ptr [rax]
0xcce8: ret
0xcce9: nop      dword ptr [rax]
0xccf0: endbr64
0xccf4: mov      rax, qword ptr [rip + 0x1e70d]
0xccfb: mov      rdx, qword ptr [rax + 0x40]
0xccff: mov      dword ptr [rax + 0x10], 5
0xcd06: mov      qword ptr [rax + 0x18], rdx
0xcd0a: ret
0xcd0b: nop      dword ptr [rax + rax]
0xcd10: endbr64

```

```

0xcd14: mov     rax, qword ptr [rip + 0x1e6ed]
0xcd1b: mov     rdx, qword ptr [rax + 0x38]
0xcd1f: mov     dword ptr [rax + 0x10], 6
0xcd26: mov     qword ptr [rax + 0x18], rdx
0xcd2a: ret
0xcd2b: nop     dword ptr [rax + rax]
0xcd30: endbr64
0xcd34: push    rbp
0xcd35: push    rbx
0xcd36: mov     ebx, edi
0xcd38: sub     rsp, 8
0xcd3c: call    0x4f00
0xcd41: movsxd  rdx, ebx
0xcd44: mov     rax, qword ptr [rax]
0xcd47: test    byte ptr [rax + rdx*2 + 1], 0x40
0xcd4c: je      0xcdab
0xcd4e: mov     rbp, qword ptr [rip + 0x1e6b3]
0xcd55: mov     rsi, qword ptr [rbp + 0x38]
0xcd59: lea     rdx, [rsi + 1]
0xcd5d: cmp     rdx, qword ptr [rbp + 0x50]
0xcd61: jae     0xcdb8
0xcd63: mov     rax, qword ptr [rbp + 0x18]
0xcd67: mov     rcx, qword ptr [rbp + 0x58]
0xcd6b: sub     rdx, rax
0xcd6e: lea     rsi, [rcx + rax]
0xcd72: lea     rdi, [rcx + rax + 1]
0xcd77: call    0x4d40
0xcd7c: mov     rax, qword ptr [rip + 0x1e685]
0xcd83: mov     rdx, qword ptr [rax + 0x58]
0xcd87: mov     rax, qword ptr [rax + 0x18]
0xcd8b: mov     byte ptr [rdx + rax], bl
0xcd8e: mov     rax, qword ptr [rip + 0x1e673]
0xcd95: add     qword ptr [rax + 0x18], 1
0xcd9a: add     qword ptr [rax + 0x38], 1
0xcd9f: add     qword ptr [rax + 0x48], 1
0xcda4: mov     dword ptr [rax + 0x10], 7
0xcdab: add     rsp, 8
0xcdaf: pop     rbx
0xcdb0: pop     rbp
0xcdb1: ret
0xcdb2: nop     word ptr [rax + rax]
0xcdb8: add     rsi, 0x21
0xcdbc: mov     rdi, qword ptr [rbp + 0x58]
0xcdc0: mov     qword ptr [rbp + 0x50], rsi
0xcdc4: call    0x1d690
0xcdc9: mov     qword ptr [rbp + 0x58], rax
0xcdcd: mov     rbp, qword ptr [rip + 0x1e634]
0xcdd4: mov     rax, qword ptr [rbp + 0x38]

```

```

0xcdd8: lea      rdx, [rax + 1]
0xcddc: jmp      0xcd63
0xcd63:
0xcdde: nop
0xcde0: endbr64
0xcde4: mov      rax, qword ptr [rip + 0x1e61d]
0xcdeb: mov      rcx, qword ptr [rax + 0x18]
0xcdef: mov      rdx, qword ptr [rax + 0x38]
0xcdf3: cmp      rcx, rdx
0xcdf6: jb       0xce00
0xcdf8: ret
0xcdf9: nop      dword ptr [rax]
0xce00: push     rbx
0xce01: mov      rax, qword ptr [rax + 0x58]
0xce05: add      rdx, 1
0xce09: sub      rdx, rcx
0xce0c: lea      rsi, [rax + rcx + 1]
0xce11: lea      rdi, [rax + rcx]
0xce15: call     0x4bb0
0xce1a: mov      rbx, qword ptr [rip + 0x1e5e7]
0xce21: mov      rsi, qword ptr [rbx + 0x38]
0xce25: sub      qword ptr [rbx + 0x48], 1
0xce2a: lea      rax, [rsi - 1]
0xce2e: mov      qword ptr [rbx + 0x38], rax
0xce32: test     al, 0xf
0xce34: je       0xce40
0xce36: mov      dword ptr [rbx + 0x10], 8
0xce3d: pop      rbx
0xce3e: ret
0xce3f: nop
0xce40: mov      qword ptr [rbx + 0x50], rsi
0xce44: mov      rdi, qword ptr [rbx + 0x58]
0xce48: call     0x1d690
0xce4d: mov      qword ptr [rbx + 0x58], rax
0xce51: mov      rbx, qword ptr [rip + 0x1e5b0]
0xce58: mov      dword ptr [rbx + 0x10], 8
0xce5f: pop      rbx
0xce60: ret
0xce61: nop      word ptr cs:[rax + rax]
0xce6c: nop      dword ptr [rax]
0xce70: endbr64
0xce74: mov      rax, qword ptr [rip + 0x1e58d]
0xce7b: mov      rcx, qword ptr [rax + 0x18]
0xce7f: cmp      rcx, qword ptr [rax + 0x40]
0xce83: ja       0xce90
0xce85: ret
0xce86: nop      word ptr cs:[rax + rax]
0xce90: push     rbx

```

```

0xce91: mov     rdx, qword ptr [rax + 0x38]
0xce95: mov     rsi, qword ptr [rax + 0x58]
0xce99: add     rdx, 1
0xce9d: lea     rdi, [rsi + rcx - 1]
0xcea2: sub     rdx, rcx
0xcea5: add     rsi, rcx
0xcea8: call    0x4bb0
0xcead: mov     rbx, qword ptr [rip + 0x1e554]
0xceb4: mov     rsi, qword ptr [rbx + 0x38]
0xceb8: sub     qword ptr [rbx + 0x18], 1
0xcebd: sub     qword ptr [rbx + 0x48], 1
0xcec2: lea     rax, [rsi - 1]
0xcec6: mov     qword ptr [rbx + 0x38], rax
0xceca: test    al, 0xf
0xcecc: je      0xee0
0xcece: mov     dword ptr [rbx + 0x10], 9
0xcd5: pop     rbx
0xcd6: ret
0xcd7: nop     word ptr [rax + rax]
0xee0: mov     qword ptr [rbx + 0x50], rsi
0xee4: mov     rdi, qword ptr [rbx + 0x58]
0xee8: call    0x1d690
0xeed: mov     qword ptr [rbx + 0x58], rax
0xef1: mov     rbx, qword ptr [rip + 0x1e510]
0xef8: mov     dword ptr [rbx + 0x10], 9
0xefff: pop     rbx
0xcf00: ret
0xcf01: nop     word ptr cs:[rax + rax]
0xcf0c: nop     dword ptr [rax]
0xcf10: endbr64
0xcf14: push    rbp
0xcf15: push    rbx
0xcf16: sub     rsp, 8
0xcf1a: mov     r9, qword ptr [rip + 0x1e4e7]
0xcf21: mov     rax, qword ptr [r9 + 0x18]
0xcf25: mov     rbx, qword ptr [r9 + 0x20]
0xcf29: mov     qword ptr [r9 + 0x20], rax
0xcf2d: xor     eax, eax
0xcf2f: call    0xcc30
0xcf34: xor     eax, eax
0xcf36: mov     edi, 3
0xcf3b: mov     rbp, qword ptr [r9 + 0x18]
0xcf3f: call    0xc6e0
0xcf44: mov     rax, qword ptr [rip + 0x1e4bd]
0xcf4b: mov     rdx, qword ptr [rax + 0x18]
0xcf4f: cmp     rdx, rbx
0xcf52: jae     0xcf65
0xcf54: lea     rcx, [rdx + rbx]

```

```

0xcf58: sub     rcx, rbp
0xcf5b: cmp     rbx, rbp
0xcf5e: cmovae  rdx, rcx
0xcf62: mov     rbx, rdx
0xcf65: mov     qword ptr [rax + 0x20], rbx
0xcf69: mov     dword ptr [rax + 0x10], 0xa
0xcf70: add     rsp, 8
0xcf74: pop     rbx
0xcf75: pop     rbp
0xcf76: ret
0xcf77: nop     word ptr [rax + rax]
0xcf80: endbr64
0xcf84: push    rbp
0xcf85: push    rbx
0xcf86: sub     rsp, 8
0xcf8a: mov     rax, qword ptr [rip + 0x1e477]
0xcf91: mov     rbp, qword ptr [rax + 0x18]
0xcf95: mov     rbx, qword ptr [rax + 0x20]
0xcf99: mov     qword ptr [rax + 0x20], rbp
0xcf9d: xor     eax, eax
0xcf9f: call    0xcb60
0xcfa4: xor     eax, eax
0xcfa6: mov     edi, 3
0xcfab: call    0xc6e0
0xcfb0: mov     rax, qword ptr [rip + 0x1e451]
0xcfb7: mov     rdx, qword ptr [rax + 0x18]
0xcfb9: cmp     rdx, rbx
0xcfbe: jae     0xcfd1
0xcfc0: lea     rcx, [rdx + rbx]
0xcfc4: sub     rcx, rbp
0xcfc7: cmp     rbx, rbp
0xcfca: cmovae  rdx, rcx
0xcfce: mov     rbx, rdx
0xcfd1: mov     qword ptr [rax + 0x20], rbx
0xcfd5: mov     dword ptr [rax + 0x10], 0xb
0xcfdc: add     rsp, 8
0xcfe0: pop     rbx
0xcfe1: pop     rbp
0xcfe2: ret
0xcfe3: nop     word ptr cs:[rax + rax]
0xcfee: nop
0xcff0: endbr64
0xcff4: push    rbx
0xcff5: mov     rbx, qword ptr [rip + 0x1e40c]
0xcffc: mov     esi, 1
0xd001: mov     qword ptr [rbx + 0x18], 0
0xd009: mov     rdi, qword ptr [rbx + 0x58]
0xd00d: mov     qword ptr [rbx + 0x20], 0

```

```

0xd015: mov     qword ptr [rbx + 0x38], 0
0xd01d: mov     qword ptr [rbx + 0x40], 0
0xd025: mov     qword ptr [rbx + 0x48], 0
0xd02d: mov     qword ptr [rbx + 0x50], 1
0xd035: call    0x1d690
0xd03a: mov     qword ptr [rbx + 0x58], rax
0xd03e: mov     rax, qword ptr [rip + 0x1e3c3]
0xd045: mov     rax, qword ptr [rax + 0x58]
0xd049: mov     byte ptr [rax], 0
0xd04c: mov     rax, qword ptr [rip + 0x1e3b5]
0xd053: mov     dword ptr [rax + 0x10], 0xc
0xd05a: pop     rbx
0xd05b: ret
0xd05c: nop     dword ptr [rax]
0xd060: endbr64
0xd064: push    rbx
0xd065: mov     edi, 0x70
0xd06a: sub     rsp, 0x10
0xd06e: mov     rax, qword ptr fs:[0x28]
0xd077: mov     qword ptr [rsp + 8], rax
0xd07c: xor     eax, eax
0xd07e: call    0x1d670
0xd083: mov     qword ptr [rax + 8], 1
0xd08b: mov     qword ptr [rax + 0x58], 0
0xd093: mov     qword ptr [rax + 0x60], 0
0xd09b: mov     qword ptr [rax + 0x28], 0
0xd0a3: mov     qword ptr [rax + 0x30], 0
0xd0ab: mov     qword ptr [rip + 0x1e356], rax
0xd0b2: xor     eax, eax
0xd0b4: call    0xcff0
0xd0b9: mov     rbx, qword ptr [rip + 0x1e348]
0xd0c0: xor     eax, eax
0xd0c2: call    0xc590
0xd0c7: lea     rdi, [rip + 0x135e4]
0xd0ce: mov     qword ptr [rbx], rax
0xd0d1: call    0x17c50
0xd0d6: mov     ecx, 1
0xd0db: mov     rsi, rsp
0xd0de: mov     edx, 1
0xd0e3: lea     rdi, [rip + 0x13e42]
0xd0ea: call    0x17320
0xd0ef: mov     rdi, qword ptr [rsp]
0xd0f3: lea     rax, [rip + 0x13e24]
0xd0fa: mov     rbx, qword ptr [rip + 0x1e307]
0xd101: test    rdi, rdi
0xd104: cmov     rdi, rax
0xd108: xor     eax, eax
0xd10a: call    0x19940

```



```

0xd10f: lea     rdi, [rip + 0x1cb32]
0xd116: mov     qword ptr [rbx + 0x68], rax
0xd11a: mov     eax, dword ptr [rip + 0x1cb40]
0xd120: test    eax, eax
0xd122: lea     rax, [rip + 0x1cb07]
0xd129: cmov     rdi, rax
0xd12d: call    0x17c50
0xd132: lea     rdi, [rip + 0x1d0c7]
0xd139: mov     edx, 6
0xd13e: lea     rsi, [rip + 0x1d0db]
0xd145: call    0x17f70
0xd14a: xor     eax, eax
0xd14c: call    0x1a310
0xd151: mov     rax, qword ptr [rip + 0x1e2b0]
0xd158: mov     rdi, qword ptr [rax + 0x68]
0xd15c: xor     eax, eax
0xd15e: call    0x1a900
0xd163: nop     dword ptr [rax + rax]
0xd168: xor     eax, eax
0xd16a: call    0x1ab40
0xd16f: test    rax, rax
0xd172: jne     0xd168
0xd174: mov     edi, 0x200
0xd179: call    0x1a620
0xd17e: mov     rax, qword ptr [rsp + 8]
0xd183: sub     rax, qword ptr fs:[0x28]
0xd18c: jne     0xd194
0xd18e: add     rsp, 0x10
0xd192: pop     rbx
0xd193: ret
0xd194: call    0x49d0
0xd199: nop     dword ptr [rax]
0xd1a0: endbr64
0xd1a4: sub     rsp, 8
0xd1a8: mov     rax, qword ptr [rip + 0x1e259]
0xd1af: or      edi, 2
0xd1b2: mov     rdx, qword ptr [rax + 0x40]
0xd1b6: mov     dword ptr [rax + 0x10], 6
0xd1bd: mov     qword ptr [rax + 0x20], rdx
0xd1c1: mov     rdx, qword ptr [rax + 0x38]
0xd1c5: mov     qword ptr [rax + 0x18], rdx
0xd1c9: xor     eax, eax
0xd1cb: call    0xc6e0
0xd1d0: mov     rax, qword ptr [rip + 0x1e231]
0xd1d7: mov     rdx, qword ptr [rax + 0x18]
0xd1db: mov     dword ptr [rax + 0x10], 0xd
0xd1e2: mov     qword ptr [rax + 0x20], rdx
0xd1e6: add     rsp, 8

```

```

0xd1ea: ret
0xd1eb: nop      dword ptr [rax + rax]
0xd1f0: endbr64
0xd1f4: mov      rax, qword ptr [rip + 0x1e20d]
0xd1fb: push     rbx
0xd1fc: mov      edi, 3
0xd201: mov      rdx, qword ptr [rax + 0x18]
0xd205: mov      rsi, qword ptr [rax + 0x20]
0xd209: mov      dword ptr [rax + 0x10], 5
0xd210: mov      rcx, qword ptr [rax + 0x40]
0xd214: mov      qword ptr [rax + 0x20], rdx
0xd218: mov      rbx, rsi
0xd21b: mov      qword ptr [rax + 0x18], rcx
0xd21f: sub      rbx, rdx
0xd222: cmp      rsi, rdx
0xd225: cmovbe   rbx, rcx
0xd229: xor      eax, eax
0xd22b: call     0xc6e0
0xd230: mov      rdx, qword ptr [rip + 0x1e1d1]
0xd237: mov      rax, qword ptr [rdx + 0x38]
0xd23b: mov      dword ptr [rdx + 0x10], 0xe
0xd242: cmp      rax, rbx
0xd245: cmova    rax, rbx
0xd249: mov      qword ptr [rdx + 0x20], rax
0xd24d: pop      rbx
0xd24e: ret
0xd24f: nop
0xd250: endbr64
0xd254: mov      rax, qword ptr [rip + 0x1e1ad]
0xd25b: push     rbx
0xd25c: mov      edi, 3
0xd261: mov      rdx, qword ptr [rax + 0x18]
0xd265: mov      rbx, qword ptr [rax + 0x20]
0xd269: mov      dword ptr [rax + 0x10], 6
0xd270: mov      qword ptr [rax + 0x20], rdx
0xd274: mov      rdx, qword ptr [rax + 0x38]
0xd278: mov      qword ptr [rax + 0x18], rdx
0xd27c: xor      eax, eax
0xd27e: call     0xc6e0
0xd283: mov      rdx, qword ptr [rip + 0x1e17e]
0xd28a: mov      rax, qword ptr [rdx + 0x38]
0xd28e: mov      dword ptr [rdx + 0x10], 0xf
0xd295: cmp      rax, rbx
0xd298: cmova    rax, rbx
0xd29c: mov      qword ptr [rdx + 0x20], rax
0xd2a0: pop      rbx
0xd2a1: ret
0xd2a2: nop      word ptr cs:[rax + rax]

```

```

0xd2ad: nop      dword ptr [rax]
0xd2b0: endbr64
0xd2b4: mov      rax, qword ptr [rip + 0x1e14d]
0xd2bb: mov      rdx, qword ptr [rax + 0x58]
0xd2bf: mov      rax, qword ptr [rax + 0x18]
0xd2c3: cmp      byte ptr [rdx + rax], 0x20
0xd2c7: je       0xd2d0
0xd2c9: ret
0xd2ca: nop      word ptr [rax + rax]
0xd2d0: sub      rsp, 8
0xd2d4: nop      dword ptr [rax]
0xd2d8: xor      eax, eax
0xd2da: call     0xcde0
0xd2df: mov      rax, qword ptr [rip + 0x1e122]
0xd2e6: mov      rdx, qword ptr [rax + 0x58]
0xd2ea: mov      rcx, qword ptr [rax + 0x18]
0xd2ee: cmp      byte ptr [rdx + rcx], 0x20
0xd2f2: je       0xd2d8
0xd2f4: cmp      qword ptr [rax + 0x48], 0
0xd2f9: jne      0xd31d
0xd2fb: jmp      0xd324
0xd324:
0xd2fd: nop      dword ptr [rax]
0xd300: xor      eax, eax
0xd302: call     0xce70
0xd307: mov      rax, qword ptr [rip + 0x1e0fa]
0xd30e: cmp      qword ptr [rax + 0x48], 0
0xd313: je       0xd324
0xd315: mov      rdx, qword ptr [rax + 0x58]
0xd319: mov      rcx, qword ptr [rax + 0x18]
0xd31d: cmp      byte ptr [rdx + rcx - 1], 0x20
0xd322: je       0xd300
0xd324: mov      dword ptr [rax + 0x10], 0x11
0xd32b: add      rsp, 8
0xd32f: ret
0xd330: endbr64
0xd334: mov      rax, qword ptr [rip + 0x1e0cd]
0xd33b: mov      rdx, qword ptr [rax + 0x58]
0xd33f: mov      rax, qword ptr [rax + 0x18]
0xd343: cmp      byte ptr [rdx + rax], 0x20
0xd347: je       0xd350
0xd349: ret
0xd34a: nop      word ptr [rax + rax]
0xd350: sub      rsp, 8
0xd354: xor      eax, eax
0xd356: call     0xd2b0
0xd35b: mov      edi, 0x20
0xd360: call     0xcd30

```

```

0xd365: mov     rax, qword ptr [rip + 0x1e09c]
0xd36c: mov     dword ptr [rax + 0x10], 0x10
0xd373: add     rsp, 8
0xd377: ret
0xd378: nop     dword ptr [rax + rax]
0xd380: endbr64
0xd384: push    rbp
0xd385: push    rbx
0xd386: sub     rsp, 8
0xd38a: mov     rbp, qword ptr [rip + 0x1e077]
0xd391: mov     rbx, qword ptr [rbp + 0x18]
0xd395: cmp     rbx, qword ptr [rbp + 0x38]
0xd399: jb      0xd3a8
0xd39b: add     rsp, 8
0xd39f: pop     rbx
0xd3a0: pop     rbp
0xd3a1: ret
0xd3a2: nop     word ptr [rax + rax]
0xd3a8: xor     eax, eax
0xd3aa: call    0xcc30
0xd3af: cmp     rbx, qword ptr [rbp + 0x18]
0xd3b3: jae     0xd3e4
0xd3b5: call    0x4ef0
0xd3ba: nop     word ptr [rax + rax]
0xd3c0: mov     rdx, qword ptr [rbp + 0x58]
0xd3c4: mov     rcx, qword ptr [rax]
0xd3c7: add     rdx, rbx
0xd3ca: add     rbx, 1
0xd3ce: movsx   rsi, byte ptr [rdx]
0xd3d2: mov     ecx, dword ptr [rcx + rsi*4]
0xd3d5: mov     byte ptr [rdx], cl
0xd3d7: mov     rbp, qword ptr [rip + 0x1e02a]
0xd3de: cmp     qword ptr [rbp + 0x18], rbx
0xd3e2: ja      0xd3c0
0xd3e4: mov     dword ptr [rbp + 0x10], 0x12
0xd3eb: add     rsp, 8
0xd3ef: pop     rbx
0xd3f0: pop     rbp
0xd3f1: ret
0xd3f2: nop     word ptr cs:[rax + rax]
0xd3fd: nop     dword ptr [rax]
0xd400: endbr64
0xd404: push    rbp
0xd405: push    rbx
0xd406: sub     rsp, 8
0xd40a: mov     rbp, qword ptr [rip + 0x1dff7]
0xd411: mov     rbx, qword ptr [rbp + 0x18]
0xd415: cmp     rbx, qword ptr [rbp + 0x38]

```

```

0xd419: jb      0xd428
0xd41b: add     rsp, 8
0xd41f: pop     rbx
0xd420: pop     rbp
0xd421: ret
0xd422: nop     word ptr [rax + rax]
0xd428: xor     eax, eax
0xd42a: call    0xcc30
0xd42f: cmp     rbx, qword ptr [rbp + 0x18]
0xd433: jae     0xd464
0xd435: call    0x47c0
0xd43a: nop     word ptr [rax + rax]
0xd440: mov     rdx, qword ptr [rbp + 0x58]
0xd444: mov     rcx, qword ptr [rax]
0xd447: add     rdx, rbx
0xd44a: add     rbx, 1
0xd44e: movsx   rsi, byte ptr [rdx]
0xd452: mov     ecx, dword ptr [rcx + rsi*4]
0xd455: mov     byte ptr [rdx], cl
0xd457: mov     rbp, qword ptr [rip + 0x1dfaa]
0xd45e: cmp     qword ptr [rbp + 0x18], rbx
0xd462: ja      0xd440
0xd464: mov     dword ptr [rbp + 0x10], 0x13
0xd46b: add     rsp, 8
0xd46f: pop     rbx
0xd470: pop     rbp
0xd471: ret
0xd472: nop     word ptr cs:[rax + rax]
0xd47d: nop     dword ptr [rax]
0xd480: endbr64
0xd484: push    r14
0xd486: push    r13
0xd488: push    r12
0xd48a: push    rbp
0xd48b: mov     rbp, qword ptr [rip + 0x1df76]
0xd492: push    rbx
0xd493: mov     rbx, qword ptr [rbp + 0x18]
0xd497: cmp     rbx, qword ptr [rbp + 0x38]
0xd49b: jb      0xd4b0
0xd49d: pop     rbx
0xd49e: pop     rbp
0xd49f: pop     r12
0xd4a1: pop     r13
0xd4a3: pop     r14
0xd4a5: ret
0xd4a6: nop     word ptr cs:[rax + rax]
0xd4b0: xor     eax, eax
0xd4b2: call    0xcc30

```

```

0xd4b7: cmp      rbx, qword ptr [rbp + 0x18]
0xd4bb: jae      0xd528
0xd4bd: call     0x4f00
0xd4c2: mov      r14d, 1
0xd4c8: mov      r13, rax
0xd4cb: jmp      0xd4f3
0xd4f3:
0xd4cd: nop      dword ptr [rax]
0xd4d0: call     0x47c0
0xd4d5: xor      r14d, r14d
0xd4d8: mov      rax, qword ptr [rax]
0xd4db: mov      eax, dword ptr [rax + r12*4]
0xd4df: mov      byte ptr [rbp], al
0xd4e2: mov      rbp, qword ptr [rip + 0x1df1f]
0xd4e9: add      rbx, 1
0xd4ed: cmp      qword ptr [rbp + 0x18], rbx
0xd4f1: jbe      0xd528
0xd4f3: mov      rax, qword ptr [rbp + 0x58]
0xd4f7: add      rax, rbx
0xd4fa: movsx    r12, byte ptr [rax]
0xd4fe: mov      rbp, rax
0xd501: mov      rax, qword ptr [r13]
0xd505: test     byte ptr [rax + r12*2], 8
0xd50a: je       0xd4e2
0xd50c: test     r14d, r14d
0xd50f: jne      0xd4d0
0xd511: call     0x4ef0
0xd516: mov      rax, qword ptr [rax]
0xd519: mov      eax, dword ptr [rax + r12*4]
0xd51d: mov      byte ptr [rbp], al
0xd520: jmp      0xd4e2
0xd4e2:
0xd522: nop      word ptr [rax + rax]
0xd528: mov      dword ptr [rbp + 0x10], 0x14
0xd52f: pop      rbx
0xd530: pop      rbp
0xd531: pop      r12
0xd533: pop      r13
0xd535: pop      r14
0xd537: ret
0xd538: nop      dword ptr [rax + rax]
0xd540: endbr64
0xd544: push     r12
0xd546: mov      r12, rdi
0xd549: push     rbp
0xd54a: push     rbx
0xd54b: call     0x49b0
0xd550: mov      rbp, qword ptr [rip + 0x1deb1]

```

```

0xd557: mov     rbx, rax
0xd55a: mov     rsi, qword ptr [rbp + 0x40]
0xd55e: sub     rax, rsi
0xd561: add     qword ptr [rbp + 0x18], rax
0xd565: add     qword ptr [rbp + 0x20], rax
0xd569: mov     rax, qword ptr [rbp + 0x48]
0xd56d: lea     rdx, [rax + 1]
0xd571: lea     r8, [rbx + rdx]
0xd575: cmp     r8, qword ptr [rbp + 0x50]
0xd579: ja      0xd5d8
0xd57b: mov     rdi, qword ptr [rbp + 0x58]
0xd57f: add     rsi, rdi
0xd582: add     rdi, rbx
0xd585: call    0x4d40
0xd58a: mov     rax, qword ptr [rip + 0x1de77]
0xd591: mov     rdx, rbx
0xd594: mov     rsi, r12
0xd597: mov     rdi, qword ptr [rax + 0x58]
0xd59b: call    0x4bb0
0xd5a0: mov     rax, qword ptr [rip + 0x1de61]
0xd5a7: mov     rsi, rbx
0xd5aa: mov     rdi, qword ptr [rax + 0x58]
0xd5ae: call    0x184d0
0xd5b3: mov     rax, qword ptr [rip + 0x1de4e]
0xd5ba: mov     qword ptr [rax + 0x40], rbx
0xd5be: add     rbx, qword ptr [rax + 0x48]
0xd5c2: mov     qword ptr [rax + 0x38], rbx
0xd5c6: mov     dword ptr [rax + 0x10], 0x15
0xd5cd: pop     rbx
0xd5ce: pop     rbp
0xd5cf: pop     r12
0xd5d1: ret
0xd5d2: nop     word ptr [rax + rax]
0xd5d8: mov     qword ptr [rbp + 0x50], r8
0xd5dc: mov     rdi, qword ptr [rbp + 0x58]
0xd5e0: mov     rsi, r8
0xd5e3: call    0x1d690
0xd5e8: mov     qword ptr [rbp + 0x58], rax
0xd5ec: mov     rbp, qword ptr [rip + 0x1de15]
0xd5f3: mov     rax, qword ptr [rbp + 0x48]
0xd5f7: mov     rsi, qword ptr [rbp + 0x40]
0xd5fb: lea     rdx, [rax + 1]
0xd5ff: jmp     0xd57b
0xd57b:
0xd604: nop     word ptr cs:[rax + rax]
0xd60f: nop
0xd610: endbr64
0xd614: test    rdi, rdi

```

```

0xd617: je      0xd6f0
0xd61d: push    r12
0xd61f: push    rbp
0xd620: mov     rbp, rdi
0xd623: push    rbx
0xd624: call    0x49b0
0xd629: mov     r12, qword ptr [rip + 0x1ddd8]
0xd630: mov     rbx, rax
0xd633: mov     rax, qword ptr [r12 + 0x38]
0xd638: lea     rdx, [rax + 1]
0xd63c: mov     rax, qword ptr [r12 + 0x50]
0xd641: lea     rcx, [rbx + rdx]
0xd645: cmp     rcx, rax
0xd648: ja      0xd6c0
0xd64a: mov     rax, qword ptr [r12 + 0x18]
0xd64f: mov     rdi, qword ptr [r12 + 0x58]
0xd654: sub     rdx, rax
0xd657: lea     rsi, [rdi + rax]
0xd65b: add     rax, rbx
0xd65e: add     rdi, rax
0xd661: call    0x4d40
0xd666: mov     rax, qword ptr [rip + 0x1dd9b]
0xd66d: mov     rdx, rbx
0xd670: mov     rsi, rbp
0xd673: mov     rdi, qword ptr [rax + 0x18]
0xd677: add     rdi, qword ptr [rax + 0x58]
0xd67b: call    0x4bb0
0xd680: mov     rax, qword ptr [rip + 0x1dd81]
0xd687: mov     rsi, rbx
0xd68a: mov     rdi, qword ptr [rax + 0x18]
0xd68e: add     rdi, qword ptr [rax + 0x58]
0xd692: call    0x184d0
0xd697: mov     rax, qword ptr [rip + 0x1dd6a]
0xd69e: add     qword ptr [rax + 0x18], rbx
0xd6a2: add     qword ptr [rax + 0x38], rbx
0xd6a6: add     qword ptr [rax + 0x48], rbx
0xd6aa: mov     dword ptr [rax + 0x10], 0x16
0xd6b1: pop     rbx
0xd6b2: pop     rbp
0xd6b3: pop     r12
0xd6b5: ret
0xd6b6: nop     word ptr cs:[rax + rax]
0xd6c0: lea     rsi, [rax + rbx + 0x21]
0xd6c5: mov     rdi, qword ptr [r12 + 0x58]
0xd6ca: mov     qword ptr [r12 + 0x50], rsi
0xd6cf: call    0x1d690
0xd6d4: mov     qword ptr [r12 + 0x58], rax
0xd6d9: mov     r12, qword ptr [rip + 0x1dd28]

```



```

0xd6e0: mov     rax, qword ptr [r12 + 0x38]
0xd6e5: lea     rdx, [rax + 1]
0xd6e9: jmp     0xd64a
0xd64a:
0xd6ee: nop
0xd6f0: ret
0xd6f1: nop     word ptr cs:[rax + rax]
0xd6fc: nop     dword ptr [rax]
0xd700: endbr64
0xd704: mov     rax, qword ptr [rip + 0x1dcfd]
0xd70b: mov     rdi, qword ptr [rax + 0x60]
0xd70f: test    rdi, rdi
0xd712: je      0xd730
0xd714: sub     rsp, 8
0xd718: call    0xd610
0xd71d: mov     rax, qword ptr [rip + 0x1dce4]
0xd724: mov     dword ptr [rax + 0x10], 0x1a
0xd72b: add     rsp, 8
0xd72f: ret
0xd730: ret
0xd731: nop     word ptr cs:[rax + rax]
0xd73c: nop     dword ptr [rax]
0xd740: endbr64
0xd744: mov     rax, qword ptr [rip + 0x1dcbd]
0xd74b: push    r12
0xd74d: push    rbp
0xd74e: push    rbx
0xd74f: mov     rbx, qword ptr [rax + 0x40]
0xd753: mov     r12, qword ptr [rax + 0x28]
0xd757: mov     eax, dword ptr [rax + 0xc]
0xd75a: lea     rbp, [rbx + 3]
0xd75e: test    eax, eax
0xd760: je      0xd7f0
0xd766: mov     edx, dword ptr [rip + 0x1caa4]
0xd76c: mov     esi, dword ptr [rip + 0x1ca9a]
0xd772: mov     edi, dword ptr [rip + 0x1ca9c]
0xd778: call    0xa570
0xd77d: mov     rcx, qword ptr [rip + 0x1dc84]
0xd784: cmp     r12, rbp
0xd787: jb      0xd810
0xd78d: mov     rsi, qword ptr [rcx + 0x18]
0xd791: mov     rdx, qword ptr [rcx + 0x40]
0xd795: xor     ebx, ebx
0xd797: mov     rdi, qword ptr [rcx + 0x28]
0xd79b: mov     rax, rdi
0xd79e: sub     rax, rdx
0xd7a1: mov     edx, 1
0xd7a6: mov     r8, rax

```

```

0xd7a9: shr      r8, 2
0xd7ad: cmp      rax, 3
0xd7b1: cmova    edx, r8d
0xd7b5: xor      eax, eax
0xd7b7: cmp      rdi, rsi
0xd7ba: ja       0xd7d3
0xd7bc: mov      rax, rsi
0xd7bf: sub      rax, rdi
0xd7c2: movsxd   rdi, edx
0xd7c5: lea      r8d, [rdx + rax]
0xd7c9: xor      edx, edx
0xd7cb: div      rdi
0xd7ce: mov      eax, r8d
0xd7d1: sub      eax, edx
0xd7d3: mov      edx, esi
0xd7d5: mov      rdi, qword ptr [rcx]
0xd7d8: xor      esi, esi
0xd7da: sub      edx, eax
0xd7dc: call     0xc680
0xd7e1: cmp      r12, rbp
0xd7e4: jb       0xd840
0xd7e6: pop      rbx
0xd7e7: pop      rbp
0xd7e8: pop      r12
0xd7ea: ret
0xd7eb: nop      dword ptr [rax + rax]
0xd7f0: mov      edx, dword ptr [rip + 0x1ca0e]
0xd7f6: mov      esi, dword ptr [rip + 0x1ca04]
0xd7fc: mov      edi, dword ptr [rip + 0x1ca06]
0xd802: jmp      0xd778
0xd778
0xd807: nop      word ptr [rax + rax]
0xd810: mov      rbx, qword ptr [rcx + 0x40]
0xd814: mov      rsi, qword ptr [rcx + 0x18]
0xd818: mov      qword ptr [rcx + 0x40], 0
0xd820: xor      edx, edx
0xd822: add      qword ptr [rcx + 0x58], rbx
0xd826: sub      rsi, rbx
0xd829: sub      qword ptr [rcx + 0x38], rbx
0xd82d: sub      qword ptr [rcx + 0x20], rbx
0xd831: movsxd   rbx, ebx
0xd834: mov      qword ptr [rcx + 0x18], rsi
0xd838: jmp      0xd797
0xd797
0xd83d: nop      dword ptr [rax]
0xd840: mov      rax, qword ptr [rip + 0x1dbc1]
0xd847: movsxd   rbx, ebx
0xd84a: mov      qword ptr [rax + 0x40], rbx

```

```

0xd84e: sub      qword ptr [rax + 0x58], rbx
0xd852: add      qword ptr [rax + 0x38], rbx
0xd856: add      qword ptr [rax + 0x18], rbx
0xd85a: add      qword ptr [rax + 0x20], rbx
0xd85e: pop      rbx
0xd85f: pop      rbp
0xd860: pop      r12
0xd862: ret
0xd863: nop      word ptr cs:[rax + rax]
0xd86e: nop
0xd870: endbr64
0xd874: push     r15
0xd876: push     r14
0xd878: push     r13
0xd87a: push     r12
0xd87c: push     rbp
0xd87d: push     rbx
0xd87e: sub      rsp, 0x28
0xd882: mov      rax, qword ptr fs:[0x28]
0xd88b: mov      qword ptr [rsp + 0x18], rax
0xd890: mov      rax, qword ptr [rip + 0x1db71]
0xd897: lea      r14, [rsp + 0x17]
0xd89c: mov      rdi, r14
0xd89f: mov      rbx, qword ptr [rax + 0x40]
0xd8a3: mov      r13, qword ptr [rax + 0x28]
0xd8a7: call     0xa6a0
0xd8ac: mov      rax, qword ptr [rip + 0x1db55]
0xd8b3: add      rbx, 3
0xd8b7: mov      ecx, dword ptr [rax + 0xc]
0xd8ba: test     ecx, ecx
0xd8bc: je       0xda20
0xd8c2: mov      edx, dword ptr [rip + 0x1c948]
0xd8c8: mov      esi, dword ptr [rip + 0x1c93e]
0xd8ce: mov      edi, dword ptr [rip + 0x1c940]
0xd8d4: call     0xa570
0xd8d9: mov      rdx, qword ptr [rip + 0x1db28]
0xd8e0: cmp      r13, rbx
0xd8e3: jb       0xda90
0xd8e9: mov      qword ptr [rsp + 8], 0
0xd8f2: mov      rax, qword ptr [rdx + 0x18]
0xd8f6: mov      rcx, qword ptr [rdx + 0x40]
0xd8fa: mov      rdi, qword ptr [rdx + 0x28]
0xd8fe: mov      r15, rdi
0xd901: sub      r15, rcx
0xd904: cmp      r15, 3
0xd908: jbe      0xda80
0xd90e: shr      r15, 2
0xd912: movsxd   rcx, r15d

```

```

0xd915: cmp     rdi, rax
0xd918: ja      0xda70
0xd91e: mov     edx, eax
0xd920: sub     rax, rdi
0xd923: sub     edx, edi
0xd925: add     r15d, edx
0xd928: xor     edx, edx
0xd92a: div     rcx
0xd92d: sub     r15d, edx
0xd930: mov     r12d, r15d
0xd933: call    0x1d670
0xd938: mov     esi, 0x20
0xd93d: mov     rbp, rax
0xd940: mov     rax, qword ptr [rip + 0x1dac1]
0xd947: mov     rdi, rbp
0xd94a: mov     rdx, qword ptr [rax + 0x28]
0xd94e: call    0x4a50
0xd953: mov     rax, qword ptr [rip + 0x1daae]
0xd95a: mov     edx, dword ptr [rax + 8]
0xd95d: test    edx, edx
0xd95f: je      0xda40
0xd965: mov     rdx, qword ptr [rax + 0x38]
0xd969: sub     rdx, qword ptr [rax + 0x40]
0xd96d: mov     rdi, rbp
0xd970: sub     rdx, r12
0xd973: mov     rsi, qword ptr [rax + 0x28]
0xd977: sub     rsi, qword ptr [rax + 0x40]
0xd97b: cmp     rdx, rsi
0xd97e: cmova   rdx, rsi
0xd982: mov     rsi, qword ptr [rax + 0x40]
0xd986: add     rsi, r12
0xd989: add     rsi, qword ptr [rax + 0x58]
0xd98d: call    0x4bb0
0xd992: mov     rax, qword ptr [rip + 0x1da6f]
0xd999: xor     edx, edx
0xd99b: xor     esi, esi
0xd99d: mov     rdi, qword ptr [rax]
0xd9a0: call    0xc680
0xd9a5: cmp     r13, rbx
0xd9a8: jae     0xdaf0
0xd9ae: mov     rax, qword ptr [rip + 0x1da53]
0xd9b5: mov     rsi, rbp
0xd9b8: mov     rdi, qword ptr [rax]
0xd9bb: mov     rdx, qword ptr [rax + 0x28]
0xd9bf: sub     edx, dword ptr [rax + 0x40]
0xd9c2: call    0xc600
0xd9c7: mov     rax, qword ptr [rip + 0x1da3a]
0xd9ce: xor     esi, esi

```

```

0xd9d0: mov     edx, dword ptr [rax + 0x18]
0xd9d3: mov     rdi, qword ptr [rax]
0xd9d6: sub     edx, r15d
0xd9d9: call    0xc680
0xd9de: cmp     r13, rbx
0xd9e1: jb     0xdac8
0xd9e7: mov     rdi, rbp
0xd9ea: call    0x18d10
0xd9ef: mov     rdi, r14
0xd9f2: call    0xa6b0
0xd9f7: mov     rax, qword ptr [rsp + 0x18]
0xd9fc: sub     rax, qword ptr fs:[0x28]
0xda05: jne     0xdb0b
0xda0b: add     rsp, 0x28
0xda0f: pop     rbx
0xda10: pop     rbp
0xda11: pop     r12
0xda13: pop     r13
0xda15: pop     r14
0xda17: pop     r15
0xda19: ret
0xda1a: nop     word ptr [rax + rax]
0xda20: mov     edx, dword ptr [rip + 0x1c7de]
0xda26: mov     esi, dword ptr [rip + 0x1c7d4]
0xda2c: mov     edi, dword ptr [rip + 0x1c7d6]
0xda32: jmp     0xd8d4
0xd8d4:
0xda37: nop     word ptr [rax + rax]
0xda40: mov     rsi, qword ptr [rax + 0x40]
0xda44: mov     rdx, qword ptr [rax + 0x38]
0xda48: mov     rdi, rbp
0xda4b: mov     rax, qword ptr [rax + 0x28]
0xda4f: sub     rdx, rsi
0xda52: sub     rdx, r12
0xda55: sub     rax, rsi
0xda58: mov     esi, 0x2a
0xda5d: cmp     rdx, rax
0xda60: cmova   rdx, rax
0xda64: call    0x4a50
0xda69: jmp     0xd992
0xd992:
0xda6e: nop
0xda70: xor     r12d, r12d
0xda73: xor     r15d, r15d
0xda76: jmp     0xd933
0xd933:
0xda7b: nop     dword ptr [rax + rax]
0xda80: mov     ecx, 1

```

```

0xda85: mov     r15d, 1
0xda8b: jmp     0xd915
0xd915:
0xda90: mov     r12, qword ptr [rdx + 0x40]
0xda94: mov     rax, qword ptr [rdx + 0x18]
0xda98: mov     qword ptr [rdx + 0x40], 0
0xdaa0: add     qword ptr [rdx + 0x58], r12
0xdaa4: movsxd  rcx, r12d
0xdaa7: sub     rax, r12
0xdaaa: sub     qword ptr [rdx + 0x38], r12
0xdaae: mov     qword ptr [rsp + 8], rcx
0xdab3: xor     ecx, ecx
0xdab5: sub     qword ptr [rdx + 0x20], r12
0xdab9: mov     qword ptr [rdx + 0x18], rax
0xdabd: jmp     0xd8fa
0xd8fa:
0xdac2: nop     word ptr [rax + rax]
0xdac8: mov     rax, qword ptr [rip + 0x1d939]
0xdacf: movsxd  r12, dword ptr [rsp + 8]
0xdad4: sub     qword ptr [rax + 0x58], r12
0xdad8: add     qword ptr [rax + 0x38], r12
0xdadc: add     qword ptr [rax + 0x18], r12
0xdae0: add     qword ptr [rax + 0x20], r12
0xdae4: mov     qword ptr [rax + 0x40], r12
0xdae8: jmp     0xd9e7
0xd9e7:
0xdaed: nop     dword ptr [rax]
0xdaf0: mov     rax, qword ptr [rip + 0x1d911]
0xdaf7: mov     rsi, qword ptr [rax + 0x58]
0xdafb: mov     edx, dword ptr [rax + 0x40]
0xdafe: mov     rdi, qword ptr [rax]
0xdb01: call    0xc600
0xdb06: jmp     0xd9ae
0xd9ae:
0xdb0b: call    0x49d0
0xdb10: endbr64
0xdb14: mov     rax, qword ptr [rip + 0x1d8ed]
0xdb1b: push    rbx
0xdb1c: mov     rbx, rdi
0xdb1f: mov     rdi, qword ptr [rdi]
0xdb22: mov     rsi, qword ptr [rax + 0x48]
0xdb26: add     rsi, 1
0xdb2a: call    0x1d690
0xdb2f: mov     qword ptr [rbx], rax
0xdb32: mov     rdi, rax
0xdb35: mov     rax, qword ptr [rip + 0x1d8cc]
0xdb3c: mov     rcx, qword ptr [rax + 0x48]
0xdb40: mov     rsi, qword ptr [rax + 0x40]

```

```

0xdb44: add     rsi, qword ptr [rax + 0x58]
0xdb48: lea     rdx, [rcx + 1]
0xdb4c: call    0x4bb0
0xdb51: mov     rax, qword ptr [rip + 0x1d8b0]
0xdb58: pop     rbx
0xdb59: mov     eax, dword ptr [rax + 0x48]
0xdb5c: ret
0xdb5d: nop     dword ptr [rax]
0xdb60: endbr64
0xdb64: push    rbp
0xdb65: xor     eax, eax
0xdb67: mov     rbp, rdi
0xdb6a: call    0xcff0
0xdb6f: test    rbp, rbp
0xdb72: lea     rax, [rip + 0x133bf]
0xdb79: cmov     rbp, rax
0xdb7d: mov     rdi, rbp
0xdb80: call    0xd540
0xdb85: xor     eax, eax
0xdb87: call    0xd870
0xdb8c: xor     eax, eax
0xdb8e: pop     rbp
0xdb8f: jmp     0xd740
0xd740
0xdb94: nop     word ptr cs:[rax + rax]
0xdb9f: nop
0xdba0: endbr64
0xdba4: mov     rax, qword ptr [rip + 0x1d85d]
0xdbab: mov     dword ptr [rax + 0xc], edi
0xdbae: ret
0xdbaf: nop
0xdbb0: endbr64
0xdbb4: push    r12
0xdbb6: push    rbp
0xdbb7: push    rbx
0xdbb8: cmp     edi, 1
0xdbbb: je      0xdbd0
0xdbbd: cmp     edi, 2
0xdbc0: je      0xdc90
0xdbc6: test    edi, edi
0xdbc8: je      0xdc40
0xdbca: pop     rbx
0xdbcb: pop     rbp
0xdbcc: pop     r12
0xdbce: ret
0xdbcf: nop
0xdbd0: mov     ecx, dword ptr [rip + 0x1d83e]
0xdbd6: test    ecx, ecx

```

```

0xdbd8: je      0xdd38
0xdbde: xor      eax, eax
0xdbe0: call    0x1ab10
0xdbe5: mov     rbp, rax
0xdbe8: test    rax, rax
0xdbeb: je      0xdbca
0xdbed: mov     r12, qword ptr [rbp]
0xdbf1: mov     rbx, qword ptr [rip + 0x1d810]
0xdbf8: mov     rdi, r12
0xdbfb: call    0x49b0
0xdc00: mov     rdi, qword ptr [rbx + 0x40]
0xdc04: mov     qword ptr [rbx + 0x48], rax
0xdc08: lea     rsi, [rax + rdi]
0xdc0c: mov     qword ptr [rbx + 0x38], rsi
0xdc10: mov     qword ptr [rbx + 0x18], rsi
0xdc14: add     rsi, 1
0xdc18: cmp     rsi, qword ptr [rbx + 0x50]
0xdc1c: ja      0xdd70
0xdc22: add     rdi, qword ptr [rbx + 0x58]
0xdc26: mov     rsi, r12
0xdc29: call    0x48b0
0xdc2e: pop     rbx
0xdc2f: xor     eax, eax
0xdc31: pop     rbp
0xdc32: pop     r12
0xdc34: jmp     0xd870
0xd870:
0xdc39: nop     dword ptr [rax]
0xdc40: mov     eax, dword ptr [rip + 0x1d7ce]
0xdc46: test    eax, eax
0xdc48: jne     0xdd18
0xdc4e: xor     eax, eax
0xdc50: call    0x1ab10
0xdc55: test    rax, rax
0xdc58: je      0xdd98
0xdc5e: mov     rdx, qword ptr [rip + 0x1d7a3]
0xdc65: mov     rsi, qword ptr [rax]
0xdc68: mov     rbp, qword ptr [rdx + 0x40]
0xdc6c: add     rbp, qword ptr [rdx + 0x58]
0xdc70: mov     rdi, rbp
0xdc73: call    0x4b30
0xdc78: test    eax, eax
0xdc7a: jne     0xdd58
0xdc80: pop     rbx
0xdc81: xor     eax, eax
0xdc83: pop     rbp
0xdc84: pop     r12
0xdc86: jmp     0x1ab40

```



```

0x1ab40
0xdc8b: nop      dword ptr [rax + rax]
0xdc90: mov      edx, dword ptr [rip + 0x1d77e]
0xdc96: test     edx, edx
0xdc98: je       0xdcf8
0xdc9a: xor      eax, eax
0xdc9c: call     0x1ab40
0xdca1: mov      rbp, rax
0xdca4: test     rax, rax
0xdca7: jne      0xdbed
0xdcad: mov      rax, qword ptr [rip + 0x1d754]
0xdcb4: mov      edi, 2
0xdcb9: mov      rdx, qword ptr [rax + 0x40]
0xdcbd: mov      dword ptr [rax + 0x10], 6
0xdcc4: mov      qword ptr [rax + 0x20], rdx
0xdcc8: mov      rdx, qword ptr [rax + 0x38]
0xdccc: mov      qword ptr [rax + 0x18], rdx
0xcdcd0: xor      eax, eax
0xcdcd2: call     0xc6e0
0xcdcd7: mov      rax, qword ptr [rip + 0x1d72a]
0xcdcde: mov      rdx, qword ptr [rax + 0x18]
0xdce2: mov      dword ptr [rax + 0x10], 0xd
0xdce9: mov      qword ptr [rax + 0x20], rdx
0xcdced: jmp      0xdc2e
0xdc2e
0xdcf2: nop      word ptr [rax + rax]
0xdcf8: mov      dword ptr [rip + 0x1d712], 1
0xdd02: xor      eax, eax
0xdd04: call     0x1a570
0xdd09: mov      dword ptr [rip + 0x1d701], eax
0xdd0f: jmp      0xdc9a
0xdc9a
0xdd11: nop      dword ptr [rax]
0xdd18: mov      edi, dword ptr [rip + 0x1d6f2]
0xdd1e: xor      eax, eax
0xdd20: call     0x1ac30
0xdd25: mov      dword ptr [rip + 0x1d6e5], 0
0xdd2f: jmp      0xdc4e
0xdc4e
0xdd34: nop      dword ptr [rax]
0xdd38: mov      dword ptr [rip + 0x1d6d2], 1
0xdd42: xor      eax, eax
0xdd44: call     0x1a570
0xdd49: mov      dword ptr [rip + 0x1d6c1], eax
0xdd4f: jmp      0xdbde
0xdbde
0xdd54: nop      dword ptr [rax]
0xdd58: xor      eax, eax

```

```

0xdd5a: mov     rdi, rbp
0xdd5d: call    0x1a380
0xdd62: xor     eax, eax
0xdd64: call    0x1ab40
0xdd69: jmp     0xdc80
0xdc80:
0xdd6e: nop
0xdd70: mov     qword ptr [rbx + 0x50], rsi
0xdd74: mov     rdi, qword ptr [rbx + 0x58]
0xdd78: call    0x1d690
0xdd7d: mov     qword ptr [rbx + 0x58], rax
0xdd81: mov     rbx, qword ptr [rip + 0x1d680]
0xdd88: mov     r12, qword ptr [rbp]
0xdd8c: mov     rdi, qword ptr [rbx + 0x40]
0xdd90: jmp     0xdc22
0xdc22:
0xdd95: nop     dword ptr [rax]
0xdd98: mov     rax, qword ptr [rip + 0x1d669]
0xdd9f: mov     rdi, qword ptr [rax + 0x40]
0xdda3: add     rdi, qword ptr [rax + 0x58]
0xdda7: xor     eax, eax
0xdda9: call    0x1a380
0xddae: jmp     0xdc80
0xdc80:
0xddb3: nop     word ptr cs:[rax + rax]
0xddbd: nop     dword ptr [rax]
0xddc0: endbr64
0xddc4: mov     eax, dword ptr [rip + 0x1be96]
0xddca: push    rbp
0xddcb: mov     rbp, rdi
0xddce: lea     rdi, [rip + 0x1be73]
0xddd5: test    eax, eax
0xddd7: lea     rax, [rip + 0x1be52]
0xddde: cmove   rdi, rax
0xdde2: call    0x17c50
0xdde7: mov     edx, 9
0xddec: lea     rsi, [rip + 0x1c4ad]
0xddf3: lea     rdi, [rip + 0x1c466]
0xddfa: call    0x17f70
0xddff: mov     rdi, rbp
0xde02: call    0x1d7a0
0xde07: mov     rdi, rax
0xde0a: mov     rbp, rax
0xde0d: mov     qword ptr [rip + 0x1d794], rax
0xde14: call    0x49b0
0xde19: mov     rdi, rbp
0xde1c: mov     rsi, rax
0xde1f: call    0x184d0

```

```

0xde24: xor     eax, eax
0xde26: call    0xc590
0xde2b: lea     rdi, [rip + 0xd5ee]
0xde32: pop     rbp
0xde33: mov     qword ptr [rip + 0xd78e], rax
0xde3a: jmp     0x4a30
0x4a30
0xde3f: nop
0xde40: endbr64
0xde44: mov     rdi, qword ptr [rip + 0xd77d]
0xde4b: jmp     0xc5c0
0xc5c0
0xde50: endbr64
0xde54: push    rbp
0xde55: mov     rbp, rdi
0xde58: push    rbx
0xde59: mov     rbx, rsi
0xde5c: sub     rsp, 8
0xde60: mov     rdi, qword ptr [rip + 0xd749]
0xde67: test    rdi, rdi
0xde6a: je      0xde71
0xde6c: call    0x18d10
0xde71: mov     rdi, rbp
0xde74: call    0xd670
0xde79: mov     rdi, qword ptr [rip + 0xd748]
0xde80: mov     edx, ebx
0xde82: xor     esi, esi
0xde84: mov     qword ptr [rip + 0xd725], rax
0xde8b: add     rsp, 8
0xde8f: mov     r8d, ebp
0xde92: mov     ecx, 1
0xde97: pop     rbx
0xde98: pop     rbp
0xde99: jmp     0xc5e0
0xc5e0
0xde9e: nop
0xdea0: endbr64
0xdea4: push    rbp
0xdea5: mov     rbp, rsp
0xdea8: push    r15
0xdeaa: push    r14
0xdeac: push    r13
0xdeae: push    r12
0xdeb0: push    rbx
0xdeb1: sub     rsp, 0x68
0xdeb5: mov     rax, qword ptr fs:[0x28]
0xdebe: mov     qword ptr [rbp - 0x38], rax
0xdec2: xor     eax, eax

```

```

0xdec4: lea      rax, [rbp - 0x61]
0xdec8: mov      rdi, rax
0xdecb: mov      qword ptr [rbp - 0x80], rax
0xdecf: call     0xa6a0
0xded4: mov      rax, qword ptr [rip + 0x1d6ed]
0xdedb: mov      esi, 0x20
0xdee0: mov      rdi, qword ptr [rip + 0x1d6c9]
0xdee7: movsxd   rdx, dword ptr [rax + 0xc]
0xdeeb: call     0x4a50
0xdef0: mov      rdi, qword ptr [rip + 0x1d6c9]
0xdef7: call     0x49b0
0xdefc: lea      r14, [rax + 1]
0xdf00: mov      rdi, r14
0xdf03: call     0x1d670
0xdf08: mov      rcx, qword ptr [rip + 0x1d6b1]
0xdf0f: mov      r13, rax
0xdf12: movzx    eax, byte ptr [rcx]
0xdf15: test     al, al
0xdf17: je       0xe430
0xdf1d: xor      esi, esi
0xdf1f: xor      ebx, ebx
0xdf21: lea      r15, [rip + 0x13188]
0xdf28: xor      edx, edx
0xdf2a: mov      dword ptr [rbp - 0x74], esi
0xdf2d: jmp      0xdf48
0xdf48:
0xdf2f: nop
0xdf30: cmp      al, 9
0xdf32: je       0xdf80
0xdf34: add      dword ptr [rbp - 0x74], 1
0xdf38: mov      byte ptr [r13 + r12], al
0xdf3d: movsxd   rdx, ebx
0xdf40: movzx    eax, byte ptr [rcx + rdx]
0xdf44: test     al, al
0xdf46: je       0xdfc0
0xdf48: movsxd   r12, dword ptr [rbp - 0x74]
0xdf4c: add      ebx, 1
0xdf4f: cmp      al, 0x5c
0xdf51: jne      0xdf30
0xdf53: lea      rdx, [rcx + rdx + 1]
0xdf58: movzx    eax, byte ptr [rdx]
0xdf5b: test     al, al
0xdf5d: je       0xe320
0xdf63: sub      eax, 0x5c
0xdf66: cmp      al, 0x17
0xdf68: ja       0xe0d8
0xdf6e: movzx    eax, al
0xdf71: movsxd   rax, dword ptr [r15 + rax*4]

```

```

0xdf75: add    rax, r15
0xdf78: jmp     rax
rax
->->->                                indirect JMP

0x18d10: endbr64
0x18d14: test    rdi, rdi
0x18d17: je      0x18d20
0x18d19: jmp     0x4800
0x4800
0x18d1e: nop
0x18d20: lea     rdi, [rip + 0x9697]
0x18d27: jmp     0x8b00
0x8b00
0x18d2c: nop     dword ptr [rax]
0x18d30: endbr64
0x18d34: xor     r8d, r8d
0x18d37: test    rdx, rdx
0x18d3a: je      0x18d90
0x18d3c: push    r14
0x18d3e: push    r13
0x18d40: push    r12
0x18d42: mov     r12, rsi
0x18d45: push    rbp
0x18d46: mov     ebp, edi
0x18d48: push    rbx
0x18d49: mov     rbx, rdx
0x18d4c: call    0x4860
0x18d51: mov     r13d, dword ptr [rax]
0x18d54: mov     r14, rax
0x18d57: jmp     0x18d66
0x18d66
0x18d59: nop     dword ptr [rax]
0x18d60: cmp     dword ptr [r14], 4
0x18d64: jne     0x18d7d
0x18d66: mov     dword ptr [r14], r13d
0x18d69: mov     rdx, rbx
0x18d6c: mov     rsi, r12
0x18d6f: mov     edi, ebp
0x18d71: call    0x47a0
0x18d76: mov     r8d, eax
0x18d79: test    eax, eax
0x18d7b: js      0x18d60
0x18d7d: pop     rbx
0x18d7e: mov     eax, r8d
0x18d81: pop     rbp
0x18d82: pop     r12
0x18d84: pop     r13
0x18d86: pop     r14

```

```

0x18d88:      ret
0x18d89:      nop        dword ptr [rax]
0x18d90:      mov         eax, r8d
0x18d93:      ret
0x18d94:      nop        word ptr cs:[rax + rax]
0x18d9f:      nop
0x18da0:      endbr64
0x18da4:      xor         r8d, r8d
0x18da7:      test        rdx, rdx
0x18daa:      je          0x18e00
0x18dac:      push        r14
0x18dae:      push        r13
0x18db0:      push        r12
0x18db2:      mov         r12, rsi
0x18db5:      push        rbp
0x18db6:      mov         ebp, edi
0x18db8:      push        rbx
0x18db9:      mov         rbx, rdx
0x18dbc:      call        0x4860
0x18dc1:      mov         r13d, dword ptr [rax]
0x18dc4:      mov         r14, rax
0x18dc7:      jmp         0x18dd6
0x18dd6:
0x18dc9:      nop        dword ptr [rax]
0x18dd0:      cmp         dword ptr [r14], 4
0x18dd4:      jne         0x18ded
0x18dd6:      mov         dword ptr [r14], r13d
0x18dd9:      mov         rdx, rbx
0x18ddc:      mov         rsi, r12
0x18ddf:      mov         edi, ebp
0x18de1:      call        0x4790
0x18de6:      mov         r8d, eax
0x18de9:      test        eax, eax
0x18deb:      js          0x18dd0
0x18ded:      pop         rbx
0x18dee:      mov         eax, r8d
0x18df1:      pop         rbp
0x18df2:      pop         r12
0x18df4:      pop         r13
0x18df6:      pop         r14
0x18df8:      ret
0x18df9:      nop        dword ptr [rax]
0x18e00:      mov         eax, r8d
0x18e03:      ret
0x18e04:      nop        word ptr cs:[rax + rax]
0x18e0f:      nop
0x18e10:      endbr64
0x18e14:      xor         r8d, r8d

```

```

0x18e17:    test    rdx, rdx
0x18e1a:    je      0x18e70
0x18e1c:    push    r14
0x18e1e:    push    r13
0x18e20:    push    r12
0x18e22:    mov     r12, rsi
0x18e25:    push    rbp
0x18e26:    mov     rbp, rdi
0x18e29:    push    rbx
0x18e2a:    mov     rbx, rdx
0x18e2d:    call    0x4860
0x18e32:    mov     r13d, dword ptr [rax]
0x18e35:    mov     r14, rax
0x18e38:    jmp     0x18e46
0x18e46:
0x18e3a:    nop     word ptr [rax + rax]
0x18e40:    cmp     dword ptr [r14], 4
0x18e44:    jne     0x18e5e
0x18e46:    mov     dword ptr [r14], r13d
0x18e49:    mov     rdx, rbx
0x18e4c:    mov     rsi, r12
0x18e4f:    mov     rdi, rbp
0x18e52:    call    0x4910
0x18e57:    mov     r8d, eax
0x18e5a:    test    eax, eax
0x18e5c:    js      0x18e40
0x18e5e:    pop     rbx
0x18e5f:    mov     eax, r8d
0x18e62:    pop     rbp
0x18e63:    pop     r12
0x18e65:    pop     r13
0x18e67:    pop     r14
0x18e69:    ret
0x18e6a:    nop     word ptr [rax + rax]
0x18e70:    mov     eax, r8d
0x18e73:    ret
0x18e74:    nop     word ptr cs:[rax + rax]
0x18e7f:    nop
0x18e80:    endbr64
0x18e84:    push    r13
0x18e86:    mov     r13, rdi
0x18e89:    push    r12
0x18e8b:    push    rbp
0x18e8c:    push    rbx
0x18e8d:    mov     ebx, 0x64
0x18e92:    sub     rsp, 0x18
0x18e96:    jmp     0x18eaa
0x18eaa:

```

```

0x18e98:      nop      dword ptr [rax + rax]
0x18ea0:      mov      rdi, rbp
0x18ea3:      add      ebx, ebx
0x18ea5:      call     0x18d10
0x18eaa:      movsxd   r12, ebx
0x18ead:      mov      rdi, r12
0x18eb0:      call     0x1d670
0x18eb5:      mov      rdx, r12
0x18eb8:      mov      rdi, r13
0x18ebb:      mov      rbp, rax
0x18ebe:      mov      rsi, rax
0x18ec1:      xor      eax, eax
0x18ec3:      call     0x18e10
0x18ec8:      cmp      ebx, eax
0x18eca:      jle      0x18ea0
0x18ecc:      mov      rdi, rbp
0x18ecf:      mov      dword ptr [rsp + 0xc], eax
0x18ed3:      call     0x18d10
0x18ed8:      mov      eax, dword ptr [rsp + 0xc]
0x18edc:      add      rsp, 0x18
0x18ee0:      pop      rbx
0x18ee1:      pop      rbp
0x18ee2:      pop      r12
0x18ee4:      pop      r13
0x18ee6:      ret
0x18ee7:      nop      word ptr [rax + rax]
0x18ef0:      endbr64
0x18ef4:      push     r13
0x18ef6:      push     r12
0x18ef8:      mov      r12d, edi
0x18efb:      push     rbp
0x18efc:      mov      rbp, rsi
0x18eff:      push     rbx
0x18f00:      sub      rsp, 8
0x18f04:      call     0x4860
0x18f09:      mov      r13d, dword ptr [rax]
0x18f0c:      mov      rbx, rax
0x18f0f:      jmp      0x18f1d
0x18f1d:
0x18f11:      nop      dword ptr [rax]
0x18f18:      cmp      dword ptr [rbx], 4
0x18f1b:      jne      0x18f2f
0x18f1d:      mov      dword ptr [rbx], r13d
0x18f20:      mov      rsi, rbp
0x18f23:      mov      edi, r12d
0x18f26:      call     0x4eb0
0x18f2b:      test     eax, eax
0x18f2d:      js       0x18f18

```



```

0x18f2f:      add     rsp, 8
0x18f33:      pop     rbx
0x18f34:      pop     rbp
0x18f35:      pop     r12
0x18f37:      pop     r13
0x18f39:      ret
0x18f3a:      nop     word ptr [rax + rax]
0x18f40:      endbr64
0x18f44:      push    r13
0x18f46:      push    r12
0x18f48:      mov     r12, rdi
0x18f4b:      push    rbp
0x18f4c:      mov     rbp, rsi
0x18f4f:      push    rbx
0x18f50:      sub     rsp, 8
0x18f54:      call    0x4860
0x18f59:      mov     r13d, dword ptr [rax]
0x18f5c:      mov     rbx, rax
0x18f5f:      jmp     0x18f6d
0x18f6d:
0x18f61:      nop     dword ptr [rax]
0x18f68:      cmp     dword ptr [rbx], 4
0x18f6b:      jne     0x18f7f
0x18f6d:      mov     dword ptr [rbx], r13d
0x18f70:      mov     rsi, rbp
0x18f73:      mov     rdi, r12
0x18f76:      call    0x4b90
0x18f7b:      test    eax, eax
0x18f7d:      js      0x18f68
0x18f7f:      add     rsp, 8
0x18f83:      pop     rbx
0x18f84:      pop     rbp
0x18f85:      pop     r12
0x18f87:      pop     r13
0x18f89:      ret
0x18f8a:      nop     word ptr [rax + rax]
0x18f90:      endbr64
0x18f94:      push    r13
0x18f96:      push    r12
0x18f98:      mov     r12, rdi
0x18f9b:      push    rbp
0x18f9c:      mov     rbp, rsi
0x18f9f:      push    rbx
0x18fa0:      sub     rsp, 8
0x18fa4:      call    0x4860
0x18fa9:      mov     r13d, dword ptr [rax]
0x18fac:      mov     rbx, rax
0x18faf:      jmp     0x18fbd

```

```

0x18fbd
0x18fb1:    nop     dword ptr [rax]
0x18fb8:    cmp     dword ptr [rbx], 4
0x18fbb:    jne     0x18fcf
0x18fbd:    mov     dword ptr [rbx], r13d
0x18fc0:    mov     rsi, rbp
0x18fc3:    mov     rdi, r12
0x18fc6:    call    0x4ad0
0x18fcb:    test    eax, eax
0x18fcd:    js      0x18fb8
0x18fcf:    add     rsp, 8
0x18fd3:    pop     rbx
0x18fd4:    pop     rbp
0x18fd5:    pop     r12
0x18fd7:    pop     r13
0x18fd9:    ret
0x18fda:    nop     word ptr [rax + rax]
0x18fe0:    endbr64
0x18fe4:    push    r12
0x18fe6:    push    rbp
0x18fe7:    push    rbx
0x18fe8:    mov     ebx, 0x40
0x18fed:    call    0x4860
0x18ff2:    mov     edi, 0x40
0x18ff7:    mov     dword ptr [rax], 0
0x18ffd:    mov     r12, rax
0x19000:    call    0x1d670
0x19005:    mov     rbp, rax
0x19008:    jmp     0x19031
0x19031
0x1900a:    nop     word ptr [rax + rax]
0x19010:    cmp     dword ptr [r12], 0x22
0x19015:    jne     0x19050
0x19017:    add     rbx, 0x40
0x1901b:    mov     rdi, rbp
0x1901e:    mov     rsi, rbx
0x19021:    call    0x1d690
0x19026:    mov     dword ptr [r12], 0
0x1902e:    mov     rbp, rax
0x19031:    mov     rsi, rbx
0x19034:    mov     rdi, rbp
0x19037:    call    0x4a80
0x1903c:    test    rax, rax
0x1903f:    je      0x19010
0x19041:    mov     rax, rbp
0x19044:    pop     rbx
0x19045:    pop     rbp
0x19046:    pop     r12

```

```

0x19048:      ret
0x19049:      nop      dword ptr [rax]
0x19050:      mov      rdi, rbp
0x19053:      xor      ebp, ebp
0x19055:      call     0x18d10
0x1905a:      mov      rax, rbp
0x1905d:      pop      rbx
0x1905e:      pop      rbp
0x1905f:      pop      r12
0x19061:      ret
0x19062:      nop      word ptr cs:[rax + rax]
0x1906d:      nop      dword ptr [rax]
0x19070:      endbr64
0x19074:      push     rbp
0x19075:      mov      esi, 0x2f
0x1907a:      mov      rbp, rdi
0x1907d:      push     rbx
0x1907e:      sub      rsp, 8
0x19082:      call     0x4a20
0x19087:      test     rax, rax
0x1908a:      je       0x190a8
0x1908c:      mov      rbx, rax
0x1908f:      mov      rdi, rbp
0x19092:      call     0x1d7a0
0x19097:      sub      rbx, rbp
0x1909a:      mov      byte ptr [rax + rbx], 0x30
0x1909e:      add      rsp, 8
0x190a2:      pop      rbx
0x190a3:      pop      rbp
0x190a4:      ret
0x190a5:      nop      dword ptr [rax]
0x190a8:      add      rsp, 8
0x190ac:      lea      rdi, [rip + 0x862b]
0x190b3:      pop      rbx
0x190b4:      pop      rbp
0x190b5:      jmp      0x1d7a0
0x1d7a0
0x190ba:      nop      word ptr [rax + rax]
0x190c0:      endbr64
0x190c4:      push     rbx
0x190c5:      mov      rbx, rdi
0x190c8:      call     0x49b0
0x190cd:      lea      rax, [rbx + rax - 1]
0x190d2:      cmp      byte ptr [rax], 0x2f
0x190d5:      jne      0x190da
0x190d7:      mov      byte ptr [rax], 0
0x190da:      mov      rdi, rbx
0x190dd:      mov      esi, 0x2f

```

```

0x190e2:      call    0x4a20
0x190e7:      lea     rdx, [rax + 1]
0x190eb:      test   rax, rax
0x190ee:      mov     rax, rbx
0x190f1:      pop     rbx
0x190f2:      cmovne  rax, rdx
0x190f6:      ret
0x190f7:      nop     word ptr [rax + rax]
0x19100:      endbr64
0x19104:      push    rbx
0x19105:      mov     esi, 0x10
0x1910a:      mov     ebx, edi
0x1910c:      mov     edi, 1
0x19111:      call    0x1d740
0x19116:      mov     dword ptr [rax + 8], ebx
0x19119:      pop     rbx
0x1911a:      ret
0x1911b:      nop     dword ptr [rax + rax]
0x19120:      endbr64
0x19124:      test    rdi, rdi
0x19127:      je      0x19148
0x19129:      push    rbp
0x1912a:      mov     rbp, rdi
0x1912d:      mov     rdi, qword ptr [rdi]
0x19130:      test    rdi, rdi
0x19133:      je      0x1913a
0x19135:      call    0x18d10
0x1913a:      mov     rdi, rbp
0x1913d:      pop     rbp
0x1913e:      jmp     0x18d10
0x18d10
0x19143:      nop     dword ptr [rax + rax]
0x19148:      ret
0x19149:      nop     dword ptr [rax]
0x19150:      endbr64
0x19154:      push    rbp
0x19155:      mov     rbp, rsi
0x19158:      push    rbx
0x19159:      mov     rbx, rdi
0x1915c:      sub     rsp, 8
0x19160:      mov     eax, dword ptr [rdi + 0xc]
0x19163:      lea     esi, [rax + 1]
0x19166:      mov     dword ptr [rdi + 0xc], esi
0x19169:      imul    esi, dword ptr [rdi + 8]
0x1916d:      mov     rdi, qword ptr [rdi]
0x19170:      movsxd  rsi, esi
0x19173:      call    0x1d690
0x19178:      mov     ecx, dword ptr [rbx + 0xc]

```

```

0x1917b:      movsxd  rdx, dword ptr [rbx + 8]
0x1917f:      mov     rsi, rbp
0x19182:      mov     qword ptr [rbx], rax
0x19185:      add     rsp, 8
0x19189:      sub     ecx, 1
0x1918c:      pop     rbx
0x1918d:      pop     rbp
0x1918e:      imul    ecx, edx
0x19191:      movsxd  rcx, ecx
0x19194:      lea     rdi, [rax + rcx]
0x19198:      jmp     0x4bb0
0x4bb0
0x1919d:      nop     dword ptr [rax]
0x191a0:      endbr64
0x191a4:      mov     eax, dword ptr [rdi + 0xc]
0x191a7:      test    eax, eax
0x191a9:      je      0x191f8
0x191ab:      push    rbp
0x191ac:      sub     eax, 1
0x191af:      mov     rbp, rsi
0x191b2:      push    rbx
0x191b3:      mov     rbx, rdi
0x191b6:      sub     rsp, 8
0x191ba:      movsxd  rdx, dword ptr [rdi + 8]
0x191be:      mov     dword ptr [rdi + 0xc], eax
0x191c1:      mov     rsi, rdx
0x191c4:      imul    esi, eax
0x191c7:      movsxd  rsi, esi
0x191ca:      add     rsi, qword ptr [rdi]
0x191cd:      mov     rdi, rbp
0x191d0:      call    0x4bb0
0x191d5:      mov     esi, dword ptr [rbx + 0xc]
0x191d8:      imul    esi, dword ptr [rbx + 8]
0x191dc:      mov     rdi, qword ptr [rbx]
0x191df:      movsxd  rsi, esi
0x191e2:      call    0x1d690
0x191e7:      mov     qword ptr [rbx], rax
0x191ea:      add     rsp, 8
0x191ee:      mov     rax, rbp
0x191f1:      pop     rbx
0x191f2:      pop     rbp
0x191f3:      ret
0x191f4:      nop     dword ptr [rax]
0x191f8:      xor     eax, eax
0x191fa:      ret
0x191fb:      nop     dword ptr [rax + rax]
0x19200:      endbr64
0x19204:      mov     ecx, dword ptr [rdi + 0xc]

```

```

0x19207:      cmp     ecx, edx
0x19209:      jl      0x19240
0x1920b:      test    ecx, ecx
0x1920d:      je       0x19240
0x1920f:      sub      rsp, 8
0x19213:      movsxd   r9, dword ptr [rdi + 8]
0x19217:      sub      ecx, edx
0x19219:      mov      r8, rsi
0x1921c:      imul     ecx, r9d
0x19220:      mov      rdx, r9
0x19223:      movsxd   rsi, ecx
0x19226:      add      rsi, qword ptr [rdi]
0x19229:      mov      rdi, r8
0x1922c:      call     0x4bb0
0x19231:      add      rsp, 8
0x19235:      ret
0x19236:      nop      word ptr cs:[rax + rax]
0x19240:      xor      eax, eax
0x19242:      ret
0x19243:      nop      word ptr cs:[rax + rax]
0x1924e:      nop
0x19250:      endbr64
0x19254:      push     rbx
0x19255:      mov      rbx, rdi
0x19258:      mov      dword ptr [rdi + 0xc], esi
0x1925b:      imul     esi, dword ptr [rdi + 8]
0x1925f:      mov      rdi, qword ptr [rdi]
0x19262:      movsxd   rsi, esi
0x19265:      call     0x1d690
0x1926a:      mov      qword ptr [rbx], rax
0x1926d:      pop      rbx
0x1926e:      ret
0x1926f:      nop
0x19270:      endbr64
0x19274:      mov      eax, dword ptr [rdi + 0xc]
0x19277:      ret
0x19278:      nop      dword ptr [rax + rax]
0x19280:      push     r12
0x19282:      mov      r12, rdi
0x19285:      mov      edi, 2
0x1928a:      push     rbp
0x1928b:      push     rbx
0x1928c:      mov      ebx, esi
0x1928e:      xor      esi, esi
0x19290:      call     0x4b50
0x19295:      xor      esi, esi
0x19297:      mov      edi, 3
0x1929c:      call     0x4b50

```

```

0x192a1:    test    ebx, ebx
0x192a3:    jne     0x192e0
0x192a5:    mov     r9, qword ptr [rip + 0x115b4]
0x192ac:    xor     r8d, r8d
0x192af:    mov     rcx, r12
0x192b2:    xor     eax, eax
0x192b4:    lea     rdi, [rip + 0x7363]
0x192bb:    lea     rdx, [rip + 0x911e]
0x192c2:    lea     rsi, [rip + 0x735a]
0x192c9:    call    0x4e30
0x192ce:    mov     edi, 0x7f
0x192d3:    call    0x48a0
0x192d8:    nop     dword ptr [rax + rax]
0x192e0:    mov     edi, 1
0x192e5:    lea     rbp, [rip + 0x90ed]
0x192ec:    call    0x4ab0
0x192f1:    mov     edi, 2
0x192f6:    call    0x4ab0
0x192fb:    mov     rdi, qword ptr [rip + 0x127ce]
0x19302:    mov     rsi, rbp
0x19305:    call    0x4da0
0x1930a:    mov     rdi, qword ptr [rip + 0x127b7]
0x19311:    mov     rsi, rbp
0x19314:    call    0x4da0
0x19319:    mov     edx, 4
0x1931e:    lea     rsi, [rip + 0x90b6]
0x19325:    mov     rdi, r12
0x19328:    call    0x4890
0x1932d:    test    eax, eax
0x1932f:    je      0x192a5
0x19335:    xor     edi, edi
0x19337:    call    0x4ab0
0x1933c:    jmp     0x192a5
0x192a5:
0x19341:    nop     word ptr cs:[rax + rax]
0x1934c:    nop     dword ptr [rax]
0x19350:    endbr64
0x19354:    push    r12
0x19356:    push    rbp
0x19357:    push    rbx
0x19358:    sub     rsp, 0x10
0x1935c:    mov     rax, qword ptr fs:[0x28]
0x19365:    mov     qword ptr [rsp + 8], rax
0x1936a:    xor     eax, eax
0x1936c:    test    rdi, rdi
0x1936f:    je      0x193d0
0x19371:    mov     rbp, rdi
0x19374:    mov     r12d, esi

```

```

0x19377:      call    0x4ed0
0x1937c:      mov     ebx, eax
0x1937e:      test    eax, eax
0x19380:      js      0x193c0
0x19382:      je      0x193dc
0x19384:      lea     rbp, [rsp + 4]
0x19389:      nop     dword ptr [rax]
0x19390:      mov     rdi, rbp
0x19393:      call    0x4ec0
0x19398:      cmp     eax, ebx
0x1939a:      jne     0x19390
0x1939c:      mov     eax, dword ptr [rsp + 4]
0x193a0:      mov     rdx, qword ptr [rsp + 8]
0x193a5:      sub     rdx, qword ptr fs:[0x28]
0x193ae:      jne     0x193d7
0x193b0:      add     rsp, 0x10
0x193b4:      pop     rbx
0x193b5:      pop     rbp
0x193b6:      pop     r12
0x193b8:      ret
0x193b9:      nop     dword ptr [rax]
0x193c0:      mov     eax, 0xffffffff
0x193c5:      jmp     0x193a0
0x193a0
0x193c7:      nop     word ptr [rax + rax]
0x193d0:      mov     eax, 1
0x193d5:      jmp     0x193a0
0x193a0
0x193d7:      call    0x49d0
0x193dc:      mov     esi, r12d
0x193df:      mov     rdi, rbp
0x193e2:      xor     eax, eax
0x193e4:      call    0x19280
0x193e9:      nop     dword ptr [rax]
0x193f0:      endbr64
0x193f4:      push    r13
0x193f6:      mov     r13, rdi
0x193f9:      push    r12
0x193fb:      push    rbp
0x193fc:      push    rbx
0x193fd:      sub     rsp, 0x18
0x19401:      mov     rax, qword ptr fs:[0x28]
0x1940a:      mov     qword ptr [rsp + 8], rax
0x1940f:      xor     eax, eax
0x19411:      test    esi, esi
0x19413:      je      0x19488
0x19415:      mov     edi, 1
0x1941a:      mov     ebp, esi

```



```

0x1941c:      call    0x18b70
0x19421:      test    r13, r13
0x19424:      je      0x19520
0x1942a:      call    0x4ed0
0x1942f:      mov     r12d, eax
0x19432:      test    eax, eax
0x19434:      js      0x19550
0x1943a:      je      0x19560
0x19440:      lea     rbx, [rsp + 4]
0x19445:      nop     dword ptr [rax]
0x19448:      mov     rdi, rbx
0x1944b:      call    0x4ec0
0x19450:      cmp     r12d, eax
0x19453:      jne     0x19448
0x19455:      mov     r12d, dword ptr [rsp + 4]
0x1945a:      xor     edi, edi
0x1945c:      call    0x18b70
0x19461:      mov     rax, qword ptr [rsp + 8]
0x19466:      sub     rax, qword ptr fs:[0x28]
0x1946f:      jne     0x1955b
0x19475:      add     rsp, 0x18
0x19479:      mov     eax, r12d
0x1947c:      pop     rbx
0x1947d:      pop     rbp
0x1947e:      pop     r12
0x19480:      pop     r13
0x19482:      ret
0x19483:      nop     dword ptr [rax + rax]
0x19488:      xor     edi, edi
0x1948a:      call    0x9b00
0x1948f:      xor     eax, eax
0x19491:      call    0xa6c0
0x19496:      mov     rdi, qword ptr [rip + 0x1142b]
0x1949d:      call    0xbb40
0x194a2:      xor     edi, edi
0x194a4:      call    0x18b90
0x194a9:      test    r13, r13
0x194ac:      je      0x19530
0x194b2:      call    0x4ed0
0x194b7:      mov     ebp, eax
0x194b9:      test    eax, eax
0x194bb:      js      0x19540
0x194c1:      je      0x1956c
0x194c7:      lea     rbx, [rsp + 4]
0x194cc:      nop     dword ptr [rax]
0x194d0:      mov     rdi, rbx
0x194d3:      call    0x4ec0
0x194d8:      cmp     ebp, eax

```

```

0x194da:    jne     0x194d0
0x194dc:    mov     r12d, dword ptr [rsp + 4]
0x194e1:    mov     edi, 1
0x194e6:    call    0x18b90
0x194eb:    mov     edx, 2
0x194f0:    mov     edi, 1
0x194f5:    lea     rsi, [rip + 0x8ee7]
0x194fc:    call    0x18da0
0x19501:    mov     edi, 1
0x19506:    call    0x9b00
0x1950b:    xor     eax, eax
0x1950d:    call    0xa6c0
0x19512:    xor     edi, edi
0x19514:    call    0x7890
0x19519:    jmp     0x19461
0x19461
0x1951e:    nop
0x19520:    mov     r12d, 1
0x19526:    jmp     0x1945a
0x1945a
0x1952b:    nop     dword ptr [rax + rax]
0x19530:    mov     r12d, 1
0x19536:    jmp     0x194e1
0x194e1
0x19538:    nop     dword ptr [rax + rax]
0x19540:    mov     r12d, 0xffffffff
0x19546:    jmp     0x194e1
0x194e1
0x19548:    nop     dword ptr [rax + rax]
0x19550:    mov     r12d, 0xffffffff
0x19556:    jmp     0x1945a
0x1945a
0x1955b:    call    0x49d0
0x19560:    mov     esi, ebp
0x19562:    mov     rdi, r13
0x19565:    xor     eax, eax
0x19567:    call    0x19280
0x1956c:    xor     esi, esi
0x1956e:    mov     rdi, r13
0x19571:    xor     eax, eax
0x19573:    call    0x19280
0x19578:    nop     dword ptr [rax + rax]
0x19580:    endbr64
0x19584:    mov     rdi, qword ptr [rip + 0x12545]
0x1958b:    test    rdi, rdi
0x1958e:    je      0x195b8
0x19590:    sub     rsp, 8
0x19594:    call    0x4870

```

```

0x19599:      mov     rdi, qword ptr [rip + 0x12528]
0x195a0:      test    rdi, rdi
0x195a3:      je       0x195b0
0x195a5:      add     rsp, 8
0x195a9:      jmp      0x4870
0x4870
0x195ae:      nop
0x195b0:      add     rsp, 8
0x195b4:      ret
0x195b5:      nop     dword ptr [rax]
0x195b8:      mov     rdi, qword ptr [rip + 0x12509]
0x195bf:      test    rdi, rdi
0x195c2:      je       0x195d0
0x195c4:      jmp      0x4870
0x4870
0x195c9:      nop     dword ptr [rax]
0x195d0:      ret
0x195d1:      nop     word ptr cs:[rax + rax]
0x195dc:      nop     dword ptr [rax]
0x195e0:      endbr64
0x195e4:      push    r12
0x195e6:      lea     rsi, [rip + 0x89dc]
0x195ed:      push    rbp
0x195ee:      mov     rbp, rdi
0x195f1:      sub     rsp, 8
0x195f5:      mov     rdi, qword ptr [rip + 0x124cc]
0x195fc:      call    0x4da0
0x19601:      test    rax, rax
0x19604:      je       0x19676
0x19606:      mov     edi, 0x801
0x1960b:      mov     r12, rax
0x1960e:      call    0x1d670
0x19613:      mov     rbp, rax
0x19616:      jmp      0x1964a
0x1964a
0x19618:      nop     dword ptr [rax + rax]
0x19620:      mov     rdi, rbp
0x19623:      call    0x49b0
0x19628:      cdqe
0x1962a:      lea     rax, [rbp + rax - 1]
0x1962f:      cmp     byte ptr [rax], 0xa
0x19632:      jne     0x19637
0x19634:      mov     byte ptr [rax], 0
0x19637:      xor     esi, esi
0x19639:      mov     edx, 0xa
0x1963e:      mov     rdi, rbp
0x19641:      call    0x8070
0x19646:      test    al, al

```

```

0x19648:      je      0x1965f
0x1964a:      mov     rdx, r12
0x1964d:      mov     esi, 0x801
0x19652:      mov     rdi, rbp
0x19655:      call    0x4b00
0x1965a:      test    rax, rax
0x1965d:      jne     0x19620
0x1965f:      mov     rdi, rbp
0x19662:      call    0x18d10
0x19667:      add     rsp, 8
0x1966b:      mov     rdi, r12
0x1966e:      pop     rbp
0x1966f:      pop     r12
0x19671:      jmp     0x4940
0x4940
0x19676:      mov     rdi, rbp
0x19679:      call    0x49b0
0x1967e:      mov     rdi, qword ptr [rip + 0x12443]
0x19685:      mov     r12, rax
0x19688:      call    0x49b0
0x1968d:      lea     rdi, [r12 + rax + 0x20]
0x19692:      call    0x1d670
0x19697:      mov     r9, qword ptr [rip + 0x1242a]
0x1969e:      mov     r8, rbp
0x196a1:      or      rdx, 0xffffffffffffffff
0x196a5:      mov     r12, rax
0x196a8:      mov     rdi, rax
0x196ab:      mov     esi, 1
0x196b0:      xor     eax, eax
0x196b2:      lea     rcx, [rip + 0x8d2d]
0x196b9:      call    0x4f10
0x196be:      mov     rdi, r12
0x196c1:      mov     edx, 0x1e
0x196c6:      xor     esi, esi
0x196c8:      call    0x8070
0x196cd:      pop     rax
0x196ce:      mov     rdi, r12
0x196d1:      pop     rbp
0x196d2:      pop     r12
0x196d4:      jmp     0x18d10
0x18d10
0x196d9:      nop     dword ptr [rax]
0x196e0:      endbr64
0x196e4:      push    r15
0x196e6:      push    r14
0x196e8:      push    r13
0x196ea:      push    r12
0x196ec:      push    rbp

```

```

0x196ed:    push    rbx
0x196ee:    sub     rsp, 8
0x196f2:    test    rdi, rdi
0x196f5:    je      0x19878
0x196fb:    mov     rbp, rdi
0x196fe:    call    0x49b0
0x19703:    lea     rdi, [rax + 1]
0x19707:    call    0x1d670
0x1970c:    mov     rsi, rbp
0x1970f:    mov     rdi, rax
0x19712:    call    0x48b0
0x19717:    cmp     byte ptr [rax], 0x7e
0x1971a:    mov     rbp, rax
0x1971d:    mov     r12, rax
0x19720:    jne     0x19800
0x19726:    movzx   eax, byte ptr [rax + 1]
0x1972a:    lea     r13, [rbp + 1]
0x1972e:    test    al, al
0x19730:    je      0x19818
0x19736:    cmp     al, 0x2f
0x19738:    je      0x19818
0x1973e:    mov     rdi, rbp
0x19741:    call    0x49b0
0x19746:    mov     rdi, rax
0x19749:    call    0x1d670
0x1974e:    movzx   ecx, byte ptr [rbp + 1]
0x19752:    mov     r14, rax
0x19755:    cmp     cl, 0x2f
0x19758:    je      0x19938
0x1975e:    test    cl, cl
0x19760:    je      0x19938
0x19766:    mov     edx, 2
0x1976b:    nop     dword ptr [rax + rax]
0x19770:    mov     byte ptr [r14 + rdx - 2], cl
0x19775:    movzx   ecx, byte ptr [rbp + rdx]
0x1977a:    lea     r13, [rbp + rdx]
0x1977f:    mov     rax, rdx
0x19782:    add     rdx, 1
0x19786:    test    cl, cl
0x19788:    je      0x1978f
0x1978a:    cmp     cl, 0x2f
0x1978d:    jne     0x19770
0x1978f:    lea     rax, [r14 + rax - 1]
0x19794:    mov     byte ptr [rax], 0
0x19797:    mov     rdi, r14
0x1979a:    call    0x4b60
0x1979f:    mov     rbx, rax
0x197a2:    test    rax, rax

```

```

0x197a5:      je      0x19880
0x197ab:      mov     rdi, qword ptr [rax + 0x20]
0x197af:      call    0x49b0
0x197b4:      mov     rdi, r13
0x197b7:      mov     r15, rax
0x197ba:      call    0x49b0
0x197bf:      lea     edi, [r15 + 1]
0x197c3:      movsxd  rdi, edi
0x197c6:      add     rdi, rax
0x197c9:      call    0x1d670
0x197ce:      mov     rsi, qword ptr [rbx + 0x20]
0x197d2:      mov     rdi, rax
0x197d5:      mov     r12, rax
0x197d8:      call    0x48b0
0x197dd:      movsxd  rdi, r15d
0x197e0:      mov     rsi, r13
0x197e3:      add     rdi, r12
0x197e6:      call    0x48b0
0x197eb:      mov     rdi, rbp
0x197ee:      call    0x18d10
0x197f3:      call    0x4c70
0x197f8:      mov     rdi, r14
0x197fb:      call    0x18d10
0x19800:      add     rsp, 8
0x19804:      mov     rax, r12
0x19807:      pop     rbx
0x19808:      pop     rbp
0x19809:      pop     r12
0x1980b:      pop     r13
0x1980d:      pop     r14
0x1980f:      pop     r15
0x19811:      ret
0x19812:      nop     word ptr [rax + rax]
0x19818:      lea     rdi, [rip + 0x6d41]
0x1981f:      call    0x47e0
0x19824:      mov     r14, rax
0x19827:      test    rax, rax
0x1982a:      je      0x198f8
0x19830:      mov     rdi, r14
0x19833:      call    0x49b0
0x19838:      mov     rdi, r13
0x1983b:      mov     rbx, rax
0x1983e:      call    0x49b0
0x19843:      movsxd  rbx, ebx
0x19846:      lea     rdi, [rbx + rax + 1]
0x1984b:      call    0x1d670
0x19850:      mov     rsi, r14
0x19853:      mov     r12, rax

```

```

0x19856:      mov     rdi, rax
0x19859:      call    0x48b0
0x1985e:      lea     rdi, [r12 + rbx]
0x19862:      mov     rsi, r13
0x19865:      call    0x48b0
0x1986a:      mov     rdi, rbp
0x1986d:      call    0x18d10
0x19872:      jmp     0x19800
0x19800
0x19874:      nop     dword ptr [rax]
0x19878:      xor     r12d, r12d
0x1987b:      jmp     0x19800
0x19800
0x1987d:      nop     dword ptr [rax]
0x19880:      mov     rdx, qword ptr [rip + 0x12251]
0x19887:      test    rdx, rdx
0x1988a:      je      0x197f3
0x19890:      mov     rdi, r14
0x19893:      xor     eax, eax
0x19895:      call    rdx
0x19897:      mov     r15, rax
0x1989a:      test    rax, rax
0x1989d:      je      0x197f3
0x198a3:      mov     rdi, rax
0x198a6:      call    0x49b0
0x198ab:      mov     rdi, r13
0x198ae:      mov     rbx, rax
0x198b1:      call    0x49b0
0x198b6:      lea     edi, [rbx + 1]
0x198b9:      movsxd  rdi, edi
0x198bc:      add     rdi, rax
0x198bf:      call    0x1d670
0x198c4:      mov     rsi, r15
0x198c7:      mov     rdi, rax
0x198ca:      mov     r12, rax
0x198cd:      call    0x48b0
0x198d2:      movsxd  rdi, ebx
0x198d5:      mov     rsi, r13
0x198d8:      add     rdi, r12
0x198db:      call    0x48b0
0x198e0:      mov     rdi, r15
0x198e3:      call    0x18d10
0x198e8:      mov     rdi, rbp
0x198eb:      call    0x18d10
0x198f0:      jmp     0x197f3
0x197f3
0x198f5:      nop     dword ptr [rax]
0x198f8:      call    0x49e0

```

```

0x198fd:    mov     edi, eax
0x198ff:    call   0x4970
0x19904:    test   rax, rax
0x19907:    je     0x19916
0x19909:    mov     r14, qword ptr [rax + 0x20]
0x1990d:    test   r14, r14
0x19910:    jne    0x19830
0x19916:    mov     rdi, r13
0x19919:    call   0x49b0
0x1991e:    lea     rdi, [rax + 1]
0x19922:    call   0x1d670
0x19927:    mov     r12, rax
0x1992a:    mov     rdi, rax
0x1992d:    jmp     0x19862
0x19862:
0x19932:    nop     word ptr [rax + rax]
0x19938:    mov     rax, r14
0x1993b:    jmp     0x19794
0x19794:
0x19940:    endbr64
0x19944:    push    r15
0x19946:    push    r14
0x19948:    mov     r14, rdi
0x1994b:    push    r13
0x1994d:    push    r12
0x1994f:    push    rbp
0x19950:    push    rbx
0x19951:    xor     ebx, ebx
0x19953:    sub     rsp, 0x48
0x19957:    mov     dword ptr [rsp + 0x1c], 0
0x1995f:    mov     qword ptr [rsp + 0x10], 0
0x19968:    nop     dword ptr [rax + rax]
0x19970:    mov     r15, qword ptr [rip + 0x10e31]
0x19977:    mov     rdi, r14
0x1997a:    mov     qword ptr [rsp + 8], r15
0x1997f:    call   0x49b0
0x19984:    mov     qword ptr [rsp + 0x20], rax
0x19989:    mov     rcx, rax
0x1998c:    movzx   eax, byte ptr [r14]
0x19990:    test    al, al
0x19992:    je     0x19bd0
0x19998:    cmp     al, 0x7e
0x1999a:    je     0x19bd0
0x199a0:    mov     dword ptr [rsp + 0x18], ecx
0x199a4:    mov     rax, rcx
0x199a7:    test    r15, r15
0x199aa:    je     0x19c50
0x199b0:    mov     edx, dword ptr [rsp + 0x20]

```



```

0x199b4:    test    edx, edx
0x199b6:    jle     0x19c4a
0x199bc:    mov     rax, qword ptr [rsp + 8]
0x199c1:    mov     dword ptr [rsp + 0x3c], ebx
0x199c5:    xor     r12d, r12d
0x199c8:    mov     rax, qword ptr [rax]
0x199cb:    mov     qword ptr [rsp + 0x28], rax
0x199d0:    mov     rax, qword ptr [rsp + 0x20]
0x199d5:    sub     eax, 1
0x199d8:    mov     qword ptr [rsp + 0x30], rax
0x199dd:    mov     rax, qword ptr [rsp + 8]
0x199e2:    mov     dword ptr [rsp + 0x38], r12d
0x199e7:    lea     rbp, [r14 + r12]
0x199eb:    lea     rbx, [rax + 8]
0x199ef:    mov     rax, qword ptr [rsp + 0x28]
0x199f4:    mov     r13, rax
0x199f7:    test    rax, rax
0x199fa:    jne     0x19a18
0x199fc:    jmp     0x19bb0
0x19bb0:
0x19a01:    nop     dword ptr [rax]
0x19a08:    mov     r13, qword ptr [rbx]
0x19a0b:    add     rbx, 8
0x19a0f:    test    r13, r13
0x19a12:    je      0x19bb0
0x19a18:    mov     rdi, r13
0x19a1b:    call    0x49b0
0x19a20:    mov     rsi, r13
0x19a23:    mov     rdi, rbp
0x19a26:    mov     rdx, rax
0x19a29:    mov     r15, rax
0x19a2c:    call    0x4890
0x19a31:    test    eax, eax
0x19a33:    jne     0x19a08
0x19a35:    mov     eax, dword ptr [rsp + 0x38]
0x19a39:    mov     ebx, dword ptr [rsp + 0x3c]
0x19a3d:    lea     eax, [rax + r15 - 1]
0x19a42:    lea     ecx, [rbx + rax]
0x19a45:    mov     dword ptr [rsp + 0x18], eax
0x19a49:    movsxd  r13, eax
0x19a4c:    mov     dword ptr [rsp + 8], ecx
0x19a50:    lea     r15, [r14 + r13]
0x19a54:    mov     ecx, dword ptr [rsp + 8]
0x19a58:    cmp     dword ptr [rsp + 0x1c], ecx
0x19a5c:    jle     0x19bf0
0x19a62:    movsxd  rbx, ebx
0x19a65:    add     rbx, qword ptr [rsp + 0x10]
0x19a6a:    mov     rdx, r13

```

```

0x19a6d:    mov     rsi, r14
0x19a70:    mov     rdi, rbx
0x19a73:    mov     qword ptr [rsp + 0x28], rbx
0x19a78:    call    0x4880
0x19a7d:    mov     rdi, r15
0x19a80:    mov     r13, qword ptr [rip + 0x10d19]
0x19a87:    call    0x49b0
0x19a8c:    mov     dword ptr [rsp + 0x20], eax
0x19a90:    test    eax, eax
0x19a92:    jle     0x19c64
0x19a98:    mov     rbx, r15
0x19a9b:    xor     ebp, ebp
0x19a9d:    movzx   eax, byte ptr [rbx]
0x19aa0:    cmp     al, 0x2f
0x19aa2:    je      0x19b80
0x19aa8:    test    al, al
0x19aaa:    je      0x19b80
0x19ab0:    test    r13, r13
0x19ab3:    je      0x19aea
0x19ab5:    mov     r14, r13
0x19ab8:    jmp     0x19ae2
0x19ae2:
0x19aba:    nop     word ptr [rax + rax]
0x19ac0:    mov     rdi, r12
0x19ac3:    add     r14, 8
0x19ac7:    call    0x49b0
0x19acc:    mov     rsi, r12
0x19acf:    mov     rdi, rbx
0x19ad2:    mov     rdx, rax
0x19ad5:    call    0x4890
0x19ada:    test    eax, eax
0x19adc:    je      0x19b80
0x19ae2:    mov     r12, qword ptr [r14]
0x19ae5:    test    r12, r12
0x19ae8:    jne     0x19ac0
0x19aea:    add     ebp, 1
0x19aed:    add     rbx, 1
0x19af1:    cmp     dword ptr [rsp + 0x20], ebp
0x19af5:    jne     0x19a9d
0x19af7:    nop     word ptr [rax + rax]
0x19b00:    lea     edi, [rbp + 1]
0x19b03:    movsxd  rbp, ebp
0x19b06:    movsxd  rdi, edi
0x19b09:    lea     r14, [r15 + rbp]
0x19b0d:    call    0x1d670
0x19b12:    mov     rdx, rbp
0x19b15:    mov     rsi, r15
0x19b18:    mov     r13, rax

```

```

0x19b1b:    mov     rdi, rax
0x19b1e:    call   0x4880
0x19b23:    mov     byte ptr [r13 + rbp], 0
0x19b29:    mov     rdi, r13
0x19b2c:    xor     eax, eax
0x19b2e:    call   0x196e0
0x19b33:    mov     rdi, r13
0x19b36:    mov     rbp, rax
0x19b39:    call   0x18d10
0x19b3e:    mov     rdi, rbp
0x19b41:    call   0x49b0
0x19b46:    mov     ecx, dword ptr [rsp + 8]
0x19b4a:    lea     ebx, [rcx + rax]
0x19b4d:    cmp     ebx, dword ptr [rsp + 0x1c]
0x19b51:    jge     0x19c20
0x19b57:    movsxd  rdi, dword ptr [rsp + 8]
0x19b5c:    mov     rsi, rbp
0x19b5f:    add     rdi, qword ptr [rsp + 0x10]
0x19b64:    call   0x48b0
0x19b69:    mov     rdi, rbp
0x19b6c:    call   0x18d10
0x19b71:    jmp     0x19970
0x19970:
0x19b76:    nop     word ptr cs:[rax + rax]
0x19b80:    or      dword ptr [rsp + 0x18], ebp
0x19b84:    mov     eax, dword ptr [rsp + 0x18]
0x19b88:    test    eax, eax
0x19b8a:    jne     0x19b00
0x19b90:    mov     rax, qword ptr [rsp + 0x28]
0x19b95:    mov     byte ptr [rax], 0
0x19b98:    mov     rax, qword ptr [rsp + 0x10]
0x19b9d:    add     rsp, 0x48
0x19ba1:    pop     rbx
0x19ba2:    pop     rbp
0x19ba3:    pop     r12
0x19ba5:    pop     r13
0x19ba7:    pop     r14
0x19ba9:    pop     r15
0x19bab:    ret
0x19bac:    nop     dword ptr [rax]
0x19bb0:    lea     rax, [r12 + 1]
0x19bb5:    cmp     qword ptr [rsp + 0x30], r12
0x19bba:    je      0x19c46
0x19bc0:    mov     r12, rax
0x19bc3:    jmp     0x199dd
0x199dd:
0x19bc8:    nop     dword ptr [rax + rax]
0x19bd0:    mov     dword ptr [rsp + 8], ebx

```

```

0x19bd4:    mov     r15, r14
0x19bd7:    mov     ecx, dword ptr [rsp + 8]
0x19bdb:    xor     r13d, r13d
0x19bde:    mov     dword ptr [rsp + 0x18], 0
0x19be6:    cmp     dword ptr [rsp + 0x1c], ecx
0x19bea:    jg      0x19a62
0x19bf0:    mov     eax, dword ptr [rsp + 0x1c]
0x19bf4:    mov     ecx, dword ptr [rsp + 0x18]
0x19bf8:    mov     rdi, qword ptr [rsp + 0x10]
0x19bfd:    lea     eax, [rax + rcx + 0x14]
0x19c01:    lea     esi, [rax + 1]
0x19c04:    mov     dword ptr [rsp + 0x1c], eax
0x19c08:    movsxd  rsi, esi
0x19c0b:    call    0x1d690
0x19c10:    mov     qword ptr [rsp + 0x10], rax
0x19c15:    jmp     0x19a62
0x19a62:
0x19c1a:    nop     word ptr [rax + rax]
0x19c20:    mov     ecx, dword ptr [rsp + 0x1c]
0x19c24:    mov     rdi, qword ptr [rsp + 0x10]
0x19c29:    lea     eax, [rcx + rax + 0x14]
0x19c2d:    lea     esi, [rax + 1]
0x19c30:    mov     dword ptr [rsp + 0x1c], eax
0x19c34:    movsxd  rsi, esi
0x19c37:    call    0x1d690
0x19c3c:    mov     qword ptr [rsp + 0x10], rax
0x19c41:    jmp     0x19b57
0x19b57:
0x19c46:    mov     ebx, dword ptr [rsp + 0x3c]
0x19c4a:    mov     rax, qword ptr [rsp + 0x20]
0x19c4f:    nop
0x19c50:    mov     ecx, eax
0x19c52:    movsxd  r13, eax
0x19c55:    add     ecx, ebx
0x19c57:    lea     r15, [r14 + r13]
0x19c5b:    mov     dword ptr [rsp + 8], ecx
0x19c5f:    jmp     0x19a54
0x19a54:
0x19c64:    xor     ebp, ebp
0x19c66:    jmp     0x19b84
0x19b84:
0x19c6b:    nop     dword ptr [rax + rax]
0x19c70:    push    r15
0x19c72:    mov     r15, rsi
0x19c75:    push    r14
0x19c77:    mov     r14d, edx
0x19c7a:    push    r13
0x19c7c:    push    r12

```

```

0x19c7e:      mov     r12, rdi
0x19c81:      push    rbp
0x19c82:      push    rbx
0x19c83:      sub     rsp, 0x18
0x19c87:      movsxd  rsi, dword ptr [rsi]
0x19c8a:      lea     r13, [rdi + rsi]
0x19c8e:      mov     edi, ecx
0x19c90:      mov     ebx, esi
0x19c92:      mov     rbp, rsi
0x19c95:      movzx   eax, byte ptr [r13]
0x19c9a:      and     edi, 1
0x19c9d:      mov     rdx, rsi
0x19ca0:      test    al, al
0x19ca2:      jne     0x19cdc
0x19ca4:      jmp     0x19db9
0x19db9:
0x19ca9:      nop     dword ptr [rax]
0x19cb0:      cmp     al, 0x5c
0x19cb2:      jne     0x19cc5
0x19cb4:      movsx   eax, byte ptr [r12 + rdx + 1]
0x19cba:      cmp     eax, r14d
0x19cbd:      sete    al
0x19cc0:      movzx   eax, al
0x19cc3:      add     ebx, eax
0x19cc5:      add     ebx, 1
0x19cc8:      movsxd  rdx, ebx
0x19ccb:      lea     r13, [r12 + rdx]
0x19ccf:      movzx   eax, byte ptr [r13]
0x19cd4:      test    al, al
0x19cd6:      je      0x19da0
0x19cdc:      movsx   ecx, al
0x19cdf:      cmp     ecx, r14d
0x19ce2:      jne     0x19cb0
0x19ce4:      cmp     ebp, ebx
0x19ce6:      jl      0x19d10
0x19ce8:      test    dil, dil
0x19ceb:      jne     0x19d10
0x19ced:      movzx   eax, byte ptr [r13]
0x19cf2:      xor     r9d, r9d
0x19cf5:      cmp     al, 1
0x19cf7:      sbb     ebx, -1
0x19cfa:      mov     dword ptr [r15], ebx
0x19cfd:      add     rsp, 0x18
0x19d01:      mov     rax, r9
0x19d04:      pop     rbx
0x19d05:      pop     rbp
0x19d06:      pop     r12
0x19d08:      pop     r13

```

```

0x19d0a:    pop     r14
0x19d0c:    pop     r15
0x19d0e:    ret
0x19d0f:    nop
0x19d10:    mov     edi, ebx
0x19d12:    mov     qword ptr [rsp], r8
0x19d16:    sub     edi, ebp
0x19d18:    mov     qword ptr [rsp + 8], rsi
0x19d1d:    add     edi, 1
0x19d20:    movsxd  rdi, edi
0x19d23:    call    0x1d670
0x19d28:    cmp     ebp, ebx
0x19d2a:    mov     r8, qword ptr [rsp]
0x19d2e:    mov     r9, rax
0x19d31:    jge     0x19dd6
0x19d37:    mov     rsi, qword ptr [rsp + 8]
0x19d3c:    mov     eax, 1
0x19d41:    jmp     0x19d5d
0x19d5d
0x19d43:    nop     dword ptr [rax + rax]
0x19d48:    add     ebp, 1
0x19d4b:    mov     byte ptr [r9 + rax - 1], dl
0x19d50:    mov     edx, eax
0x19d52:    add     rax, 1
0x19d56:    cmp     ebp, ebx
0x19d58:    jge     0x19d80
0x19d5a:    movsxd  rsi, ebp
0x19d5d:    movzx   edx, byte ptr [r12 + rsi]
0x19d62:    cmp     dl, 0x5c
0x19d65:    jne     0x19d48
0x19d67:    movsx   esi, byte ptr [r12 + rsi + 1]
0x19d6d:    cmp     esi, r14d
0x19d70:    jne     0x19d48
0x19d72:    add     ebp, 1
0x19d75:    mov     edx, esi
0x19d77:    jmp     0x19d48
0x19d48
0x19d79:    nop     dword ptr [rax]
0x19d80:    movsxd  rax, edx
0x19d83:    add     rax, r9
0x19d86:    mov     byte ptr [rax], 0
0x19d89:    mov     dword ptr [r8], edx
0x19d8c:    movzx   eax, byte ptr [r13]
0x19d91:    jmp     0x19cf5
0x19cf5
0x19d96:    nop     word ptr cs:[rax + rax]
0x19da0:    cmp     ebp, ebx
0x19da2:    jl      0x19d10

```

```

0x19da8:      test    dil, dil
0x19dab:      jne     0x19d10
0x19db1:      xor     r9d, r9d
0x19db4:      jmp     0x19cfa
0x19cfa
0x19db9:      test    ecx, ecx
0x19dbb:      je      0x19db1
0x19dbd:      mov     edi, 1
0x19dc2:      mov     qword ptr [rsp], r8
0x19dc6:      call    0x1d670
0x19dcb:      mov     r8, qword ptr [rsp]
0x19dcf:      xor     edx, edx
0x19dd1:      mov     r9, rax
0x19dd4:      jmp     0x19d86
0x19d86
0x19dd6:      xor     edx, edx
0x19dd8:      jmp     0x19d86
0x19d86
0x19dda:      nop     word ptr [rax + rax]
0x19de0:      push    r15
0x19de2:      push    r14
0x19de4:      push    r13
0x19de6:      push    r12
0x19de8:      push    rbp
0x19de9:      push    rbx
0x19dea:      sub     rsp, 0x28
0x19dee:      mov     eax, dword ptr [rip + 0x11cf0]
0x19df4:      mov     ebx, dword ptr [rip + 0x11d02]
0x19dfa:      mov     dword ptr [rsp + 8], esi
0x19dfe:      mov     dword ptr [rsp + 0x18], edx
0x19e02:      mov     dword ptr [rsp + 0x1c], eax
0x19e06:      test    eax, eax
0x19e08:      je      0x19f9d
0x19e0e:      mov     rbp, rdi
0x19e11:      cmp     eax, ebx
0x19e13:      je      0x19fb0
0x19e19:      call    0x49b0
0x19e1e:      mov     ecx, dword ptr [rsp + 8]
0x19e22:      mov     dword ptr [rsp + 0xc], eax
0x19e26:      test    ecx, ecx
0x19e28:      js      0x19fbb
0x19e2e:      mov     rax, qword ptr [rip + 0x11ccb]
0x19e35:      movsxd  r14, dword ptr [rsp + 0xc]
0x19e3a:      mov     qword ptr [rsp + 0x10], rax
0x19e3f:      mov     edx, dword ptr [rsp + 8]
0x19e43:      test    edx, edx
0x19e45:      js      0x19f16
0x19e4b:      cmp     dword ptr [rsp + 0x1c], ebx

```

```

0x19e4f:      je      0x19f9d
0x19e55:      mov     rcx, qword ptr [rsp + 0x10]
0x19e5a:      movsxd  rax, ebx
0x19e5d:      mov     rax, qword ptr [rcx + rax*8]
0x19e61:      mov     r13, qword ptr [rax]
0x19e64:      mov     rdi, r13
0x19e67:      call    0x49b0
0x19e6c:      mov     r10, rax
0x19e6f:      cmp     dword ptr [rsp + 0xc], eax
0x19e73:      jg      0x19f07
0x19e79:      cmp     dword ptr [rsp + 0x18], 1
0x19e7e:      movzx   r12d, byte ptr [rbp]
0x19e83:      je      0x19ee0
0x19e85:      mov     r15d, dword ptr [rsp + 0x18]
0x19e8a:      sub     r10d, dword ptr [rsp + 0xc]
0x19e8f:      jmp     0x19ea5
0x19ea5:
0x19e91:      nop     dword ptr [rax]
0x19e98:      add     r15d, 1
0x19e9c:      add     r13, 1
0x19ea0:      cmp     r15d, r10d
0x19ea3:      jg      0x19f07
0x19ea5:      cmp     byte ptr [r13], r12b
0x19ea9:      jne     0x19e98
0x19eab:      mov     rdx, r14
0x19eae:      mov     rsi, r13
0x19eb1:      mov     rdi, rbp
0x19eb4:      mov     dword ptr [rsp + 4], r10d
0x19eb9:      call    0x4890
0x19ebe:      mov     r10d, dword ptr [rsp + 4]
0x19ec3:      test    eax, eax
0x19ec5:      jne     0x19e98
0x19ec7:      mov     dword ptr [rip + 0x11c2f], ebx
0x19ecd:      add     rsp, 0x28
0x19ed1:      mov     eax, r15d
0x19ed4:      pop     rbx
0x19ed5:      pop     rbp
0x19ed6:      pop     r12
0x19ed8:      pop     r13
0x19eda:      pop     r14
0x19edc:      pop     r15
0x19ede:      ret
0x19edf:      nop
0x19ee0:      cmp     byte ptr [r13], r12b
0x19ee4:      jne     0x19efb
0x19ee6:      mov     rdx, r14
0x19ee9:      mov     rsi, r13
0x19eec:      mov     rdi, rbp

```



```

0x19eef:      call    0x4890
0x19ef4:      mov     r15d, eax
0x19ef7:      test    eax, eax
0x19ef9:      je       0x19ec7
0x19efb:      mov     eax, dword ptr [rsp + 8]
0x19eff:      test    eax, eax
0x19f01:      js       0x19f90
0x19f07:      mov     edx, dword ptr [rsp + 8]
0x19f0b:      add     ebx, 1
0x19f0e:      test    edx, edx
0x19f10:      jns      0x19e4b
0x19f16:      mov     rcx, qword ptr [rsp + 0x10]
0x19f1b:      movsxd   rax, ebx
0x19f1e:      mov     rax, qword ptr [rcx + rax*8]
0x19f22:      mov     r13, qword ptr [rax]
0x19f25:      mov     rdi, r13
0x19f28:      call    0x49b0
0x19f2d:      mov     r8, rax
0x19f30:      cmp     dword ptr [rsp + 0xc], eax
0x19f34:      jg       0x19f90
0x19f36:      cmp     dword ptr [rsp + 0x18], 1
0x19f3b:      movzx   r12d, byte ptr [rbp]
0x19f40:      je       0x19ee0
0x19f42:      mov     r15d, r8d
0x19f45:      sub     r15d, dword ptr [rsp + 0xc]
0x19f4a:      movsxd   rcx, r15d
0x19f4d:      add     r13, rcx
0x19f50:      jmp     0x19f66
0x19f66:      nop     word ptr [rax + rax]
0x19f52:      sub     r15d, 1
0x19f58:      sub     r13, 1
0x19f5c:      sub     r13, 1
0x19f60:      cmp     r15d, -1
0x19f64:      je       0x19f90
0x19f66:      cmp     byte ptr [r13], r12b
0x19f6a:      jne     0x19f58
0x19f6c:      mov     rdx, r14
0x19f6f:      mov     rsi, r13
0x19f72:      mov     rdi, rbp
0x19f75:      call    0x4890
0x19f7a:      test    eax, eax
0x19f7c:      je       0x19ec7
0x19f82:      sub     r15d, 1
0x19f86:      sub     r13, 1
0x19f8a:      cmp     r15d, -1
0x19f8e:      jne     0x19f66
0x19f90:      sub     ebx, 1
0x19f93:      test    dword ptr [rsp + 8], ebx

```

```

0x19f97:      jns      0x19e3f
0x19f9d:      mov     r15d, 0xffffffff
0x19fa3:      jmp     0x19ecd
0x19ecd
0x19fa8:      nop     dword ptr [rax + rax]
0x19fb0:      test    esi, esi
0x19fb2:      jns     0x19f9d
0x19fb4:      mov     eax, dword ptr [rsp + 0x1c]
0x19fb8:      lea     ebx, [rax - 1]
0x19fbb:      mov     rdi, rbp
0x19fbe:      call    0x49b0
0x19fc3:      mov     dword ptr [rsp + 0xc], eax
0x19fc7:      test    dword ptr [rsp + 8], ebx
0x19fcb:      jns     0x19e2e
0x19fd1:      jmp     0x19f9d
0x19f9d
0x19fd3:      nop     word ptr cs:[rax + rax]
0x19fde:      nop
0x19fe0:      push    r14
0x19fe2:      push    r13
0x19fe4:      movsxd  r13, esi
0x19fe7:      sub     edx, r13d
0x19fea:      push    r12
0x19fec:      push    rbp
0x19fed:      mov     rbp, rdi
0x19ff0:      push    rbx
0x19ff1:      movsxd  rbx, edx
0x19ff4:      cmp     ecx, 2
0x19ff7:      je      0x1a070
0x19ff9:      mov     eax, 0x1d
0x19ffe:      lea     r14, [rip + 0x843a]
0x1a005:      cmp     ecx, 3
0x1a008:      je      0x1a028
0x1a00a:      xor     eax, eax
0x1a00c:      lea     r14, [rip + 0x83f5]
0x1a013:      lea     rdx, [rip + 0x83fe]
0x1a01a:      cmp     ecx, 1
0x1a01d:      sete    al
0x1a020:      cmov     r14, rdx
0x1a024:      lea     eax, [rax + rax*2 + 0xf]
0x1a028:      lea     edi, [rax + rbx + 3]
0x1a02c:      movsxd  rdi, edi
0x1a02f:      call    0x1d670
0x1a034:      lea     rsi, [rbp + r13]
0x1a039:      mov     rdx, rbx
0x1a03c:      mov     r12, rax
0x1a03f:      mov     rdi, rax
0x1a042:      call    0x4880

```

```

0x1a047:    mov     byte ptr [r12 + rbx], 0x3a
0x1a04c:    lea     rdi, [r12 + rbx + 2]
0x1a051:    mov     rsi, r14
0x1a054:    mov     byte ptr [r12 + rbx + 1], 0x20
0x1a05a:    call    0x48b0
0x1a05f:    pop     rbx
0x1a060:    mov     rax, r12
0x1a063:    pop     rbp
0x1a064:    pop     r12
0x1a066:    pop     r13
0x1a068:    pop     r14
0x1a06a:    ret
0x1a06b:    nop     dword ptr [rax + rax]
0x1a070:    mov     eax, 0x13
0x1a075:    lea     r14, [rip + 0x83af]
0x1a07c:    jmp     0x1a028
0x1a028:
0x1a07e:    nop
0x1a080:    push    r12
0x1a082:    push    rbp
0x1a083:    push    rbx
0x1a084:    test    rdi, rdi
0x1a087:    je      0x1a0b0
0x1a089:    mov     rbp, rdi
0x1a08c:    call    0x49b0
0x1a091:    lea     rdi, [rax + 1]
0x1a095:    call    0x1d670
0x1a09a:    pop     rbx
0x1a09b:    mov     rsi, rbp
0x1a09e:    pop     rbp
0x1a09f:    mov     rdi, rax
0x1a0a2:    pop     r12
0x1a0a4:    jmp     0x48b0
0x48b0:
0x1a0a9:    nop     dword ptr [rax]
0x1a0b0:    lea     rdi, [rip + 0x64a9]
0x1a0b7:    call    0x47e0
0x1a0bc:    mov     r12, rax
0x1a0bf:    test    rax, rax
0x1a0c2:    je      0x1a110
0x1a0c4:    mov     rdi, rax
0x1a0c7:    call    0x49b0
0x1a0cc:    lea     edi, [rax + 0xa]
0x1a0cf:    movsxd  rbp, eax
0x1a0d2:    movsxd  rdi, edi
0x1a0d5:    lea     rbx, [rbp + 1]
0x1a0d9:    call    0x1d670
0x1a0de:    mov     rsi, r12

```

```

0x1a0e1:      mov     rdi, rax
0x1a0e4:      call    0x48b0
0x1a0e9:      movabs   rdx, 0x79726f747369682e
0x1a0f3:      mov     byte ptr [rax + rbp], 0x2f
0x1a0f7:      mov     r8, rax
0x1a0fa:      lea     rax, [rax + rbx]
0x1a0fe:      mov     qword ptr [rax], rdx
0x1a101:      mov     byte ptr [rax + 8], 0
0x1a105:      mov     rax, r8
0x1a108:      pop     rbx
0x1a109:      pop     rbp
0x1a10a:      pop     r12
0x1a10c:      ret
0x1a10d:      nop     dword ptr [rax]
0x1a110:      mov     ebx, 2
0x1a115:      mov     ebp, 1
0x1a11a:      mov     edi, 0xb
0x1a11f:      lea     r12, [rip + 0x75b8]
0x1a126:      jmp     0x1a0d9
0x1a0d9:
0x1a128:      nop     dword ptr [rax + rax]
0x1a130:      push    r15
0x1a132:      xor     eax, eax
0x1a134:      push    r14
0x1a136:      mov     r14d, esi
0x1a139:      push    r13
0x1a13b:      push    r12
0x1a13d:      push    rbp
0x1a13e:      push    rbx
0x1a13f:      mov     ebx, edx
0x1a141:      sub     rsp, 0x18
0x1a145:      call    0x1a080
0x1a14a:      cmp     ebx, 1
0x1a14d:      mov     edx, 0x1b6
0x1a152:      sbb     esi, esi
0x1a154:      mov     r13, rax
0x1a157:      mov     rdi, rax
0x1a15a:      xor     eax, eax
0x1a15c:      and     esi, 0x1c0
0x1a162:      add     esi, 0x241
0x1a168:      call    0x4d80
0x1a16d:      cmp     eax, -1
0x1a170:      je      0x1a270
0x1a176:      mov     ebp, eax
0x1a178:      mov     eax, dword ptr [rip + 0x11966]
0x1a17e:      cmp     eax, r14d
0x1a181:      mov     edx, eax
0x1a183:      cmovle  r14d, eax

```

```

0x1a187:    sub     edx, r14d
0x1a18a:    cmp     eax, edx
0x1a18c:    jle     0x1a290
0x1a192:    sub     eax, 1
0x1a195:    mov     rcx, qword ptr [rip + 0x11964]
0x1a19c:    movsxd  rsi, edx
0x1a19f:    xor     r12d, r12d
0x1a1a2:    sub     eax, edx
0x1a1a4:    add     rax, rsi
0x1a1a7:    lea     rbx, [rcx + rsi*8]
0x1a1ab:    lea     r15, [rcx + rax*8 + 8]
0x1a1b0:    mov     rax, qword ptr [rbx]
0x1a1b3:    add     rbx, 8
0x1a1b7:    mov     rdi, qword ptr [rax]
0x1a1ba:    call    0x49b0
0x1a1bf:    lea     r12d, [r12 + rax + 1]
0x1a1c4:    cmp     r15, rbx
0x1a1c7:    jne     0x1a1b0
0x1a1c9:    movsxd  rax, r12d
0x1a1cc:    mov     qword ptr [rsp + 8], rax
0x1a1d1:    mov     rdi, qword ptr [rsp + 8]
0x1a1d6:    call    0x1d670
0x1a1db:    mov     r15, rax
0x1a1de:    mov     eax, dword ptr [rip + 0x11900]
0x1a1e4:    mov     ebx, eax
0x1a1e6:    sub     ebx, r14d
0x1a1e9:    cmp     eax, ebx
0x1a1eb:    jle     0x1a238
0x1a1ed:    mov     r12, qword ptr [rip + 0x1190c]
0x1a1f4:    movsxd  rbx, ebx
0x1a1f7:    xor     r14d, r14d
0x1a1fa:    nop     word ptr [rax + rax]
0x1a200:    mov     rax, qword ptr [r12 + rbx*8]
0x1a204:    movsxd  rdi, r14d
0x1a207:    add     rdi, r15
0x1a20a:    mov     rsi, qword ptr [rax]
0x1a20d:    call    0x48b0
0x1a212:    mov     rax, qword ptr [r12 + rbx*8]
0x1a216:    add     rbx, 1
0x1a21a:    mov     rdi, qword ptr [rax]
0x1a21d:    call    0x49b0
0x1a222:    add     eax, r14d
0x1a225:    lea     r14d, [rax + 1]
0x1a229:    cdqe
0x1a22b:    mov     byte ptr [r15 + rax], 0xa
0x1a230:    cmp     dword ptr [rip + 0x118ae], ebx
0x1a236:    jg      0x1a200
0x1a238:    mov     rdx, qword ptr [rsp + 8]

```

```

0x1a23d:    mov     rsi, r15
0x1a240:    mov     edi, ebp
0x1a242:    call    0x4790
0x1a247:    mov     rdi, r15
0x1a24a:    call    0x18d10
0x1a24f:    mov     edi, ebp
0x1a251:    call    0x4ab0
0x1a256:    mov     rdi, r13
0x1a259:    call    0x18d10
0x1a25e:    add     rsp, 0x18
0x1a262:    xor     eax, eax
0x1a264:    pop     rbx
0x1a265:    pop     rbp
0x1a266:    pop     r12
0x1a268:    pop     r13
0x1a26a:    pop     r14
0x1a26c:    pop     r15
0x1a26e:    ret
0x1a26f:    nop
0x1a270:    mov     rdi, r13
0x1a273:    call    0x18d10
0x1a278:    call    0x4860
0x1a27d:    mov     eax, dword ptr [rax]
0x1a27f:    add     rsp, 0x18
0x1a283:    pop     rbx
0x1a284:    pop     rbp
0x1a285:    pop     r12
0x1a287:    pop     r13
0x1a289:    pop     r14
0x1a28b:    pop     r15
0x1a28d:    ret
0x1a28e:    nop
0x1a290:    mov     qword ptr [rsp + 8], 0
0x1a299:    jmp     0x1a1d1
0x1a1d1:
0x1a29e:    nop
0x1a2a0:    endbr64
0x1a2a4:    sub     rsp, 8
0x1a2a8:    mov     edi, 0x18
0x1a2ad:    call    0x1d670
0x1a2b2:    mov     rdx, qword ptr [rip + 0x11847]
0x1a2b9:    mov     qword ptr [rax], rdx
0x1a2bc:    mov     edx, dword ptr [rip + 0x1183a]
0x1a2c2:    mov     dword ptr [rax + 8], edx
0x1a2c5:    mov     edx, dword ptr [rip + 0x11819]
0x1a2cb:    mov     dword ptr [rax + 0xc], edx
0x1a2ce:    mov     edx, dword ptr [rip + 0x11824]
0x1a2d4:    mov     dword ptr [rax + 0x10], edx

```

```

0x1a2d7:    add     rsp, 8
0x1a2db:    ret
0x1a2dc:    nop     dword ptr [rax]
0x1a2e0:    endbr64
0x1a2e4:    mov     rax, qword ptr [rdi]
0x1a2e7:    mov     qword ptr [rip + 0x11812], rax
0x1a2ee:    mov     eax, dword ptr [rdi + 8]
0x1a2f1:    mov     dword ptr [rip + 0x11805], eax
0x1a2f7:    mov     eax, dword ptr [rdi + 0xc]
0x1a2fa:    mov     dword ptr [rip + 0x117e4], eax
0x1a300:    mov     eax, dword ptr [rdi + 0x10]
0x1a303:    mov     dword ptr [rip + 0x117ef], eax
0x1a309:    ret
0x1a30a:    nop     word ptr [rax + rax]
0x1a310:    endbr64
0x1a314:    mov     eax, dword ptr [rip + 0x117ca]
0x1a31a:    mov     dword ptr [rip + 0x117dc], eax
0x1a320:    ret
0x1a321:    nop     word ptr cs:[rax + rax]
0x1a32c:    nop     dword ptr [rax]
0x1a330:    endbr64
0x1a334:    push    r12
0x1a336:    xor     r12d, r12d
0x1a339:    push    rbx
0x1a33a:    sub     rsp, 8
0x1a33e:    mov     rbx, qword ptr [rip + 0x117bb]
0x1a345:    test    rbx, rbx
0x1a348:    jne     0x1a35b
0x1a34a:    jmp     0x1a367
0x1a367:
0x1a34c:    nop     dword ptr [rax]
0x1a350:    mov     rdi, qword ptr [rax]
0x1a353:    call    0x49b0
0x1a358:    add     r12d, eax
0x1a35b:    mov     rax, qword ptr [rbx]
0x1a35e:    add     rbx, 8
0x1a362:    test    rax, rax
0x1a365:    jne     0x1a350
0x1a367:    add     rsp, 8
0x1a36b:    mov     eax, r12d
0x1a36e:    pop     rbx
0x1a36f:    pop     r12
0x1a371:    ret
0x1a372:    nop     word ptr cs:[rax + rax]
0x1a37d:    nop     dword ptr [rax]
0x1a380:    endbr64
0x1a384:    push    rbp
0x1a385:    mov     rbp, rdi

```

```

0x1a388:    push    rbx
0x1a389:    sub     rsp, 8
0x1a38d:    mov     eax, dword ptr [rip + 0x11759]
0x1a393:    test    eax, eax
0x1a395:    je      0x1a3aa
0x1a397:    movsxd  rdx, dword ptr [rip + 0x11746]
0x1a39e:    cmp     edx, dword ptr [rip + 0x11744]
0x1a3a4:    je      0x1a458
0x1a3aa:    mov     eax, dword ptr [rip + 0x11748]
0x1a3b0:    test    eax, eax
0x1a3b2:    je      0x1a428
0x1a3b4:    mov     edx, dword ptr [rip + 0x1172a]
0x1a3ba:    lea     ecx, [rax - 1]
0x1a3bd:    cmp     ecx, edx
0x1a3bf:    je      0x1a4b0
0x1a3c5:    add     edx, 1
0x1a3c8:    mov     dword ptr [rip + 0x11716], edx
0x1a3ce:    mov     edi, 0x10
0x1a3d3:    call    0x1d670
0x1a3d8:    mov     rdi, rbp
0x1a3db:    mov     rbx, rax
0x1a3de:    call    0x49b0
0x1a3e3:    lea     rdi, [rax + 1]
0x1a3e7:    call    0x1d670
0x1a3ec:    mov     rsi, rbp
0x1a3ef:    mov     rdi, rax
0x1a3f2:    call    0x48b0
0x1a3f7:    movsxd  rdx, dword ptr [rip + 0x116e6]
0x1a3fe:    mov     qword ptr [rbx + 8], 0
0x1a406:    mov     qword ptr [rbx], rax
0x1a409:    mov     rax, qword ptr [rip + 0x116f0]
0x1a410:    mov     qword ptr [rax + rdx*8], 0
0x1a418:    mov     qword ptr [rax + rdx*8 - 8], rbx
0x1a41d:    add     rsp, 8
0x1a421:    pop     rbx
0x1a422:    pop     rbp
0x1a423:    ret
0x1a424:    nop     dword ptr [rax]
0x1a428:    mov     dword ptr [rip + 0x116c6], 0x32
0x1a432:    mov     edi, 0x190
0x1a437:    call    0x1d670
0x1a43c:    mov     dword ptr [rip + 0x1169e], 1
0x1a446:    mov     qword ptr [rip + 0x116b3], rax
0x1a44d:    jmp     0x1a3ce
0x1a3ce:
0x1a452:    nop     word ptr [rax + rax]
0x1a458:    test    edx, edx
0x1a45a:    je      0x1a41d

```



```

0x1a45c:    mov     rax, qword ptr [rip + 0x1169d]
0x1a463:    mov     rax, qword ptr [rax]
0x1a466:    test    rax, rax
0x1a469:    je      0x1a489
0x1a46b:    mov     rdi, qword ptr [rax]
0x1a46e:    call    0x18d10
0x1a473:    mov     rax, qword ptr [rip + 0x11686]
0x1a47a:    mov     rdi, qword ptr [rax]
0x1a47d:    call    0x18d10
0x1a482:    movsxd  rdx, dword ptr [rip + 0x1165b]
0x1a489:    test    edx, edx
0x1a48b:    jle     0x1a4a1
0x1a48d:    mov     rdi, qword ptr [rip + 0x1166c]
0x1a494:    shl     rdx, 3
0x1a498:    lea     rsi, [rdi + 8]
0x1a49c:    call    0x4d40
0x1a4a1:    add     dword ptr [rip + 0x10340], 1
0x1a4a8:    jmp     0x1a3ce
0x1a3ce:
0x1a4ad:    nop     dword ptr [rax]
0x1a4b0:    add     eax, 0x32
0x1a4b3:    mov     rdi, qword ptr [rip + 0x11646]
0x1a4ba:    mov     dword ptr [rip + 0x11638], eax
0x1a4c0:    cdqe
0x1a4c2:    lea     rsi, [rax*8]
0x1a4ca:    call    0x1d690
0x1a4cf:    mov     edx, dword ptr [rip + 0x1160f]
0x1a4d5:    mov     qword ptr [rip + 0x11624], rax
0x1a4dc:    jmp     0x1a3c5
0x1a3c5:
0x1a4e1:    nop     word ptr cs:[rax + rax]
0x1a4ec:    nop     dword ptr [rax]
0x1a4f0:    endbr64
0x1a4f4:    push    r14
0x1a4f6:    xor     r14d, r14d
0x1a4f9:    push    r13
0x1a4fb:    mov     r13, rdx
0x1a4fe:    push    r12
0x1a500:    mov     r12, rsi
0x1a503:    push    rbp
0x1a504:    push    rbx
0x1a505:    movsxd  rbx, edi
0x1a508:    mov     edi, 0x10
0x1a50d:    call    0x1d670
0x1a512:    cmp     dword ptr [rip + 0x115cc], ebx
0x1a518:    jle     0x1a557
0x1a51a:    mov     rbp, rax
0x1a51d:    mov     rax, qword ptr [rip + 0x115dc]

```

```

0x1a524:      mov     rdi, r12
0x1a527:      mov     r14, qword ptr [rax + rbx*8]
0x1a52b:      call    0x49b0
0x1a530:      lea     rdi, [rax + 1]
0x1a534:      call    0x1d670
0x1a539:      mov     rsi, r12
0x1a53c:      mov     rdi, rax
0x1a53f:      call    0x48b0
0x1a544:      mov     qword ptr [rbp + 8], r13
0x1a548:      mov     qword ptr [rbp], rax
0x1a54c:      mov     rax, qword ptr [rip + 0x115ad]
0x1a553:      mov     qword ptr [rax + rbx*8], rbp
0x1a557:      pop     rbx
0x1a558:      mov     rax, r14
0x1a55b:      pop     rbp
0x1a55c:      pop     r12
0x1a55e:      pop     r13
0x1a560:      pop     r14
0x1a562:      ret
0x1a563:      nop     word ptr cs:[rax + rax]
0x1a56e:      nop
0x1a570:      endbr64
0x1a574:      mov     eax, dword ptr [rip + 0x11582]
0x1a57a:      ret
0x1a57b:      nop     dword ptr [rax + rax]
0x1a580:      endbr64
0x1a584:      xor     edx, edx
0x1a586:      xor     eax, eax
0x1a588:      jmp     0x19de0
0x19de0
0x1a58d:      nop     dword ptr [rax]
0x1a590:      endbr64
0x1a594:      mov     edx, 1
0x1a599:      xor     eax, eax
0x1a59b:      jmp     0x19de0
0x19de0
0x1a5a0:      endbr64
0x1a5a4:      push    r12
0x1a5a6:      push    rbx
0x1a5a7:      sub     rsp, 8
0x1a5ab:      mov     eax, dword ptr [rip + 0x11533]
0x1a5b1:      cmp     eax, edi
0x1a5b3:      jle     0x1a610
0x1a5b5:      test    eax, eax
0x1a5b7:      je      0x1a610
0x1a5b9:      lea     ebx, [rax - 1]
0x1a5bc:      movsxd  rdx, edi
0x1a5bf:      mov     rcx, qword ptr [rip + 0x1153a]

```

```

0x1a5c6:    mov     esi, ebx
0x1a5c8:    shl     rdx, 3
0x1a5cc:    sub     esi, edi
0x1a5ce:    cmp     eax, edi
0x1a5d0:    lea     r9, [rcx + rdx]
0x1a5d4:    mov     eax, 8
0x1a5d9:    lea     r8, [rsi*8 + 8]
0x1a5e1:    mov     r12, qword ptr [r9]
0x1a5e4:    mov     rdi, r9
0x1a5e7:    lea     rsi, [rcx + rdx + 8]
0x1a5ec:    cmovle  r8, rax
0x1a5f0:    mov     rdx, r8
0x1a5f3:    call    0x4d40
0x1a5f8:    mov     dword ptr [rip + 0x114e6], ebx
0x1a5fe:    add     rsp, 8
0x1a602:    mov     rax, r12
0x1a605:    pop     rbx
0x1a606:    pop     r12
0x1a608:    ret
0x1a609:    nop     dword ptr [rax]
0x1a610:    xor     r12d, r12d
0x1a613:    add     rsp, 8
0x1a617:    mov     rax, r12
0x1a61a:    pop     rbx
0x1a61b:    pop     r12
0x1a61d:    ret
0x1a61e:    nop
0x1a620:    endbr64
0x1a624:    push    r12
0x1a626:    mov     eax, dword ptr [rip + 0x114b8]
0x1a62c:    push    rbp
0x1a62d:    xor     ebp, ebp
0x1a62f:    test    edi, edi
0x1a631:    cmovns  ebp, edi
0x1a634:    push    rbx
0x1a635:    cmp     eax, ebp
0x1a637:    jle     0x1a6ce
0x1a63d:    sub     eax, ebp
0x1a63f:    mov     r12d, edi
0x1a642:    test    eax, eax
0x1a644:    jle     0x1a6e8
0x1a64a:    xor     ebx, ebx
0x1a64c:    nop     dword ptr [rax]
0x1a650:    mov     rax, qword ptr [rip + 0x114a9]
0x1a657:    mov     rax, qword ptr [rax + rbx*8]
0x1a65b:    mov     rdi, qword ptr [rax]
0x1a65e:    call    0x18d10
0x1a663:    mov     rax, qword ptr [rip + 0x11496]

```

```

0x1a66a:    mov     rdi, qword ptr [rax + rbx*8]
0x1a66e:    call   0x18d10
0x1a673:    mov     eax, dword ptr [rip + 0x1146b]
0x1a679:    lea     edx, [rbx + 1]
0x1a67c:    add     rbx, 1
0x1a680:    sub     eax, ebp
0x1a682:    cmp     eax, ebx
0x1a684:    jg      0x1a650
0x1a686:    mov     dword ptr [rip + 0x1015c], edx
0x1a68c:    mov     rcx, qword ptr [rip + 0x1146d]
0x1a693:    test    r12d, r12d
0x1a696:    jle     0x1a6f0
0x1a698:    cdqe
0x1a69a:    lea     rsi, [rcx + rax*8]
0x1a69e:    xor     eax, eax
0x1a6a0:    mov     rdx, qword ptr [rsi + rax*8]
0x1a6a4:    mov     qword ptr [rcx + rax*8], rdx
0x1a6a8:    add     rax, 1
0x1a6ac:    cmp     ebp, eax
0x1a6ae:    jg      0x1a6a0
0x1a6b0:    test    ebp, ebp
0x1a6b2:    mov     eax, 1
0x1a6b7:    cmovg   eax, ebp
0x1a6ba:    movsxd  rdx, eax
0x1a6bd:    lea     rcx, [rcx + rdx*8]
0x1a6c1:    mov     qword ptr [rcx], 0
0x1a6c8:    mov     dword ptr [rip + 0x11416], eax
0x1a6ce:    pop     rbx
0x1a6cf:    mov     dword ptr [rip + 0x11413], ebp
0x1a6d5:    mov     dword ptr [rip + 0x1140d], 1
0x1a6df:    pop     rbp
0x1a6e0:    pop     r12
0x1a6e2:    ret
0x1a6e3:    nop     dword ptr [rax + rax]
0x1a6e8:    xor     edx, edx
0x1a6ea:    jmp     0x1a686
0x1a686:
0x1a6ec:    nop     dword ptr [rax]
0x1a6f0:    xor     eax, eax
0x1a6f2:    jmp     0x1a6c1
0x1a6c1:
0x1a6f4:    nop     word ptr cs:[rax + rax]
0x1a6ff:    nop
0x1a700:    endbr64
0x1a704:    mov     edx, dword ptr [rip + 0x113e2]
0x1a70a:    mov     eax, dword ptr [rip + 0x113d8]
0x1a710:    test    edx, edx
0x1a712:    je      0x1a720

```

```

0x1a714:    mov     dword ptr [rip + 0x113ce], 0
0x1a71e:    neg     eax
0x1a720:    ret
0x1a721:    nop     word ptr cs:[rax + rax]
0x1a72c:    nop     dword ptr [rax]
0x1a730:    endbr64
0x1a734:    push    r15
0x1a736:    push    r14
0x1a738:    mov     r14d, esi
0x1a73b:    push    r13
0x1a73d:    push    r12
0x1a73f:    mov     r12d, edx
0x1a742:    push    rbp
0x1a743:    push    rbx
0x1a744:    sub     rsp, 0xa8
0x1a74b:    mov     rax, qword ptr fs:[0x28]
0x1a754:    mov     qword ptr [rsp + 0x98], rax
0x1a75c:    xor     eax, eax
0x1a75e:    call    0x1a080
0x1a763:    xor     esi, esi
0x1a765:    mov     edx, 0x1b6
0x1a76a:    mov     rbp, rax
0x1a76d:    mov     rdi, rax
0x1a770:    xor     eax, eax
0x1a772:    call    0x4d80
0x1a777:    test    eax, eax
0x1a779:    js      0x1a80f
0x1a77f:    mov     rsi, rsp
0x1a782:    mov     edi, eax
0x1a784:    mov     r15d, eax
0x1a787:    call    0x18ef0
0x1a78c:    cmp     eax, -1
0x1a78f:    je      0x1a807
0x1a791:    mov     eax, dword ptr [rsp + 0x30]
0x1a795:    lea     edi, [rax + 1]
0x1a798:    movsxd  rdi, edi
0x1a79b:    call    0x1d670
0x1a7a0:    mov     rdx, qword ptr [rsp + 0x30]
0x1a7a5:    mov     edi, r15d
0x1a7a8:    mov     rsi, rax
0x1a7ab:    mov     r13, rax
0x1a7ae:    call    0x47a0
0x1a7b3:    cmp     qword ptr [rsp + 0x30], rax
0x1a7b8:    je      0x1a819
0x1a7ba:    mov     edi, r15d
0x1a7bd:    call    0x4ab0
0x1a7c2:    mov     rdi, rbp
0x1a7c5:    call    0x18d10

```

```

0x1a7ca:    test    r13, r13
0x1a7cd:    je      0x1a7d7
0x1a7cf:    mov     rdi, r13
0x1a7d2:    call    0x18d10
0x1a7d7:    call    0x4860
0x1a7dc:    mov     eax, dword ptr [rax]
0x1a7de:    mov     rdx, qword ptr [rsp + 0x98]
0x1a7e6:    sub     rdx, qword ptr fs:[0x28]
0x1a7ef:    jne     0x1a8f0
0x1a7f5:    add     rsp, 0xa8
0x1a7fc:    pop     rbx
0x1a7fd:    pop     rbp
0x1a7fe:    pop     r12
0x1a800:    pop     r13
0x1a802:    pop     r14
0x1a804:    pop     r15
0x1a806:    ret
0x1a807:    mov     edi, r15d
0x1a80a:    call    0x4ab0
0x1a80f:    mov     rdi, rbp
0x1a812:    call    0x18d10
0x1a817:    jmp     0x1a7d7
0x1a7d7:
0x1a819:    mov     edi, r15d
0x1a81c:    call    0x4ab0
0x1a821:    mov     rcx, qword ptr [rsp + 0x30]
0x1a826:    test    r12d, r12d
0x1a829:    jns     0x1a82e
0x1a82b:    mov     r12d, ecx
0x1a82e:    xor     ebx, ebx
0x1a830:    xor     esi, esi
0x1a832:    movsxd  rdx, esi
0x1a835:    cmp     ebx, r14d
0x1a838:    jge     0x1a885
0x1a83a:    cmp     rdx, rcx
0x1a83d:    jge     0x1a885
0x1a83f:    lea     eax, [rsi + 1]
0x1a842:    cdqe
0x1a844:    cmp     rdx, rcx
0x1a847:    jl      0x1a85c
0x1a849:    jmp     0x1a87b
0x1a87b:
0x1a84b:    nop     dword ptr [rax + rax]
0x1a850:    lea     rdx, [rax + 1]
0x1a854:    cmp     rax, rcx
0x1a857:    je      0x1a878
0x1a859:    mov     rax, rdx
0x1a85c:    cmp     byte ptr [r13 + rax - 1], 0xa

```

```

0x1a862:    jne     0x1a850
0x1a864:    add     ebx, 1
0x1a867:    movsxd  rsi, eax
0x1a86a:    cmp     r14d, ebx
0x1a86d:    je      0x1a832
0x1a86f:    lea     rdx, [rax + 1]
0x1a873:    cmp     rax, rcx
0x1a876:    jne     0x1a859
0x1a878:    movsxd  rdx, esi
0x1a87b:    cmp     rcx, rdx
0x1a87e:    jle     0x1a885
0x1a880:    cmp     r14d, ebx
0x1a883:    jg      0x1a83f
0x1a885:    lea     r14, [r13 + rdx]
0x1a88a:    lea     r15d, [rsi + 1]
0x1a88e:    cmp     rcx, rdx
0x1a891:    jg      0x1a8aa
0x1a893:    jmp     0x1a8d4
0x1a8d4:
0x1a895:    nop     dword ptr [rax]
0x1a898:    movsxd  rax, r15d
0x1a89b:    add     r14, 1
0x1a89f:    add     r15d, 1
0x1a8a3:    cmp     rax, qword ptr [rsp + 0x30]
0x1a8a8:    jge     0x1a8d4
0x1a8aa:    cmp     byte ptr [r14], 0xa
0x1a8ae:    jne     0x1a898
0x1a8b0:    lea     rdi, [r13 + rsi]
0x1a8b5:    mov     byte ptr [r14], 0
0x1a8b9:    cmp     byte ptr [rdi], 0
0x1a8bc:    jne     0x1a8cb
0x1a8be:    add     ebx, 1
0x1a8c1:    cmp     r12d, ebx
0x1a8c4:    jle     0x1a8d4
0x1a8c6:    movsxd  rsi, r15d
0x1a8c9:    jmp     0x1a898
0x1a898:
0x1a8cb:    xor     eax, eax
0x1a8cd:    call    0x1a380
0x1a8d2:    jmp     0x1a8be
0x1a8be:
0x1a8d4:    mov     rdi, rbp
0x1a8d7:    call    0x18d10
0x1a8dc:    test    r13, r13
0x1a8df:    je      0x1a8e9
0x1a8e1:    mov     rdi, r13
0x1a8e4:    call    0x18d10
0x1a8e9:    xor     eax, eax

```

```

0x1a8eb:      jmp      0x1a7de
0x1a7de
0x1a8f0:      call     0x49d0
0x1a8f5:      nop      word ptr cs:[rax + rax]
0x1a900:      endbr64
0x1a904:      mov      edx, 0xffffffff
0x1a909:      xor      esi, esi
0x1a90b:      xor      eax, eax
0x1a90d:      jmp      0x1a730
0x1a730
0x1a912:      nop      word ptr cs:[rax + rax]
0x1a91d:      nop      dword ptr [rax]
0x1a920:      endbr64
0x1a924:      push     r14
0x1a926:      push     r13
0x1a928:      push     r12
0x1a92a:      mov      r12d, esi
0x1a92d:      push     rbp
0x1a92e:      push     rbx
0x1a92f:      sub      rsp, 0xa0
0x1a936:      mov      rax, qword ptr fs:[0x28]
0x1a93f:      mov      qword ptr [rsp + 0x98], rax
0x1a947:      xor      eax, eax
0x1a949:      call     0x1a080
0x1a94e:      xor      esi, esi
0x1a950:      mov      edx, 0x1b6
0x1a955:      mov      rbp, rax
0x1a958:      mov      rdi, rax
0x1a95b:      xor      eax, eax
0x1a95d:      call     0x4d80
0x1a962:      cmp      eax, -1
0x1a965:      je       0x1aa58
0x1a96b:      mov      rsi, rsp
0x1a96e:      mov      edi, eax
0x1a970:      mov      r14d, eax
0x1a973:      call     0x18ef0
0x1a978:      cmp      eax, -1
0x1a97b:      je       0x1aa58
0x1a981:      mov      eax, dword ptr [rsp + 0x30]
0x1a985:      lea      edi, [rax + 1]
0x1a988:      movsxd   rdi, edi
0x1a98b:      call     0x1d670
0x1a990:      mov      rdx, qword ptr [rsp + 0x30]
0x1a995:      mov      edi, r14d
0x1a998:      mov      rsi, rax
0x1a99b:      mov      r13, rax
0x1a99e:      call     0x47a0
0x1a9a3:      mov      edi, r14d

```



```

0x1a9a6:    mov     rbx, rax
0x1a9a9:    call   0x4ab0
0x1a9ae:    test    ebx, ebx
0x1a9b0:    jle     0x1aa88
0x1a9b6:    lea     eax, [rbx - 1]
0x1a9b9:    cmp     ebx, 1
0x1a9bc:    je      0x1a9f3
0x1a9be:    test    r12d, r12d
0x1a9c1:    je      0x1a9f3
0x1a9c3:    lea     edx, [rbx - 2]
0x1a9c6:    movsxd  rdx, edx
0x1a9c9:    nop     dword ptr [rax]
0x1a9d0:    cmp     byte ptr [r13 + rdx + 1], 0xa
0x1a9d6:    jne     0x1a9dc
0x1a9d8:    sub     r12d, 1
0x1a9dc:    test    r12d, r12d
0x1a9df:    mov     eax, edx
0x1a9e1:    setne   sil
0x1a9e5:    test    edx, edx
0x1a9e7:    setne   cl
0x1a9ea:    sub     rdx, 1
0x1a9ee:    test    sil, cl
0x1a9f1:    jne     0x1a9d0
0x1a9f3:    test    eax, eax
0x1a9f5:    je      0x1aa88
0x1a9fb:    cdqe
0x1a9fd:    jmp     0x1aa08
0x1aa08:
0x1a9ff:    nop
0x1aa00:    sub     rax, 1
0x1aa04:    test    eax, eax
0x1aa06:    je      0x1aa50
0x1aa08:    cmp     byte ptr [r13 + rax], 0xa
0x1aa0e:    mov     ebx, eax
0x1aa10:    jne     0x1aa00
0x1aa12:    mov     edx, 0x1b6
0x1aa17:    mov     esi, 0x201
0x1aa1c:    mov     rdi, rbp
0x1aa1f:    xor     eax, eax
0x1aa21:    call   0x4d80
0x1aa26:    mov     r12d, eax
0x1aa29:    cmp     eax, -1
0x1aa2c:    je      0x1aa50
0x1aa2e:    mov     rdx, qword ptr [rsp + 0x30]
0x1aa33:    lea     esi, [rbx + 1]
0x1aa36:    mov     edi, eax
0x1aa38:    movsxd  rsi, esi
0x1aa3b:    sub     rdx, rsi

```

```

0x1aa3e:      add     rsi, r13
0x1aa41:      call    0x4790
0x1aa46:      mov     edi, r12d
0x1aa49:      call    0x4ab0
0x1aa4e:      nop
0x1aa50:      mov     rdi, r13
0x1aa53:      call    0x18d10
0x1aa58:      mov     rdi, rbp
0x1aa5b:      call    0x18d10
0x1aa60:      mov     rax, qword ptr [rsp + 0x98]
0x1aa68:      sub     rax, qword ptr fs:[0x28]
0x1aa71:      jne     0x1aa8f
0x1aa73:      add     rsp, 0xa0
0x1aa7a:      xor     eax, eax
0x1aa7c:      pop     rbx
0x1aa7d:      pop     rbp
0x1aa7e:      pop     r12
0x1aa80:      pop     r13
0x1aa82:      pop     r14
0x1aa84:      ret
0x1aa85:      nop     dword ptr [rax]
0x1aa88:      test    r13, r13
0x1aa8b:      je      0x1aa58
0x1aa8d:      jmp     0x1aa50
0x1aa50
0x1aa8f:      call    0x49d0
0x1aa94:      nop     word ptr cs:[rax + rax]
0x1aa9f:      nop
0x1aaa0:      endbr64
0x1aaa4:      mov     r8d, edi
0x1aaa7:      xor     edx, edx
0x1aaa9:      mov     rdi, rsi
0x1aaac:      xor     eax, eax
0x1aaae:      mov     esi, r8d
0x1aab1:      jmp     0x1a130
0x1a130
0x1aab6:      nop     word ptr cs:[rax + rax]
0x1aac0:      endbr64
0x1aac4:      mov     esi, dword ptr [rip + 0x1101a]
0x1aaca:      mov     edx, 1
0x1aacf:      xor     eax, eax
0x1aad1:      jmp     0x1a130
0x1a130
0x1aad6:      nop     word ptr cs:[rax + rax]
0x1aae0:      endbr64
0x1aae4:      movsxd  rdx, dword ptr [rip + 0x11011]
0x1aaeb:      cmp     edx, dword ptr [rip + 0x10ff3]
0x1aaf1:      je      0x1ab08

```

```

0x1aaf3:    mov     rax, qword ptr [rip + 0x11006]
0x1aafa:    test    rax, rax
0x1aafd:    je      0x1ab0a
0x1aaff:    mov     rax, qword ptr [rax + rdx*8]
0x1ab03:    ret
0x1ab04:    nop     dword ptr [rax]
0x1ab08:    xor     eax, eax
0x1ab0a:    ret
0x1ab0b:    nop     dword ptr [rax + rax]
0x1ab10:    endbr64
0x1ab14:    mov     eax, dword ptr [rip + 0x10fe2]
0x1ab1a:    xor     r8d, r8d
0x1ab1d:    test    eax, eax
0x1ab1f:    je      0x1ab37
0x1ab21:    sub     eax, 1
0x1ab24:    mov     rdx, qword ptr [rip + 0x10fd5]
0x1ab2b:    mov     dword ptr [rip + 0x10fcb], eax
0x1ab31:    cdqe
0x1ab33:    mov     r8, qword ptr [rdx + rax*8]
0x1ab37:    mov     rax, r8
0x1ab3a:    ret
0x1ab3b:    nop     dword ptr [rax + rax]
0x1ab40:    endbr64
0x1ab44:    mov     eax, dword ptr [rip + 0x10fb2]
0x1ab4a:    cmp     eax, dword ptr [rip + 0x10f94]
0x1ab50:    je      0x1ab70
0x1ab52:    add     eax, 1
0x1ab55:    mov     rdx, qword ptr [rip + 0x10fa4]
0x1ab5c:    mov     dword ptr [rip + 0x10f9a], eax
0x1ab62:    cdqe
0x1ab64:    mov     rax, qword ptr [rdx + rax*8]
0x1ab68:    ret
0x1ab69:    nop     dword ptr [rax]
0x1ab70:    xor     eax, eax
0x1ab72:    ret
0x1ab73:    nop     word ptr cs:[rax + rax]
0x1ab7e:    nop
0x1ab80:    endbr64
0x1ab84:    mov     rax, qword ptr [rip + 0x10f75]
0x1ab8b:    ret
0x1ab8c:    nop     dword ptr [rax]
0x1ab90:    endbr64
0x1ab94:    sub     edi, dword ptr [rip + 0xfc4e]
0x1ab9a:    cmp     dword ptr [rip + 0x10f44], edi
0x1aba0:    jle     0x1abc0
0x1aba2:    test    edi, edi
0x1aba4:    js      0x1abc0
0x1aba6:    mov     rax, qword ptr [rip + 0x10f53]

```

```

0x1abad:    test    rax, rax
0x1abb0:    je      0x1abc2
0x1abb2:    movsxd  rdi, edi
0x1abb5:    mov     rax, qword ptr [rax + rdi*8]
0x1abb9:    ret
0x1abba:    nop     word ptr [rax + rax]
0x1abc0:    xor     eax, eax
0x1abc2:    ret
0x1abc3:    nop     word ptr cs:[rax + rax]
0x1abce:    nop
0x1abd0:    endbr64
0x1abd4:    cmp     edx, dword ptr [rip + 0x10f0a]
0x1abda:    push    rbx
0x1abdb:    mov     ebx, dword ptr [rip + 0x10f1b]
0x1abe1:    jg      0x1abf7
0x1abe3:    test    edx, edx
0x1abe5:    js      0x1abf7
0x1abe7:    cmp     qword ptr [rip + 0x10f11], 0
0x1abef:    je      0x1abf7
0x1abf1:    mov     dword ptr [rip + 0x10f05], edx
0x1abf7:    xor     edx, edx
0x1abf9:    xor     eax, eax
0x1abfb:    call    0x19de0
0x1ac00:    mov     edx, ebx
0x1ac02:    shr     edx, 0x1f
0x1ac05:    cmp     eax, -1
0x1ac08:    je      0x1ac10
0x1ac0a:    mov     eax, dword ptr [rip + 0x10eec]
0x1ac10:    cmp     ebx, dword ptr [rip + 0x10ece]
0x1ac16:    jg      0x1ac2c
0x1ac18:    test    dl, dl
0x1ac1a:    jne     0x1ac2c
0x1ac1c:    cmp     qword ptr [rip + 0x10edc], 0
0x1ac24:    je      0x1ac2c
0x1ac26:    mov     dword ptr [rip + 0x10ed0], ebx
0x1ac2c:    pop     rbx
0x1ac2d:    ret
0x1ac2e:    nop
0x1ac30:    endbr64
0x1ac34:    cmp     dword ptr [rip + 0x10eaa], edi
0x1ac3a:    jl      0x1ac60
0x1ac3c:    test    edi, edi
0x1ac3e:    js      0x1ac60
0x1ac40:    cmp     qword ptr [rip + 0x10eb8], 0
0x1ac48:    je      0x1ac60
0x1ac4a:    mov     dword ptr [rip + 0x10eac], edi
0x1ac50:    mov     eax, 1
0x1ac55:    ret

```

```

0x1ac56:      nop      word ptr cs:[rax + rax]
0x1ac60:      xor      eax, eax
0x1ac62:      ret
0x1ac63:      nop      word ptr cs:[rax + rax]
0x1ac6e:      nop
0x1ac70:      endbr64
0x1ac74:      movsxd   rax, dword ptr [rsi]
0x1ac77:      movzx    r8d, byte ptr [rip + 0xfb6e]
0x1ac7f:      cmp      byte ptr [rdi + rax], r8b
0x1ac83:      jne      0x1afd0
0x1ac89:      push     r15
0x1ac8b:      lea      ecx, [rax + 1]
0x1ac8e:      mov      r9d, edx
0x1ac91:      mov      rdx, rax
0x1ac94:      push     r14
0x1ac96:      movsxd   r10, ecx
0x1ac99:      push     r13
0x1ac9b:      mov      r13, rsi
0x1ac9e:      lea      rsi, [rdi + r10]
0x1aca2:      push     r12
0x1aca4:      push     rbp
0x1aca5:      push     rbx
0x1aca6:      sub      rsp, 0x18
0x1acaa:      movzx    eax, byte ptr [rsi]
0x1acad:      cmp      r8b, al
0x1acb0:      je       0x1af20
0x1acb6:      mov      r8d, 1
0x1acbc:      cmp      al, 0x2d
0x1acbe:      je       0x1aef8
0x1acc4:      lea      edx, [rax - 0x30]
0x1acc7:      cmp      dl, 9
0x1acca:      jbe      0x1ae68
0x1acd0:      mov      ebp, 1
0x1acd5:      xor      ebx, ebx
0x1acd7:      cmp      al, 0x3f
0x1acd9:      jne      0x1acef
0x1acdb:      add      ecx, 1
0x1acde:      xor      ebp, ebp
0x1ace0:      mov      ebx, 1
0x1ace5:      movsxd   r10, ecx
0x1ace8:      lea      rsi, [rdi + r10]
0x1acec:      movzx    eax, byte ptr [rsi]
0x1acef:      test     al, al
0x1acf1:      je       0x1b042
0x1acf7:      lea      rdx, [rdi + r10 + 1]
0x1acfc:      mov      r12d, ecx
0x1acff:      mov      r8, rsi
0x1ad02:      movabs   r10, 0x400000100000200

```

```

0x1ad0c:      jmp      0x1ad48
0x1ad48
0x1ad0e:      nop
0x1ad10:      cmp     al, 0x3a
0x1ad12:      ja      0x1ad1e
0x1ad14:      bt      r10, rax
0x1ad18:      jb      0x1afd8
0x1ad1e:      movsx   edi, al
0x1ad21:      cmp     edi, r9d
0x1ad24:      je      0x1b010
0x1ad2a:      cmp     al, 0xa
0x1ad2c:      je      0x1ae10
0x1ad32:      mov     r8, rdx
0x1ad35:      movzx   eax, byte ptr [rdx]
0x1ad38:      lea     rdx, [rdx + 1]
0x1ad3c:      add     r12d, 1
0x1ad40:      test    al, al
0x1ad42:      je      0x1ae10
0x1ad48:      test    ebx, ebx
0x1ad4a:      je      0x1ad10
0x1ad4c:      cmp     al, 0xa
0x1ad4e:      je      0x1ae10
0x1ad54:      cmp     al, 0x3f
0x1ad56:      jne     0x1ad32
0x1ad58:      mov     r15d, r12d
0x1ad5b:      mov     qword ptr [rsp + 8], r8
0x1ad60:      sub     r15d, ecx
0x1ad63:      mov     qword ptr [rsp], rsi
0x1ad67:      lea     edi, [r15 + 1]
0x1ad6b:      movsxd  r15, r15d
0x1ad6e:      movsxd  rdi, edi
0x1ad71:      call    0x1d670
0x1ad76:      mov     rsi, qword ptr [rsp]
0x1ad7a:      mov     rdx, r15
0x1ad7d:      mov     rdi, rax
0x1ad80:      mov     r14, rax
0x1ad83:      call    0x4880
0x1ad88:      mov     r8, qword ptr [rsp + 8]
0x1ad8d:      xor     eax, eax
0x1ad8f:      mov     byte ptr [r14 + r15], 0
0x1ad94:      cmp     byte ptr [r8], 0x3f
0x1ad98:      sete    al
0x1ad9b:      add     r12d, eax
0x1ad9e:      nop
0x1ada0:      mov     dword ptr [r13], r12d
0x1ada4:      mov     r12d, ebx
0x1ada7:      and     r12d, 1
0x1adab:      jmp     0x1add3

```

```

0x1add3
0x1adad:      nop      dword ptr [rax]
0x1adb0:      movsxd   rax, dword ptr [rip + 0x10d45]
0x1adb7:      je       0x1af78
0x1adb8:      test      r12b, r12b
0x1adc0:      jne       0x1af78
0x1adc6:      test      eax, eax
0x1adc8:      je       0x1ade8
0x1adca:      sub       eax, 1
0x1adcd:      mov       dword ptr [rip + 0x10d29], eax
0x1add3:      xor       eax, eax
0x1add5:      mov       edx, ebp
0x1add7:      mov       esi, 0xffffffff
0x1addc:      mov       rdi, r14
0x1addf:      call      0x19de0
0x1ade4:      test      eax, eax
0x1ade6:      jns       0x1adb0
0x1ade8:      mov       eax, dword ptr [rip + 0x10cf6]
0x1adee:      mov       rdi, r14
0x1adf1:      mov       dword ptr [rip + 0x10d05], eax
0x1adf7:      call      0x18d10
0x1adfc:      add       rsp, 0x18
0x1ae00:      xor       eax, eax
0x1ae02:      pop       rbx
0x1ae03:      pop       rbp
0x1ae04:      pop       r12
0x1ae06:      pop       r13
0x1ae08:      pop       r14
0x1ae0a:      pop       r15
0x1ae0c:      ret
0x1ae0d:      nop      dword ptr [rax]
0x1ae10:      mov       r15d, r12d
0x1ae13:      sub       r15d, ecx
0x1ae16:      lea       edi, [r15 + 1]
0x1ae1a:      movsxd   r15, r15d
0x1ae1d:      movsxd   rdi, edi
0x1ae20:      mov       qword ptr [rsp + 8], r8
0x1ae25:      mov       qword ptr [rsp], rsi
0x1ae29:      call      0x1d670
0x1ae2e:      mov       rsi, qword ptr [rsp]
0x1ae32:      mov       rdx, r15
0x1ae35:      mov       rdi, rax
0x1ae38:      mov       r14, rax
0x1ae3b:      call      0x4880
0x1ae40:      test      ebx, ebx
0x1ae42:      mov       byte ptr [r14 + r15], 0
0x1ae47:      mov       r8, qword ptr [rsp + 8]
0x1ae4c:      je       0x1ada0

```

```

0x1ae52:    xor     eax, eax
0x1ae54:    cmp     byte ptr [r8], 0x3f
0x1ae58:    sete    al
0x1ae5b:    add     r12d, eax
0x1ae5e:    jmp     0x1ada0
0x1ada0:
0x1ae63:    nop     dword ptr [rax + rax]
0x1ae68:    lea     edx, [rcx + 1]
0x1ae6b:    xor     ecx, ecx
0x1ae6d:    movsxd  rdx, edx
0x1ae70:    sub     eax, 0x30
0x1ae73:    lea     ecx, [rcx + rcx*4]
0x1ae76:    mov     r9d, edx
0x1ae79:    movsx   eax, al
0x1ae7c:    lea     ecx, [rax + rcx*2]
0x1ae7f:    movzx   eax, byte ptr [rdi + rdx]
0x1ae83:    add     rdx, 1
0x1ae87:    lea     esi, [rax - 0x30]
0x1ae8a:    cmp     sil, 9
0x1ae8e:    jbe     0x1ae70
0x1ae90:    mov     dword ptr [r13], r9d
0x1ae94:    mov     eax, dword ptr [rip + 0x10c4a]
0x1ae9a:    mov     edx, dword ptr [rip + 0xf948]
0x1aea0:    cmp     r8d, -1
0x1aea4:    jne     0x1aead
0x1aea6:    lea     esi, [rax + rdx]
0x1aea9:    sub     esi, ecx
0x1aeab:    mov     ecx, esi
0x1aead:    sub     ecx, edx
0x1aeaf:    cmp     ecx, eax
0x1aeb1:    jge     0x1adfc
0x1aeb7:    test    ecx, ecx
0x1aeb9:    js      0x1adfc
0x1aebf:    mov     rax, qword ptr [rip + 0x10c3a]
0x1aec6:    test    rax, rax
0x1aec9:    je      0x1adfc
0x1aecf:    movsxd  rcx, ecx
0x1aed2:    mov     rax, qword ptr [rax + rcx*8]
0x1aed6:    test    rax, rax
0x1aed9:    je      0x1adfc
0x1aedef:    mov     rax, qword ptr [rax]
0x1aee2:    add     rsp, 0x18
0x1aee6:    pop     rbx
0x1aee7:    pop     rbp
0x1aee8:    pop     r12
0x1aeea:    pop     r13
0x1aeec:    pop     r14
0x1aeef:    pop     r15

```



```

0x1aef0:      ret
0x1aef1:      nop        dword ptr [rax]
0x1aef8:      lea        ecx, [rdx + 2]
0x1aefb:      mov        r8d, 0xffffffff
0x1af01:      movsxd     r10, ecx
0x1af04:      lea        rsi, [rdi + r10]
0x1af08:      movzx      eax, byte ptr [rsi]
0x1af0b:      lea        edx, [rax - 0x30]
0x1af0e:      cmp        dl, 9
0x1af11:      ja         0x1acd0
0x1af17:      jmp        0x1ae68
0x1ae68
0x1af1c:      nop        dword ptr [rax]
0x1af20:      mov        ecx, dword ptr [rip + 0x10bbe]
0x1af26:      mov        eax, dword ptr [rip + 0xf8bc]
0x1af2c:      add        edx, 2
0x1af2f:      mov        dword ptr [r13], edx
0x1af33:      lea        eax, [rcx + rax - 1]
0x1af37:      sub        eax, dword ptr [rip + 0xf8ab]
0x1af3d:      cmp        eax, dword ptr [rip + 0x10ba1]
0x1af43:      jge        0x1adfc
0x1af49:      test       eax, eax
0x1af4b:      js         0x1adfc
0x1af51:      mov        rdx, qword ptr [rip + 0x10ba8]
0x1af58:      test       rdx, rdx
0x1af5b:      je         0x1adfc
0x1af61:      cdqe
0x1af63:      mov        rax, qword ptr [rdx + rax*8]
0x1af67:      test       rax, rax
0x1af6a:      jne        0x1aedf
0x1af70:      jmp        0x1adfc
0x1adfc
0x1af75:      nop        dword ptr [rax]
0x1af78:      mov        edx, dword ptr [rip + 0x10b66]
0x1af7e:      cmp        edx, eax
0x1af80:      je         0x1b03b
0x1af86:      mov        rbp, qword ptr [rip + 0x10b73]
0x1af8d:      test       rbp, rbp
0x1af90:      je         0x1af97
0x1af92:      mov        rbp, qword ptr [rbp + rax*8]
0x1af97:      mov        dword ptr [rip + 0x10b5f], edx
0x1af9d:      test       ebx, ebx
0x1af9f:      je         0x1b02e
0x1afa5:      mov        rdi, qword ptr [rip + 0x10b44]
0x1afac:      test       rdi, rdi
0x1afaf:      je         0x1afb6
0x1afb1:      call       0x18d10
0x1afb6:      mov        qword ptr [rip + 0x10b33], r14

```

```

0x1afb0:    mov     rax, qword ptr [rbp]
0x1afc1:    add     rsp, 0x18
0x1afc5:    pop     rbx
0x1afc6:    pop     rbp
0x1afc7:    pop     r12
0x1afc9:    pop     r13
0x1afcb:    pop     r14
0x1afcd:    pop     r15
0x1afcf:    ret
0x1afd0:    xor     eax, eax
0x1afd2:    ret
0x1afd3:    nop     dword ptr [rax + rax]
0x1afd8:    mov     eax, r12d
0x1afdb:    mov     qword ptr [rsp], rsi
0x1afdf:    sub     eax, ecx
0x1afe1:    movsxd  r15, eax
0x1afe4:    add     eax, 1
0x1afe7:    movsxd  rdi, eax
0x1afea:    call    0x1d670
0x1afef:    mov     r14, rax
0x1aff2:    mov     rsi, qword ptr [rsp]
0x1aff6:    mov     rdx, r15
0x1aff9:    mov     rdi, r14
0x1affc:    call    0x4880
0x1b001:    mov     byte ptr [r14 + r15], 0
0x1b006:    jmp     0x1ada0
0x1ada0:
0x1b00b:    nop     dword ptr [rax + rax]
0x1b010:    mov     r15d, r12d
0x1b013:    mov     qword ptr [rsp], rsi
0x1b017:    sub     r15d, ecx
0x1b01a:    lea     edi, [r15 + 1]
0x1b01e:    movsxd  r15, r15d
0x1b021:    movsxd  rdi, edi
0x1b024:    call    0x1d670
0x1b029:    mov     r14, rax
0x1b02c:    jmp     0x1aff2
0x1aff2:
0x1b02e:    mov     rdi, r14
0x1b031:    call    0x18d10
0x1b036:    jmp     0x1afb0
0x1afb0:
0x1b03b:    xor     ebp, ebp
0x1b03d:    jmp     0x1af97
0x1af97:
0x1b042:    mov     r12d, ecx
0x1b045:    mov     r8, rsi
0x1b048:    xor     r15d, r15d

```

```

0x1b04b:      mov     edi, 1
0x1b050:      jmp     0x1ae20
0x1ae20
0x1b055:      nop     word ptr cs:[rax + rax]
0x1b060:      endbr64
0x1b064:      push    r15
0x1b066:      push    r14
0x1b068:      push    r13
0x1b06a:      push    r12
0x1b06c:      push    rbp
0x1b06d:      push    rbx
0x1b06e:      sub     rsp, 0x28
0x1b072:      movzx   r14d, byte ptr [rdi]
0x1b076:      test    r14b, r14b
0x1b079:      je      0x1b30d
0x1b07f:      mov     qword ptr [rsp], 0
0x1b087:      mov     r12, rdi
0x1b08a:      mov     r13, rdi
0x1b08d:      xor     ebx, ebx
0x1b08f:      mov     dword ptr [rsp + 8], 0
0x1b097:      xor     ebp, ebp
0x1b099:      mov     dword ptr [rsp + 0xc], 0
0x1b0a1:      jmp     0x1b0c0
0x1b0c0
0x1b0a3:      nop     dword ptr [rax + rax]
0x1b0a8:      add     ebx, 1
0x1b0ab:      movsxd  rbp, ebx
0x1b0ae:      lea     r13, [r12 + rbp]
0x1b0b2:      movzx   r14d, byte ptr [r13]
0x1b0b7:      test    r14b, r14b
0x1b0ba:      je      0x1b315
0x1b0c0:      cmp     r14b, 0x20
0x1b0c4:      je      0x1b0a8
0x1b0c6:      cmp     r14b, 9
0x1b0ca:      je      0x1b0a8
0x1b0cc:      cmp     byte ptr [rip + 0x10a0d], r14b
0x1b0d3:      je      0x1b315
0x1b0d9:      movsx   r15d, r14b
0x1b0dd:      lea     rdi, [rip + 0x7379]
0x1b0e4:      mov     esi, r15d
0x1b0e7:      call    0x4a00
0x1b0ec:      test    rax, rax
0x1b0ef:      je      0x1b190
0x1b0f5:      add     ebx, 1
0x1b0f8:      mov     r14d, 1
0x1b0fe:      mov     r9d, 2
0x1b104:      movsxd  rbp, ebx
0x1b107:      lea     rcx, [r12 + rbp]

```

```

0x1b10b:    mov     eax, dword ptr [rsp + 8]
0x1b10f:    add     eax, 2
0x1b112:    cmp     eax, dword ptr [rsp + 0xc]
0x1b116:    jge     0x1b2a0
0x1b11c:    movsxd  r8, dword ptr [rsp + 8]
0x1b121:    mov     rdx, qword ptr [rsp]
0x1b125:    mov     rdi, r9
0x1b128:    mov     qword ptr [rsp + 0x18], rcx
0x1b12d:    shl     r8, 3
0x1b131:    lea     r15, [rdx + r8]
0x1b135:    mov     qword ptr [rsp + 0x10], r8
0x1b13a:    call    0x1d670
0x1b13f:    mov     rdx, r14
0x1b142:    mov     rsi, r13
0x1b145:    mov     qword ptr [r15], rax
0x1b148:    mov     rdi, rax
0x1b14b:    call    0x4880
0x1b150:    mov     rax, qword ptr [r15]
0x1b153:    mov     r8, qword ptr [rsp + 0x10]
0x1b158:    mov     rcx, qword ptr [rsp + 0x18]
0x1b15d:    add     dword ptr [rsp + 8], 1
0x1b162:    mov     byte ptr [rax + r14], 0
0x1b167:    mov     rax, qword ptr [rsp]
0x1b16b:    mov     qword ptr [rax + r8 + 8], 0
0x1b174:    movzx   r14d, byte ptr [rcx]
0x1b178:    test    r14b, r14b
0x1b17b:    je      0x1b315
0x1b181:    mov     r13, rcx
0x1b184:    jmp     0x1b0c0
0x1b0c0:
0x1b189:    nop     dword ptr [rax]
0x1b190:    mov     esi, r15d
0x1b193:    lea     rdi, [rip + 0x72c7]
0x1b19a:    call    0x4a00
0x1b19f:    test    rax, rax
0x1b1a2:    je      0x1b369
0x1b1a8:    movsx   edx, byte ptr [r12 + rbp + 1]
0x1b1ae:    cmp     dl, r14b
0x1b1b1:    je      0x1b2d8
0x1b1b7:    cmp     edx, 0x26
0x1b1ba:    je      0x1b340
0x1b1c0:    cmp     edx, 0x3e
0x1b1c3:    jne     0x1b3ac
0x1b1c9:    cmp     r14b, 0x26
0x1b1cd:    je      0x1b34e
0x1b1d3:    cmp     r14b, 0x24
0x1b1d7:    jne     0x1b0f5
0x1b1dd:    mov     edx, ebx

```

```

0x1b1df:      xor     r15d, r15d
0x1b1e2:      movsxd  r8, edx
0x1b1e5:      lea     rcx, [r12 + r8]
0x1b1e9:      mov     rbp, r8
0x1b1ec:      movzx   eax, byte ptr [rcx]
0x1b1ef:      test    al, al
0x1b1f1:      je      0x1b404
0x1b1f7:      mov     qword ptr [rsp + 0x10], r13
0x1b1fc:      lea     r14, [rip + 0x7265]
0x1b203:      mov     r13, r8
0x1b206:      mov     dword ptr [rsp + 0x18], ebx
0x1b20a:      mov     rbx, r12
0x1b20d:      mov     r12d, edx
0x1b210:      jmp     0x1b265
0x1b265:
0x1b212:      nop     word ptr [rax + rax]
0x1b218:      movsx   ebp, al
0x1b21b:      test    r15d, r15d
0x1b21e:      jne     0x1b290
0x1b220:      mov     esi, ebp
0x1b222:      lea     rdi, [rip + 0x7243]
0x1b229:      call    0x4a00
0x1b22e:      test    rax, rax
0x1b231:      jne     0x1b3c0
0x1b237:      mov     esi, ebp
0x1b239:      mov     rdi, r14
0x1b23c:      call    0x4a00
0x1b241:      lea     esi, [r12 + 1]
0x1b246:      test    rax, rax
0x1b249:      cmovne  r15d, ebp
0x1b24d:      movsxd  rbp, esi
0x1b250:      mov     r12d, esi
0x1b253:      lea     rcx, [rbx + rbp]
0x1b257:      mov     r13, rbp
0x1b25a:      movzx   eax, byte ptr [rcx]
0x1b25d:      test    al, al
0x1b25f:      je      0x1b389
0x1b265:      cmp     al, 0x5c
0x1b267:      jne     0x1b218
0x1b269:      cmp     byte ptr [rbx + r13 + 1], 0xa
0x1b26f:      lea     esi, [r12 + 1]
0x1b274:      je      0x1b330
0x1b27a:      xor     eax, eax
0x1b27c:      cmp     r15d, 0x27
0x1b280:      setne   al
0x1b283:      add     esi, eax
0x1b285:      jmp     0x1b24d
0x1b24d:

```

```

0x1b287:      nop      word ptr [rax + rax]
0x1b290:      xor      eax, eax
0x1b292:      cmp      r15d, ebp
0x1b295:      lea      esi, [r12 + 1]
0x1b29a:      cmov     r15d, eax
0x1b29e:      jmp      0x1b24d
0x1b24d
0x1b2a0:      add      dword ptr [rsp + 0xc], 0xa
0x1b2a5:      movsxd   rsi, dword ptr [rsp + 0xc]
0x1b2aa:      mov      rdi, qword ptr [rsp]
0x1b2ae:      mov      qword ptr [rsp + 0x18], rcx
0x1b2b3:      shl      rsi, 3
0x1b2b7:      mov      qword ptr [rsp + 0x10], r9
0x1b2bc:      call     0x1d690
0x1b2c1:      mov      rcx, qword ptr [rsp + 0x18]
0x1b2c6:      mov      r9, qword ptr [rsp + 0x10]
0x1b2cb:      mov      qword ptr [rsp], rax
0x1b2cf:      jmp      0x1b11c
0x1b11c
0x1b2d4:      nop      dword ptr [rax]
0x1b2d8:      cmp      edx, 0x24
0x1b2db:      je       0x1b1d3
0x1b2e1:      mov      eax, ebx
0x1b2e3:      cmp      edx, 0x3c
0x1b2e6:      je       0x1b3f0
0x1b2ec:      add      eax, 2
0x1b2ef:      mov      r14d, eax
0x1b2f2:      movsxd   rbp, eax
0x1b2f5:      sub      r14d, ebx
0x1b2f8:      lea      rcx, [r12 + rbp]
0x1b2fc:      mov      ebx, eax
0x1b2fe:      lea      r9d, [r14 + 1]
0x1b302:      movsxd   r14, r14d
0x1b305:      movsxd   r9, r9d
0x1b308:      jmp      0x1b10b
0x1b10b
0x1b30d:      mov      qword ptr [rsp], 0
0x1b315:      mov      rax, qword ptr [rsp]
0x1b319:      add      rsp, 0x28
0x1b31d:      pop      rbx
0x1b31e:      pop      rbp
0x1b31f:      pop      r12
0x1b321:      pop      r13
0x1b323:      pop      r14
0x1b325:      pop      r15
0x1b327:      ret
0x1b328:      nop      dword ptr [rax + rax]
0x1b330:      lea      esi, [r12 + 2]

```

```

0x1b335:      jmp      0x1b24d
0x1b24d
0x1b33a:      nop      word ptr [rax + rax]
0x1b340:      mov      eax, r14d
0x1b343:      and      eax, 0xffffffff
0x1b346:      cmp      al, 0x3c
0x1b348:      jne      0x1b1d3
0x1b34e:      add      ebx, 2
0x1b351:      mov      r14d, 2
0x1b357:      mov      r9d, 3
0x1b35d:      movsxd    rbp, ebx
0x1b360:      lea      rcx, [r12 + rbp]
0x1b364:      jmp      0x1b10b
0x1b10b
0x1b369:      mov      esi, r15d
0x1b36c:      lea      rdi, [rip + 0x70f5]
0x1b373:      call     0x4a00
0x1b378:      test     rax, rax
0x1b37b:      je       0x1b1dd
0x1b381:      lea      edx, [rbx + 1]
0x1b384:      jmp      0x1b1e2
0x1b1e2
0x1b389:      mov      r12, rbx
0x1b38c:      mov      ebx, dword ptr [rsp + 0x18]
0x1b390:      mov      r14d, esi
0x1b393:      mov      r13, qword ptr [rsp + 0x10]
0x1b398:      sub      r14d, ebx
0x1b39b:      mov      ebx, esi
0x1b39d:      lea      r9d, [r14 + 1]
0x1b3a1:      movsxd    r14, r14d
0x1b3a4:      movsxd    r9, r9d
0x1b3a7:      jmp      0x1b10b
0x1b10b
0x1b3ac:      cmp      r14b, 0x24
0x1b3b0:      jne      0x1b0f5
0x1b3b6:      cmp      edx, 0x28
0x1b3b9:      je       0x1b34e
0x1b3bb:      jmp      0x1b1d3
0x1b1d3
0x1b3c0:      mov      edx, r12d
0x1b3c3:      mov      r12, rbx
0x1b3c6:      mov      ebx, dword ptr [rsp + 0x18]
0x1b3ca:      mov      r8, r13
0x1b3cd:      mov      r14d, edx
0x1b3d0:      mov      r13, qword ptr [rsp + 0x10]
0x1b3d5:      mov      rbp, r8
0x1b3d8:      lea      rcx, [r12 + r8]
0x1b3dc:      sub      r14d, ebx

```

```

0x1b3df:      mov     ebx, edx
0x1b3e1:      lea     r9d, [r14 + 1]
0x1b3e5:      movsxd  r14, r14d
0x1b3e8:      movsxd  r9, r9d
0x1b3eb:      jmp     0x1b10b
0x1b10b
0x1b3f0:      cmp     byte ptr [r12 + rbp + 2], 0x2d
0x1b3f6:      jne     0x1b2ec
0x1b3fc:      lea     eax, [rbx + 1]
0x1b3ff:      jmp     0x1b2ec
0x1b2ec
0x1b404:      mov     r14d, edx
0x1b407:      sub     r14d, ebx
0x1b40a:      mov     ebx, edx
0x1b40c:      lea     r9d, [r14 + 1]
0x1b410:      movsxd  r14, r14d
0x1b413:      movsxd  r9, r9d
0x1b416:      jmp     0x1b10b
0x1b10b
0x1b41b:      nop     dword ptr [rax + rax]
0x1b420:      endbr64
0x1b424:      push    r15
0x1b426:      xor     eax, eax
0x1b428:      push    r14
0x1b42a:      push    r13
0x1b42c:      push    r12
0x1b42e:      mov     r12d, esi
0x1b431:      push    rbp
0x1b432:      push    rbx
0x1b433:      mov     ebx, edi
0x1b435:      mov     rdi, rdx
0x1b438:      sub     rsp, 8
0x1b43c:      call    0x1b060
0x1b441:      test    rax, rax
0x1b444:      je      0x1b598
0x1b44a:      cmp     qword ptr [rax], 0
0x1b44e:      mov     rbp, rax
0x1b451:      je      0x1b590
0x1b457:      mov     eax, 1
0x1b45c:      nop     dword ptr [rax]
0x1b460:      mov     r14d, eax
0x1b463:      add     rax, 1
0x1b467:      cmp     qword ptr [rbp + rax*8 - 8], 0
0x1b46d:      jne     0x1b460
0x1b46f:      lea     eax, [r14 + r12 - 1]
0x1b474:      test    r12d, r12d
0x1b477:      cmovs   r12d, eax
0x1b47b:      lea     eax, [r14 + rbx - 1]

```



```

0x1b480:    test    ebx, ebx
0x1b482:    cmovs   ebx, eax
0x1b485:    lea     eax, [r12 + 1]
0x1b48a:    cmp     r12d, 0x24
0x1b48e:    jne     0x1b497
0x1b490:    lea     r12d, [r14 - 1]
0x1b494:    mov     eax, r14d
0x1b497:    lea     edx, [r14 - 1]
0x1b49b:    cmp     ebx, 0x24
0x1b49e:    cmove    ebx, edx
0x1b4a1:    mov     edx, eax
0x1b4a3:    cmp     ebx, eax
0x1b4a5:    cmovge   edx, ebx
0x1b4a8:    xor     r13d, r13d
0x1b4ab:    cmp     edx, r14d
0x1b4ae:    jg      0x1b547
0x1b4b4:    mov     ecx, ebx
0x1b4b6:    or      ecx, eax
0x1b4b8:    js      0x1b547
0x1b4be:    cmp     ebx, eax
0x1b4c0:    jge     0x1b59d
0x1b4c6:    movsxd   rbx, ebx
0x1b4c9:    xor     r15d, r15d
0x1b4cc:    mov     r13, rbx
0x1b4cf:    nop
0x1b4d0:    mov     rdi, qword ptr [rbp + r13*8]
0x1b4d5:    add     r13, 1
0x1b4d9:    call    0x49b0
0x1b4de:    lea     r15d, [r15 + rax + 1]
0x1b4e3:    lea     eax, [r13 - 1]
0x1b4e7:    cmp     r12d, eax
0x1b4ea:    jg      0x1b4d0
0x1b4ec:    lea     edi, [r15 + 1]
0x1b4f0:    xor     r15d, r15d
0x1b4f3:    movsxd   rdi, edi
0x1b4f6:    call    0x1d670
0x1b4fb:    mov     r13, rax
0x1b4fe:    nop
0x1b500:    mov     rsi, qword ptr [rbp + rbx*8]
0x1b505:    movsxd   rdi, r15d
0x1b508:    add     rdi, r13
0x1b50b:    call    0x48b0
0x1b510:    mov     rdi, qword ptr [rbp + rbx*8]
0x1b515:    call    0x49b0
0x1b51a:    add     r15d, eax
0x1b51d:    cmp     r12d, ebx
0x1b520:    jle     0x1b53b
0x1b522:    lea     eax, [r15 + 1]

```

```

0x1b526:      movsxd  r15, r15d
0x1b529:      movsxd  rdx, eax
0x1b52c:      mov     byte ptr [r13 + r15], 0x20
0x1b532:      mov     r15d, eax
0x1b535:      mov     byte ptr [r13 + rdx], 0
0x1b53b:      add     rbx, 1
0x1b53f:      lea     eax, [rbx - 1]
0x1b542:      cmp     r12d, eax
0x1b545:      jg      0x1b500
0x1b547:      test    r14d, r14d
0x1b54a:      je      0x1b571
0x1b54c:      lea     eax, [r14 - 1]
0x1b550:      mov     rbx, rbp
0x1b553:      lea     r12, [rbp + rax*8 + 8]
0x1b558:      nop     dword ptr [rax + rax]
0x1b560:      mov     rdi, qword ptr [rbx]
0x1b563:      add     rbx, 8
0x1b567:      call    0x18d10
0x1b56c:      cmp     r12, rbx
0x1b56f:      jne     0x1b560
0x1b571:      mov     rdi, rbp
0x1b574:      call    0x18d10
0x1b579:      add     rsp, 8
0x1b57d:      mov     rax, r13
0x1b580:      pop     rbx
0x1b581:      pop     rbp
0x1b582:      pop     r12
0x1b584:      pop     r13
0x1b586:      pop     r14
0x1b588:      pop     r15
0x1b58a:      ret
0x1b58b:      nop     dword ptr [rax + rax]
0x1b590:      xor     r14d, r14d
0x1b593:      jmp     0x1b46f
0x1b46f
0x1b598:      xor     r13d, r13d
0x1b59b:      jmp     0x1b579
0x1b579
0x1b59d:      mov     edi, 1
0x1b5a2:      call    0x1d670
0x1b5a7:      mov     r13, rax
0x1b5aa:      jmp     0x1b547
0x1b547
0x1b5ac:      nop     dword ptr [rax]
0x1b5b0:      endbr64
0x1b5b4:      push    r15
0x1b5b6:      push    r14
0x1b5b8:      push    r13

```

```

0x1b5ba:    push    r12
0x1b5bc:    push    rbp
0x1b5bd:    push    rbx
0x1b5be:    mov     rbx, rdi
0x1b5c1:    sub     rsp, 0x88
0x1b5c8:    mov     qword ptr [rsp + 0x20], rdi
0x1b5cd:    mov     qword ptr [rsp + 0x30], rsi
0x1b5d2:    mov     rax, qword ptr fs:[0x28]
0x1b5db:    mov     qword ptr [rsp + 0x78], rax
0x1b5e0:    xor     eax, eax
0x1b5e2:    cmp     byte ptr [rip + 0xf204], 0
0x1b5e9:    je      0x1b656
0x1b5eb:    mov     edi, 0x100
0x1b5f0:    call    0x1d670
0x1b5f5:    mov     rbx, qword ptr [rsp + 0x20]
0x1b5fa:    mov     byte ptr [rax], 0
0x1b5fd:    mov     r15, rax
0x1b600:    mov     rdi, rbx
0x1b603:    call    0x49b0
0x1b608:    mov     dword ptr [rsp + 0xc], eax
0x1b60c:    mov     r12, rax
0x1b60f:    movzx   eax, byte ptr [rbx]
0x1b612:    cmp     al, byte ptr [rip + 0xf1d4]
0x1b618:    je      0x1c1e0
0x1b61e:    movzx   ebp, byte ptr [rip + 0xf1c8]
0x1b625:    mov     r13, qword ptr [rip + 0xf1c4]
0x1b62c:    lea     rbx, [rbx + 1]
0x1b630:    test    al, al
0x1b632:    jne     0x1b6c7
0x1b638:    nop     dword ptr [rax + rax]
0x1b640:    test    bpl, bpl
0x1b643:    je      0x1b6d8
0x1b649:    mov     rdi, r15
0x1b64c:    call    0x18d10
0x1b651:    mov     rbx, qword ptr [rsp + 0x20]
0x1b656:    mov     rdi, rbx
0x1b659:    call    0x49b0
0x1b65e:    lea     rdi, [rax + 1]
0x1b662:    call    0x1d670
0x1b667:    mov     rsi, rbx
0x1b66a:    mov     rdi, rax
0x1b66d:    call    0x48b0
0x1b672:    mov     rcx, qword ptr [rsp + 0x30]
0x1b677:    mov     qword ptr [rcx], rax
0x1b67a:    xor     eax, eax
0x1b67c:    mov     rdx, qword ptr [rsp + 0x78]
0x1b681:    sub     rdx, qword ptr fs:[0x28]
0x1b68a:    jne     0x1c487

```

```

0x1b690:    add     rsp, 0x88
0x1b697:    pop     rbx
0x1b698:    pop     rbp
0x1b699:    pop     r12
0x1b69b:    pop     r13
0x1b69d:    pop     r14
0x1b69f:    pop     r15
0x1b6a1:    ret
0x1b6a2:    nop     word ptr [rax + rax]
0x1b6a8:    mov     rdi, r13
0x1b6ab:    call    0x4a00
0x1b6b0:    test    rax, rax
0x1b6b3:    je      0x1b6d8
0x1b6b5:    nop     dword ptr [rax]
0x1b6b8:    movzx   eax, byte ptr [rbx]
0x1b6bb:    add     rbx, 1
0x1b6bf:    test    al, al
0x1b6c1:    je      0x1b640
0x1b6c7:    movsx   esi, byte ptr [rbx]
0x1b6ca:    cmp     al, bpl
0x1b6cd:    jne     0x1b6b8
0x1b6cf:    test    esi, esi
0x1b6d1:    je      0x1b6b8
0x1b6d3:    jmp     0x1b6a8
0x1b6a8
0x1b6d5:    nop     dword ptr [rax]
0x1b6d8:    mov     rbp, qword ptr [rsp + 0x20]
0x1b6dd:    test    r12d, r12d
0x1b6e0:    jle     0x1c40a
0x1b6e6:    mov     dword ptr [rsp + 4], 0x100
0x1b6ee:    xor     r13d, r13d
0x1b6f1:    xor     ebx, ebx
0x1b6f3:    mov     dword ptr [rsp + 0x28], 0
0x1b6fb:    mov     dword ptr [rsp + 0x3c], 0
0x1b703:    mov     dword ptr [rsp + 8], 0
0x1b70b:    jmp     0x1b758
0x1b758
0x1b70d:    nop     dword ptr [rax]
0x1b710:    cmp     r12b, byte ptr [rip + 0xf0d6]
0x1b717:    je      0x1b7d0
0x1b71d:    cmp     r12b, 0xfd
0x1b721:    je      0x1b7d0
0x1b727:    cmp     r12b, 0x5c
0x1b72b:    je      0x1bde0
0x1b731:    cmp     ebx, dword ptr [rsp + 4]
0x1b735:    jge     0x1bfe0
0x1b73b:    lea     eax, [rbx + 1]
0x1b73e:    mov     byte ptr [r15 + rbx], r12b

```

```

0x1b742:    movsxd  rdx, eax
0x1b745:    movsxd  rbx, eax
0x1b748:    mov     byte ptr [r15 + rdx], 0
0x1b74d:    add     r13d, 1
0x1b751:    cmp     r13d, dword ptr [rsp + 0xc]
0x1b756:    jge     0x1b79d
0x1b758:    movsxd  r14, r13d
0x1b75b:    mov     r10d, dword ptr [rsp + 8]
0x1b760:    lea     rdx, [rbp + r14]
0x1b765:    movzx   r12d, byte ptr [rdx]
0x1b769:    test    r10d, r10d
0x1b76c:    je      0x1b710
0x1b76e:    cmp     ebx, dword ptr [rsp + 4]
0x1b772:    jge     0x1be10
0x1b778:    lea     eax, [rbx + 1]
0x1b77b:    mov     byte ptr [r15 + rbx], r12b
0x1b77f:    add     r13d, 1
0x1b783:    movsxd  rdx, eax
0x1b786:    mov     dword ptr [rsp + 8], 0
0x1b78e:    movsxd  rbx, eax
0x1b791:    mov     byte ptr [r15 + rdx], 0
0x1b796:    cmp     r13d, dword ptr [rsp + 0xc]
0x1b79b:    jl      0x1b758
0x1b79d:    mov     rax, qword ptr [rsp + 0x30]
0x1b7a2:    mov     qword ptr [rax], r15
0x1b7a5:    cmp     qword ptr [rsp + 0x20], rbp
0x1b7aa:    je      0x1b7b4
0x1b7ac:    mov     rdi, rbp
0x1b7af:    call    0x18d10
0x1b7b4:    mov     ecx, dword ptr [rsp + 0x28]
0x1b7b8:    test    ecx, ecx
0x1b7ba:    jne     0x1c35b
0x1b7c0:    mov     edx, dword ptr [rsp + 0x3c]
0x1b7c4:    xor     eax, eax
0x1b7c6:    test    edx, edx
0x1b7c8:    setne   al
0x1b7cb:    jmp     0x1b67c
0x1b67c:
0x1b7d0:    lea     rcx, [rbp + r14 + 1]
0x1b7d5:    mov     qword ptr [rsp + 0x10], rdx
0x1b7da:    movsx   esi, byte ptr [rcx]
0x1b7dd:    test    esi, esi
0x1b7df:    je      0x1b830
0x1b7e1:    mov     rdi, qword ptr [rip + 0xf008]
0x1b7e8:    mov     qword ptr [rsp + 0x18], rcx
0x1b7ed:    call    0x4a00
0x1b7f2:    mov     rcx, qword ptr [rsp + 0x18]
0x1b7f7:    test    rax, rax

```

```

0x1b7fa:      je      0x1b830
0x1b7fc:      cmp     ebx, dword ptr [rsp + 4]
0x1b800:      jl      0x1b73b
0x1b806:      add     dword ptr [rsp + 4], 0x40
0x1b80b:      movsxd   rsi, dword ptr [rsp + 4]
0x1b810:      mov      rdi, r15
0x1b813:      call     0x1d690
0x1b818:      mov      rdx, qword ptr [rsp + 0x10]
0x1b81d:      mov      r15, rax
0x1b820:      movzx    r12d, byte ptr [rdx]
0x1b824:      jmp      0x1b73b
0x1b73b:
0x1b829:      nop      dword ptr [rax]
0x1b830:      mov      edi, 0x80
0x1b835:      mov      qword ptr [rsp + 0x18], rcx
0x1b83a:      call     0x1d670
0x1b83f:      mov      rcx, qword ptr [rsp + 0x18]
0x1b844:      mov      dword ptr [rsp + 0x6c], r13d
0x1b849:      mov      qword ptr [rsp + 0x10], rax
0x1b84e:      movzx    r12d, byte ptr [rcx]
0x1b852:      test     r12b, r12b
0x1b855:      je      0x1be30
0x1b85b:      movsx     esi, r12b
0x1b85f:      lea      rdi, [rip + 0x6c11]
0x1b866:      call     0x4a00
0x1b86b:      test     rax, rax
0x1b86e:      je      0x1bfc0
0x1b874:      lea      eax, [r13 + 1]
0x1b878:      lea      rsi, [rsp + 0x70]
0x1b87d:      xor      edx, edx
0x1b87f:      mov      byte ptr [rsp + 0x77], 0
0x1b884:      mov      dword ptr [rsp + 0x6c], eax
0x1b888:      movzx    eax, byte ptr [rip + 0xef5e]
0x1b88f:      lea      rdi, [rsp + 0x75]
0x1b894:      mov      dword ptr [rsp + 0x70], 0
0x1b89c:      mov      byte ptr [rsp + 0x76], al
0x1b8a0:      mov      byte ptr [rsp + 0x75], al
0x1b8a4:      xor      eax, eax
0x1b8a6:      call     0x1ac70
0x1b8ab:      mov      r12d, dword ptr [rsp + 0x6c]
0x1b8b0:      mov      r14, rax
0x1b8b3:      test     r14, r14
0x1b8b6:      je      0x1c435
0x1b8bc:      movsxd   rax, r12d
0x1b8bf:      xor      esi, esi
0x1b8c1:      mov      edx, r12d
0x1b8c4:      movzx    eax, byte ptr [rbp + rax]
0x1b8c9:      cmp      al, 0x3a

```

```

0x1b8cb:    jne     0x1b8df
0x1b8cd:    lea     edx, [r12 + 1]
0x1b8d2:    mov     esi, 1
0x1b8d7:    movsxd  rax, edx
0x1b8da:    movzx   eax, byte ptr [rbp + rax]
0x1b8df:    cmp     al, 0x25
0x1b8e1:    je      0x1c010
0x1b8e7:    cmp     al, 0x2d
0x1b8e9:    je      0x1c100
0x1b8ef:    jg      0x1bd48
0x1b8f5:    cmp     al, 0x24
0x1b8f7:    je      0x1bf80
0x1b8fd:    cmp     al, 0x2a
0x1b8ff:    jne     0x1bd54
0x1b905:    add     edx, 1
0x1b908:    mov     esi, 0x24
0x1b90d:    mov     edi, 1
0x1b912:    xor     eax, eax
0x1b914:    mov     dword ptr [rsp + 0x6c], edx
0x1b918:    mov     rdx, r14
0x1b91b:    call    0x1b420
0x1b920:    mov     r13, rax
0x1b923:    test    rax, rax
0x1b926:    je      0x1c080
0x1b92c:    lea     rax, [rip + 0x101ed]
0x1b933:    cmp     r13, rax
0x1b936:    je      0x1bdae
0x1b93c:    mov     rdi, r13
0x1b93f:    call    0x49b0
0x1b944:    lea     rdi, [rax + 1]
0x1b948:    call    0x1d670
0x1b94d:    mov     rsi, r13
0x1b950:    mov     rdi, rax
0x1b953:    call    0x48b0
0x1b958:    mov     rdi, r13
0x1b95b:    mov     r14, rax
0x1b95e:    call    0x18d10
0x1b963:    movsxd  rcx, dword ptr [rsp + 0x6c]
0x1b968:    cmp     byte ptr [rbp + rcx], 0x3a
0x1b96d:    mov     dword ptr [rsp + 0x44], ecx
0x1b971:    jne     0x1c3d4
0x1b977:    mov     dword ptr [rsp + 0x18], 0
0x1b97f:    lea     r12, [rip + 0x6afe]
0x1b986:    mov     dword ptr [rsp + 0x38], 0
0x1b98e:    mov     dword ptr [rsp + 0x40], ebx
0x1b992:    mov     qword ptr [rsp + 0x28], r15
0x1b997:    mov     r15, r14
0x1b99a:    mov     r14d, ecx

```

```

0x1b99d:      nop      dword ptr [rax]
0x1b9a0:      movzx    eax, byte ptr [rbp + rcx + 1]
0x1b9a5:      cmp      al, 0x67
0x1b9a7:      jne       0x1b9bf
0x1b9a9:      add       r14d, 1
0x1b9ad:      mov       dword ptr [rsp + 0x18], 1
0x1b9b5:      mov       dword ptr [rsp + 0x6c], r14d
0x1b9ba:      movzx     eax, byte ptr [rbp + rcx + 2]
0x1b9bf:      cmp      al, 0x26
0x1b9c1:      je        0x1be70
0x1b9c7:      sub       eax, 0x65
0x1b9ca:      cmp      al, 0xf
0x1b9cc:      ja        0x1b9e0
0x1b9ce:      movzx     eax, al
0x1b9d1:      movsxd    rax, dword ptr [r12 + rax*4]
0x1b9d5:      add       rax, r12
0x1b9d8:      jmp       rax
rax
->->->                                indirect JMP
0xe430: mov     rax, r13
0xe433: jmp     0xdfc7
0xdfc7
0xe438: mov     eax, dword ptr [rcx]
0xe43a: mov     dword ptr [r12], eax
0xe43e: mov     eax, edx
0xe440: mov     ecx, dword ptr [rcx + rax - 4]
0xe444: mov     dword ptr [r12 + rax - 4], ecx
0xe449: jmp     0xe1a7
0xe1a7
0xe44e: mov     eax, dword ptr [rbp - 0x60]
0xe451: mov     dword ptr [r12], eax
0xe455: mov     eax, edx
0xe457: mov     ecx, dword ptr [r10 + rax - 4]
0xe45c: mov     dword ptr [r12 + rax - 4], ecx
0xe461: jmp     0xe2d7
0xe2d7
0xe466: mov     eax, edx
0xe468: movzx   ecx, word ptr [r10 + rax - 2]
0xe46e: mov     word ptr [r12 + rax - 2], cx
0xe474: jmp     0xe2d7
0xe2d7
0xe479: call    0x49d0
0xe47e: nop
0xe480: endbr64
0xe484: push    r12
0xe486: mov     r12d, esi
0xe489: push    rbp
0xe48a: mov     rbp, rdi

```



```

0xe48d: mov     rdi, qword ptr [rip + 0x1d12c]
0xe494: push    rbx
0xe495: mov     ebx, edx
0xe497: test    rdi, rdi
0xe49a: je      0xe4a1
0xe49c: call    0x18d10
0xe4a1: mov     rdi, rbp
0xe4a4: call    0x1d7a0
0xe4a9: mov     rdi, rax
0xe4ac: mov     rbp, rax
0xe4af: mov     qword ptr [rip + 0x1d10a], rax
0xe4b6: call    0x49b0
0xe4bb: mov     rdi, rbp
0xe4be: mov     rsi, rax
0xe4c1: call    0x184d0
0xe4c6: mov     byte ptr [rip + 0x1d0ec], r12b
0xe4cd: xor     eax, eax
0xe4cf: mov     byte ptr [rip + 0x1d0e3], bl
0xe4d5: pop     rbx
0xe4d6: pop     rbp
0xe4d7: pop     r12
0xe4d9: jmp     0xdea0
0xdea0
0xe4de: nop
0xe4e0: endbr64
0xe4e4: push    rbp
0xe4e5: mov     rdi, qword ptr [rip + 0x1d0bc]
0xe4ec: call    0x1d7a0
0xe4f1: mov     rdi, qword ptr [rip + 0x1d0c8]
0xe4f8: mov     rbp, rax
0xe4fb: test    rdi, rdi
0xe4fe: je      0xe505
0xe500: call    0x18d10
0xe505: mov     rdi, rbp
0xe508: call    0x1d7a0
0xe50d: mov     rdi, rax
0xe510: mov     rbp, rax
0xe513: mov     qword ptr [rip + 0x1d0a6], rax
0xe51a: call    0x49b0
0xe51f: mov     rdi, rbp
0xe522: mov     rsi, rax
0xe525: call    0x184d0
0xe52a: xor     eax, eax
0xe52c: pop     rbp
0xe52d: mov     byte ptr [rip + 0x1d085], 0
0xe534: mov     byte ptr [rip + 0x1d07d], 0
0xe53b: jmp     0xdea0
0xdea0

```

```

0xe540: endbr64
0xe544: mov     rax, qword ptr [rsi]
0xe547: mov     rdx, qword ptr [rdi]
0xe54a: mov     rax, qword ptr [rax + 8]
0xe54e: sub     eax, dword ptr [rdx + 8]
0xe551: ret
0xe552: nop     word ptr cs:[rax + rax]
0xe55d: nop     dword ptr [rax]
0xe560: push    rbp
0xe561: mov     rbp, rdi
0xe564: mov     rdi, rdx
0xe567: push    rbx
0xe568: mov     ebx, esi
0xe56a: lea     rsi, [rip + 0x1bfff]
0xe571: sub     rsp, 8
0xe575: call    0x48b0
0xe57a: movsxd  rsi, ebx
0xe57d: mov     rcx, rax
0xe580: lea     rsi, [rsi + rsi*4]
0xe584: mov     rax, qword ptr [rbp + 0x38]
0xe588: shl     rsi, 4
0xe58c: mov     edx, dword ptr [rax + rsi + 0x10]
0xe590: mov     eax, 0x2d
0xe595: mov     edi, edx
0xe597: and     edi, 0xf000
0xe59d: cmp     edi, 0x8000
0xe5a3: je      0xe600
0xe5a5: mov     eax, 0x64
0xe5aa: cmp     edi, 0x4000
0xe5b0: je      0xe600
0xe5b2: mov     eax, 0x63
0xe5b7: cmp     edi, 0x2000
0xe5bd: je      0xe600
0xe5bf: mov     eax, 0x62
0xe5c4: cmp     edi, 0x6000
0xe5ca: je      0xe600
0xe5cc: mov     eax, 0x70
0xe5d1: cmp     edi, 0x1000
0xe5d7: je      0xe600
0xe5d9: mov     eax, 0x73
0xe5de: cmp     edi, 0xc000
0xe5e4: je      0xe600
0xe5e6: cmp     edi, 0xa000
0xe5ec: mov     eax, 0x6c
0xe5f1: mov     edi, 0x3f
0xe5f6: cmovne  eax, edi
0xe5f9: nop     dword ptr [rax]
0xe600: mov     byte ptr [rcx], al

```

```

0xe602: lea      rax, [rcx + 9]
0xe606: nop      word ptr cs:[rax + rax]
0xe610: test     dl, 1
0xe613: jne      0xe618
0xe615: mov      byte ptr [rax], 0x2d
0xe618: sub      rax, 1
0xe61c: shr      edx, 1
0xe61e: cmp      rax, rcx
0xe621: jne      0xe610
0xe623: mov      rax, qword ptr [rbp + 0x38]
0xe627: mov      eax, dword ptr [rax + rsi + 0x10]
0xe62b: test     ah, 8
0xe62e: je       0xe644
0xe630: cmp      byte ptr [rcx + 3], 0x78
0xe634: mov      edx, 0x73
0xe639: mov      esi, 0x53
0xe63e: cmovne   edx, esi
0xe641: mov      byte ptr [rcx + 3], dl
0xe644: test     ah, 4
0xe647: je       0xe65d
0xe649: cmp      byte ptr [rcx + 6], 0x78
0xe64d: mov      edx, 0x73
0xe652: mov      esi, 0x53
0xe657: cmovne   edx, esi
0xe65a: mov      byte ptr [rcx + 6], dl
0xe65d: test     ah, 2
0xe660: je       0xe676
0xe662: cmp      byte ptr [rcx + 9], 0x78
0xe666: mov      eax, 0x74
0xe66b: mov      edx, 0x54
0xe670: cmovne   eax, edx
0xe673: mov      byte ptr [rcx + 9], al
0xe676: add      rsp, 8
0xe67a: pop      rbx
0xe67b: pop      rbp
0xe67c: ret
0xe67d: nop      dword ptr [rax]
0xe680: endbr64
0xe684: push     r13
0xe686: push     r12
0xe688: push     rbp
0xe689: push     rbx
0xe68a: sub      rsp, 8
0xe68e: movzx    eax, byte ptr [rdi + 0x48]
0xe692: test     al, al
0xe694: sete     cl
0xe697: cmp      byte ptr [rsi + 0x48], 0
0xe69b: sete     dl

```

```

0xe69e: cmp     cl, dl
0xe6a0: je      0xe6c0
0xe6a2: cmp     al, 1
0xe6a4: sbb     r12d, r12d
0xe6a7: or      r12d, 1
0xe6ab: add     rsp, 8
0xe6af: mov     eax, r12d
0xe6b2: pop     rbx
0xe6b3: pop     rbp
0xe6b4: pop     r12
0xe6b6: pop     r13
0xe6b8: ret
0xe6b9: nop     dword ptr [rax]
0xe6c0: mov     r12d, dword ptr [rip + 0x1cf61]
0xe6c7: mov     rbx, rdi
0xe6ca: mov     rbp, rsi
0xe6cd: cmp     r12d, 8
0xe6d1: ja      0xe6e7
0xe6d3: lea     rdx, [rip + 0x131c6]
0xe6da: mov     eax, r12d
0xe6dd: movsxd  rax, dword ptr [rdx + rax*4]
0xe6e1: add     rax, rdx
0xe6e4: jmp     rax
rax
->->->                                indirect JMP
0xdf80: add     r14, 8
0xdf84: mov     rdi, r13
0xdf87: mov     rsi, r14
0xdf8a: call    0x1d690
0xdf8f: mov     rcx, qword ptr [rip + 0x1d62a]
0xdf96: movsxd  rdx, ebx
0xdf99: add     dword ptr [rbp - 0x74], 8
0xdf9d: mov     r13, rax
0xdfa0: movabs  rax, 0x2020202020202020
0xdfaa: mov     qword ptr [r13 + r12], rax
0xdfaf: movzx   eax, byte ptr [rcx + rdx]
0xdfb3: test    al, al
0xdfb5: jne     0xdf48
0xdfb7: nop     word ptr [rax + rax]
0xdfc0: movsxd  rax, dword ptr [rbp - 0x74]
0xdfc4: add     rax, r13
0xdfc7: mov     byte ptr [rax], 0
0xdfca: mov     rdi, r13
0xdfcd: call    0x49b0
0xdfd2: cmp     byte ptr [rip + 0x1d5df], 0
0xdfd9: mov     rdi, qword ptr [rip + 0x1d5d0]
0xdfe0: mov     rdx, rax
0xdfe3: mov     rax, qword ptr [rip + 0x1d5de]

```

```

0xdfea: mov     ecx, edx
0xdfec: jne     0xe350
0xdff2: movsxd  rax, dword ptr [rax + 0xc]
0xdff6: cmp     edx, eax
0xdff8: jl      0xe120
0xdffe: cmp     ecx, eax
0xe000: mov     rsi, r13
0xe003: cmovle  eax, ecx
0xe006: movsxd  rdx, eax
0xe009: call    0x4bb0
0xe00e: mov     rdi, r13
0xe011: call    0x18d10
0xe016: mov     rsi, qword ptr [rip + 0x1d5ab]
0xe01d: mov     rcx, qword ptr [rip + 0x1d58c]
0xe024: xor     eax, eax
0xe026: mov     edx, dword ptr [rsi + 0xc]
0xe029: test    edx, edx
0xe02b: jg      0xe039
0xe02d: jmp     0xe054
0xe054:
0xe02f: nop
0xe030: add     rax, 1
0xe034: cmp     dword ptr [rsi + 0xc], eax
0xe037: jle     0xe054
0xe039: movzx   edx, byte ptr [rcx + rax]
0xe03d: cmp     dl, 0xd
0xe040: je      0xe047
0xe042: cmp     dl, 0xa
0xe045: jne     0xe030
0xe047: mov     byte ptr [rcx + rax], 0x20
0xe04b: add     rax, 1
0xe04f: cmp     dword ptr [rsi + 0xc], eax
0xe052: jg      0xe039
0xe054: movzx   eax, byte ptr [rip + 0x1d55e]
0xe05b: cmp     al, 1
0xe05d: je      0xe330
0xe063: cmp     al, 2
0xe065: jne     0xe100
0xe06b: mov     edx, dword ptr [rip + 0x1c20b]
0xe071: mov     esi, dword ptr [rip + 0x1c201]
0xe077: mov     edi, dword ptr [rip + 0x1c203]
0xe07d: call    0xa570
0xe082: mov     rdi, qword ptr [rip + 0x1d53f]
0xe089: xor     edx, edx
0xe08b: xor     esi, esi
0xe08d: call    0xc680
0xe092: mov     rdi, qword ptr [rip + 0x1d52f]
0xe099: mov     rsi, qword ptr [rip + 0x1d510]

```

```

0xe0a0: mov     edx, dword ptr [rdi + 0xc]
0xe0a3: call    0xc600
0xe0a8: mov     rdi, qword ptr [rbp - 0x80]
0xe0ac: call    0xa6b0
0xe0b1: mov     rax, qword ptr [rbp - 0x38]
0xe0b5: sub     rax, qword ptr fs:[0x28]
0xe0be: jne     0xe479
0xe0c4: lea     rsp, [rbp - 0x28]
0xe0c8: pop     rbx
0xe0c9: pop     r12
0xe0cb: pop     r13
0xe0cd: pop     r14
0xe0cf: pop     r15
0xe0d1: pop     rbp
0xe0d2: ret
0xe0d3: nop     dword ptr [rax + rax]
0xe0d8: mov     edi, dword ptr [rbp - 0x74]
0xe0db: mov     byte ptr [r13 + r12], 0x5c
0xe0e1: add     ebx, 1
0xe0e4: movzx   edx, byte ptr [rdx]
0xe0e7: lea     eax, [rdi + 1]
0xe0ea: add     edi, 2
0xe0ed: cdqe
0xe0ef: mov     dword ptr [rbp - 0x74], edi
0xe0f2: mov     byte ptr [r13 + rax], dl
0xe0f7: jmp     0xdf3d
0xdf3d:
0xe0fc: nop     dword ptr [rax]
0xe100: mov     edx, dword ptr [rip + 0x1c15e]
0xe106: mov     esi, dword ptr [rip + 0x1c154]
0xe10c: mov     edi, dword ptr [rip + 0x1c156]
0xe112: call    0xa570
0xe117: jmp     0xe082
0xe082:
0xe11c: nop     dword ptr [rax]
0xe120: sub     rax, rdx
0xe123: mov     rsi, r13
0xe126: shr     rax, 1
0xe129: add     rdi, rax
0xe12c: call    0x4bb0
0xe131: jmp     0xe00e
0xe00e:
0xe136: nop     word ptr cs:[rax + rax]
0xe140: lea     rax, [rip + 0x1d2d9]
0xe147: lea     rcx, [rip + 0x1d2d2]
0xe14e: movzx   eax, byte ptr [rax]
0xe151: test    al, al
0xe153: je      0xe1d2

```

```

0xe155: mov     rdi, rcx
0xe158: mov     qword ptr [rbp - 0x90], rcx
0xe15f: call    0x49b0
0xe164: mov     rdi, r13
0xe167: add     r14, rax
0xe16a: mov     qword ptr [rbp - 0x88], rax
0xe171: mov     rsi, r14
0xe174: call    0x1d690
0xe179: mov     rdx, qword ptr [rbp - 0x88]
0xe180: mov     rcx, qword ptr [rbp - 0x90]
0xe187: add     r12, rax
0xe18a: mov     r13, rax
0xe18d: cmp     edx, 8
0xe190: jae     0xe360
0xe196: test    dl, 4
0xe199: jne     0xe438
0xe19f: test    edx, edx
0xe1a1: jne     0xe408
0xe1a7: add     dword ptr [rbp - 0x74], edx
0xe1aa: mov     rcx, qword ptr [rip + 0x1d40f]
0xe1b1: add     ebx, 1
0xe1b4: jmp     0xdf3d
0xdf3d
0xe1b9: nop     dword ptr [rax]
0xe1c0: movzx   eax, byte ptr [rip + 0x1d35d]
0xe1c7: lea     rcx, [rip + 0x1d356]
0xe1ce: test    al, al
0xe1d0: jne     0xe155
0xe1d2: add     r14, 6
0xe1d6: mov     rdi, r13
0xe1d9: add     ebx, 1
0xe1dc: mov     rsi, r14
0xe1df: call    0x1d690
0xe1e4: mov     edi, 0x2965
0xe1e9: mov     edx, 6
0xe1ee: add     dword ptr [rbp - 0x74], edx
0xe1f1: add     r12, rax
0xe1f4: mov     rcx, qword ptr [rip + 0x1d3c5]
0xe1fb: mov     r13, rax
0xe1fe: mov     dword ptr [r12], 0x6e6f6e28
0xe206: mov     word ptr [r12 + 4], di
0xe20c: jmp     0xdf3d
0xdf3d
0xe211: nop     dword ptr [rax]
0xe218: xor     eax, eax
0xe21a: call    0x18330
0xe21f: lea     r10, [rbp - 0x60]
0xe223: mov     edi, dword ptr [rax + 0x14]

```

```

0xe226: movsxd  rsi, dword ptr [rax + 0x10]
0xe22a: movsxd  rcx, dword ptr [rax + 0x18]
0xe22e: lea     edx, [rdi + 0x76c]
0xe234: mov     rdi, r10
0xe237: push    rdx
0xe238: mov     eax, dword ptr [rax + 0xc]
0xe23b: mov     edx, 0x20
0xe240: push    rax
0xe241: lea     rax, [rip + 0x1c4b8]
0xe248: mov     r9, qword ptr [rax + rsi*8]
0xe24c: lea     rax, [rip + 0x1c50d]
0xe253: mov     esi, 1
0xe258: mov     r8, qword ptr [rax + rcx*8]
0xe25c: lea     rcx, [rip + 0x12d65]
0xe263: xor     eax, eax
0xe265: mov     qword ptr [rbp - 0x88], r10
0xe26c: call    0x4f10
0xe271: mov     r10, qword ptr [rbp - 0x88]
0xe278: mov     rdi, r10
0xe27b: mov     qword ptr [rbp - 0x90], r10
0xe282: call    0x49b0
0xe287: mov     rdi, r13
0xe28a: add     r14, rax
0xe28d: mov     qword ptr [rbp - 0x88], rax
0xe294: mov     rsi, r14
0xe297: call    0x1d690
0xe29c: mov     rdx, qword ptr [rbp - 0x88]
0xe2a3: mov     r10, qword ptr [rbp - 0x90]
0xe2aa: add     r12, rax
0xe2ad: mov     r13, rax
0xe2b0: cmp     edx, 8
0xe2b3: jae     0xe3a8
0xe2b9: test    dl, 4
0xe2bc: jne     0xe44e
0xe2c2: test    edx, edx
0xe2c4: je      0xe2d7
0xe2c6: movzx   eax, byte ptr [rbp - 0x60]
0xe2ca: mov     byte ptr [r12], al
0xe2ce: test    dl, 2
0xe2d1: jne     0xe466
0xe2d7: pop     rcx
0xe2d8: add     dword ptr [rbp - 0x74], edx
0xe2db: add     ebx, 1
0xe2de: mov     rcx, qword ptr [rip + 0x1d2db]
0xe2e5: pop     rsi
0xe2e6: jmp     0xdf3d
0xdf3d:
0xe2eb: nop     dword ptr [rax + rax]

```



```

0xe2f0: add     dword ptr [rbp - 0x74], 1
0xe2f4: add     ebx, 1
0xe2f7: mov     byte ptr [r13 + r12], 0x5c
0xe2fd: jmp     0xdf3d
0xdf3d:
0xe302: nop     word ptr [rax + rax]
0xe308: movzx   eax, byte ptr [rip + 0x1d152]
0xe30f: lea     rcx, [rip + 0x1d14b]
0xe316: jmp     0xe151
0xe151:
0xe31b: nop     dword ptr [rax + rax]
0xe320: add     dword ptr [rbp - 0x74], 1
0xe324: mov     byte ptr [r13 + r12], 0x5c
0xe32a: jmp     0xdf3d
0xdf3d:
0xe32f: nop
0xe330: mov     edx, dword ptr [rip + 0x1bf3a]
0xe336: mov     esi, dword ptr [rip + 0x1bf30]
0xe33c: mov     edi, dword ptr [rip + 0x1bf32]
0xe342: call    0xa570
0xe347: jmp     0xe082
0xe082:
0xe34c: nop     dword ptr [rax]
0xe350: mov     eax, dword ptr [rax + 0xc]
0xe353: jmp     0xdffe
0xdffe:
0xe358: nop     dword ptr [rax + rax]
0xe360: mov     rax, qword ptr [rcx]
0xe363: lea     rdi, [r12 + 8]
0xe368: add     ebx, 1
0xe36b: and     rdi, 0xfffffffffffffff8
0xe36f: mov     qword ptr [r12], rax
0xe373: mov     eax, edx
0xe375: mov     rsi, qword ptr [rcx + rax - 8]
0xe37a: mov     qword ptr [r12 + rax - 8], rsi
0xe37f: sub     r12, rdi
0xe382: mov     rsi, rcx
0xe385: sub     rsi, r12
0xe388: add     r12d, edx
0xe38b: shr     r12d, 3
0xe38f: mov     ecx, r12d
0xe392: rep movsq      qword ptr [rdi], qword ptr [rsi]
0xe395: add     dword ptr [rbp - 0x74], edx
0xe398: mov     rcx, qword ptr [rip + 0x1d221]
0xe39f: jmp     0xdf3d
0xdf3d:
0xe3a4: nop     dword ptr [rax]
0xe3a8: mov     rax, qword ptr [rbp - 0x60]

```

```

0xe3ac: lea      rsi, [r12 + 8]
0xe3b1: and      rsi, 0xfffffffffffffff8
0xe3b5: mov      qword ptr [r12], rax
0xe3b9: mov      eax, edx
0xe3bb: mov      rcx, qword ptr [r10 + rax - 8]
0xe3c0: mov      qword ptr [r12 + rax - 8], rcx
0xe3c5: sub      r12, rsi
0xe3c8: sub      r10, r12
0xe3cb: add      r12d, edx
0xe3ce: and      r12d, 0xffffffff8
0xe3d2: cmp      r12d, 8
0xe3d6: jb       0xe2d7
0xe3dc: mov      r11d, dword ptr [rbp - 0x74]
0xe3e0: and      r12d, 0xffffffff8
0xe3e4: xor      eax, eax
0xe3e6: mov      ecx, eax
0xe3e8: add      eax, 8
0xe3eb: mov      rdi, qword ptr [r10 + rcx]
0xe3ef: mov      qword ptr [rsi + rcx], rdi
0xe3f3: cmp      eax, r12d
0xe3f6: jb       0xe3e6
0xe3f8: mov      dword ptr [rbp - 0x74], r11d
0xe3fc: jmp      0xe2d7
0xe2d7:
0xe401: nop      dword ptr [rax]
0xe408: movzx    eax, byte ptr [rcx]
0xe40b: mov      byte ptr [r12], al
0xe40f: test     dl, 2
0xe412: je       0xe1a7
0xe418: mov      eax, edx
0xe41a: movzx    ecx, word ptr [rcx + rax - 2]
0xe41f: mov      word ptr [r12 + rax - 2], cx
0xe425: jmp      0xe1a7
0xe1a7:
0xe42a: nop      word ptr [rax + rax]
0xe430: mov      rax, r13
0xe433: jmp      0xdfc7
0xdfc7:
0xe438: mov      eax, dword ptr [rcx]
0xe43a: mov      dword ptr [r12], eax
0xe43e: mov      eax, edx
0xe440: mov      ecx, dword ptr [rcx + rax - 4]
0xe444: mov      dword ptr [r12 + rax - 4], ecx
0xe449: jmp      0xe1a7
0xe1a7:
0xe44e: mov      eax, dword ptr [rbp - 0x60]
0xe451: mov      dword ptr [r12], eax
0xe455: mov      eax, edx

```

```

0xe457: mov     ecx, dword ptr [r10 + rax - 4]
0xe45c: mov     dword ptr [r12 + rax - 4], ecx
0xe461: jmp     0xe2d7
0xe2d7:
0xe466: mov     eax, edx
0xe468: movzx   ecx, word ptr [r10 + rax - 2]
0xe46e: mov     word ptr [r12 + rax - 2], cx
0xe474: jmp     0xe2d7
0xe2d7:
0xe479: call    0x49d0
0xe47e: nop
0xe480: endbr64
0xe484: push    r12
0xe486: mov     r12d, esi
0xe489: push    rbp
0xe48a: mov     rbp, rdi
0xe48d: mov     rdi, qword ptr [rip + 0x1d12c]
0xe494: push    rbx
0xe495: mov     ebx, edx
0xe497: test    rdi, rdi
0xe49a: je      0xe4a1
0xe49c: call    0x18d10
0xe4a1: mov     rdi, rbp
0xe4a4: call    0x1d7a0
0xe4a9: mov     rdi, rax
0xe4ac: mov     rbp, rax
0xe4af: mov     qword ptr [rip + 0x1d10a], rax
0xe4b6: call    0x49b0
0xe4bb: mov     rdi, rbp
0xe4be: mov     rsi, rax
0xe4c1: call    0x184d0
0xe4c6: mov     byte ptr [rip + 0x1d0ec], r12b
0xe4cd: xor     eax, eax
0xe4cf: mov     byte ptr [rip + 0x1d0e3], bl
0xe4d5: pop     rbx
0xe4d6: pop     rbp
0xe4d7: pop     r12
0xe4d9: jmp     0xdea0
0xdea0:
0xe4de: nop
0xe4e0: endbr64
0xe4e4: push    rbp
0xe4e5: mov     rdi, qword ptr [rip + 0x1d0bc]
0xe4ec: call    0x1d7a0
0xe4f1: mov     rdi, qword ptr [rip + 0x1d0c8]
0xe4f8: mov     rbp, rax
0xe4fb: test    rdi, rdi
0xe4fe: je      0xe505

```

```

0xe500: call    0x18d10
0xe505: mov     rdi, rbp
0xe508: call    0x1d7a0
0xe50d: mov     rdi, rax
0xe510: mov     rbp, rax
0xe513: mov     qword ptr [rip + 0x1d0a6], rax
0xe51a: call    0x49b0
0xe51f: mov     rdi, rbp
0xe522: mov     rsi, rax
0xe525: call    0x184d0
0xe52a: xor     eax, eax
0xe52c: pop     rbp
0xe52d: mov     byte ptr [rip + 0x1d085], 0
0xe534: mov     byte ptr [rip + 0x1d07d], 0
0xe53b: jmp     0xdea0
0xdea0
0xe540: endbr64
0xe544: mov     rax, qword ptr [rsi]
0xe547: mov     rdx, qword ptr [rdi]
0xe54a: mov     rax, qword ptr [rax + 8]
0xe54e: sub     eax, dword ptr [rdx + 8]
0xe551: ret
0xe552: nop     word ptr cs:[rax + rax]
0xe55d: nop     dword ptr [rax]
0xe560: push    rbp
0xe561: mov     rbp, rdi
0xe564: mov     rdi, rdx
0xe567: push    rbx
0xe568: mov     ebx, esi
0xe56a: lea     rsi, [rip + 0x1bdff]
0xe571: sub     rsp, 8
0xe575: call    0x48b0
0xe57a: movsxd  rsi, ebx
0xe57d: mov     rcx, rax
0xe580: lea     rsi, [rsi + rsi*4]
0xe584: mov     rax, qword ptr [rbp + 0x38]
0xe588: shl     rsi, 4
0xe58c: mov     edx, dword ptr [rax + rsi + 0x10]
0xe590: mov     eax, 0x2d
0xe595: mov     edi, edx
0xe597: and     edi, 0xf000
0xe59d: cmp     edi, 0x8000
0xe5a3: je      0xe600
0xe5a5: mov     eax, 0x64
0xe5aa: cmp     edi, 0x4000
0xe5b0: je      0xe600
0xe5b2: mov     eax, 0x63
0xe5b7: cmp     edi, 0x2000

```

```

0xe5bd: je      0xe600
0xe5bf: mov     eax, 0x62
0xe5c4: cmp     edi, 0x6000
0xe5ca: je      0xe600
0xe5cc: mov     eax, 0x70
0xe5d1: cmp     edi, 0x1000
0xe5d7: je      0xe600
0xe5d9: mov     eax, 0x73
0xe5de: cmp     edi, 0xc000
0xe5e4: je      0xe600
0xe5e6: cmp     edi, 0xa000
0xe5ec: mov     eax, 0x6c
0xe5f1: mov     edi, 0x3f
0xe5f6: cmovne  eax, edi
0xe5f9: nop     dword ptr [rax]
0xe600: mov     byte ptr [rcx], al
0xe602: lea     rax, [rcx + 9]
0xe606: nop     word ptr cs:[rax + rax]
0xe610: test    dl, 1
0xe613: jne     0xe618
0xe615: mov     byte ptr [rax], 0x2d
0xe618: sub     rax, 1
0xe61c: shr     edx, 1
0xe61e: cmp     rax, rcx
0xe621: jne     0xe610
0xe623: mov     rax, qword ptr [rbp + 0x38]
0xe627: mov     eax, dword ptr [rax + rsi + 0x10]
0xe62b: test    ah, 8
0xe62e: je      0xe644
0xe630: cmp     byte ptr [rcx + 3], 0x78
0xe634: mov     edx, 0x73
0xe639: mov     esi, 0x53
0xe63e: cmovne  edx, esi
0xe641: mov     byte ptr [rcx + 3], dl
0xe644: test    ah, 4
0xe647: je      0xe65d
0xe649: cmp     byte ptr [rcx + 6], 0x78
0xe64d: mov     edx, 0x73
0xe652: mov     esi, 0x53
0xe657: cmovne  edx, esi
0xe65a: mov     byte ptr [rcx + 6], dl
0xe65d: test    ah, 2
0xe660: je      0xe676
0xe662: cmp     byte ptr [rcx + 9], 0x78
0xe666: mov     eax, 0x74
0xe66b: mov     edx, 0x54
0xe670: cmovne  eax, edx
0xe673: mov     byte ptr [rcx + 9], al

```

```

0xe676: add    rsp, 8
0xe67a: pop     rbx
0xe67b: pop     rbp
0xe67c: ret
0xe67d: nop     dword ptr [rax]
0xe680: endbr64
0xe684: push    r13
0xe686: push    r12
0xe688: push    rbp
0xe689: push    rbx
0xe68a: sub     rsp, 8
0xe68e: movzx   eax, byte ptr [rdi + 0x48]
0xe692: test    al, al
0xe694: sete    cl
0xe697: cmp     byte ptr [rsi + 0x48], 0
0xe69b: sete    dl
0xe69e: cmp     cl, dl
0xe6a0: je      0xe6c0
0xe6a2: cmp     al, 1
0xe6a4: sbb     r12d, r12d
0xe6a7: or      r12d, 1
0xe6ab: add     rsp, 8
0xe6af: mov     eax, r12d
0xe6b2: pop     rbx
0xe6b3: pop     rbp
0xe6b4: pop     r12
0xe6b6: pop     r13
0xe6b8: ret
0xe6b9: nop     dword ptr [rax]
0xe6c0: mov     r12d, dword ptr [rip + 0x1cf61]
0xe6c7: mov     rbx, rdi
0xe6ca: mov     rbp, rsi
0xe6cd: cmp     r12d, 8
0xe6d1: ja      0xe6e7
0xe6d3: lea     rdx, [rip + 0x131c6]
0xe6da: mov     eax, r12d
0xe6dd: movsxd  rax, dword ptr [rdx + rax*4]
0xe6e1: add     rax, rdx
0xe6e4: jmp     rax
rax
                                ->->->                                indirect JMP
0x99e0: push    r13
0x99e2: push    r12
0x99e4: mov     r12, qword ptr [rip + 0x20f8d]
0x99eb: push    rbp
0x99ec: test    r12, r12
0x99ef: je      0x9a58
0x99f1: mov     r13, rdi

```

```

0x99f4: jmp      0x9a13
0x9a13:
0x99f6: nop      word ptr cs:[rax + rax]
0x9a00: js       0x9a30
0x9a02: mov      r12, qword ptr [r12 + 8]
0x9a07: mov      qword ptr [rip + 0x20f6a], r12
0x9a0e: test     r12, r12
0x9a11: je       0x9a58
0x9a13: mov      rbp, qword ptr [r12]
0x9a17: mov      rdi, r13
0x9a1a: mov      rsi, rbp
0x9a1d: call     0x4b30
0x9a22: test     eax, eax
0x9a24: jne      0x9a00
0x9a26: mov      rax, r12
0x9a29: pop      rbp
0x9a2a: pop      r12
0x9a2c: pop      r13
0x9a2e: ret
0x9a2f: nop
0x9a30: mov      rdi, r13
0x9a33: call     0x49b0
0x9a38: mov      rsi, rbp
0x9a3b: mov      rdi, r13
0x9a3e: mov      rdx, rax
0x9a41: call     0x4890
0x9a46: pop      rbp
0x9a47: neg      eax
0x9a49: sbb      r12, r12
0x9a4c: mov      rax, r12
0x9a4f: pop      r12
0x9a51: pop      r13
0x9a53: ret
0x9a54: nop      dword ptr [rax]
0x9a58: mov      r12, -1
0x9a5f: pop      rbp
0x9a60: mov      rax, r12
0x9a63: pop      r12
0x9a65: pop      r13
0x9a67: ret
0x9a68: nop      dword ptr [rax + rax]
0x9a70: endbr64
0x9a74: movsxd   rdx, dword ptr [rip + 0x2148d]
0x9a7b: push     r12
0x9a7d: push     rbp
0x9a7e: mov      ebp, edi
0x9a80: push     rbx
0x9a81: cmp      edx, 0x400

```

```

0x9a87: je      0x9ab0
0x9a89: lea     eax, [rdx + 1]
0x9a8c: lea     r12, [rip + 0x2148d]
0x9a93: mov     byte ptr [r12 + rdx], bpl
0x9a97: pop     rbx
0x9a98: mov     dword ptr [rip + 0x2146a], eax
0x9a9e: pop     rbp
0x9a9f: mov     eax, 1
0x9aa4: pop     r12
0x9aa6: ret
0x9aa7: nop     word ptr [rax + rax]
0x9ab0: xor     ebx, ebx
0x9ab2: lea     r12, [rip + 0x21467]
0x9ab9: jmp     0x9acc
0x9acc:
0x9abb: nop     dword ptr [rax + rax]
0x9ac0: mov     edx, dword ptr [rip + 0x21442]
0x9ac6: add     ebx, eax
0x9ac8: cmp     edx, ebx
0x9aca: jle     0x9ae5
0x9acc: sub     edx, ebx
0x9ace: movsxd  rsi, ebx
0x9ad1: mov     edi, 1
0x9ad6: add     rsi, r12
0x9ad9: movsxd  rdx, edx
0x9adc: call    0x18da0
0x9ae1: test    eax, eax
0x9ae3: jns     0x9ac0
0x9ae5: mov     eax, 1
0x9aea: xor     edx, edx
0x9aec: jmp     0x9a93
0x9a93:
0x9aee: nop
0x9af0: endbr64
0x9af4: mov     dword ptr [rip + 0x21862], edi
0x9afa: ret
0x9afb: nop     dword ptr [rax + rax]
0x9b00: endbr64
0x9b04: push    r12
0x9b06: push    rbp
0x9b07: push    rbx
0x9b08: mov     ebx, edi
0x9b0a: sub     rsp, 0x50
0x9b0e: mov     rax, qword ptr fs:[0x28]
0x9b17: mov     qword ptr [rsp + 0x48], rax
0x9b1c: xor     eax, eax
0x9b1e: cmp     edi, 1
0x9b21: je      0x9b60

```



```

0x9b23: lea      rdx, [rip + 0x21896]
0x9b2a: mov      esi, 1
0x9b2f: mov      edi, 1
0x9b34: call     0x4d70
0x9b39: mov      dword ptr [rip + 0x20e19], ebx
0x9b3f: mov      rax, qword ptr [rsp + 0x48]
0x9b44: sub      rax, qword ptr fs:[0x28]
0x9b4d: jne      0x9c7e
0x9b53: add      rsp, 0x50
0x9b57: pop      rbx
0x9b58: pop      rbp
0x9b59: pop      r12
0x9b5b: ret
0x9b5c: nop      dword ptr [rax]
0x9b60: mov      rax, qword ptr [rip + 0x21889]
0x9b67: mov      rdx, qword ptr [rip + 0x21852]
0x9b6e: lea      r12, [rip + 0x2180b]
0x9b75: mov      esi, 1
0x9b7a: movdqa   xmm0, xmmword ptr [rip + 0x2183e]
0x9b82: movdqa   xmm1, xmmword ptr [rip + 0x21846]
0x9b8a: mov      edi, 1
0x9b8f: mov      qword ptr [rip + 0x2181a], rax
0x9b96: mov      eax, dword ptr [rip + 0x2185c]
0x9b9c: movaps   xmmword ptr [rip + 0x217dd], xmm0
0x9ba3: movdqa   xmm2, xmmword ptr [rip + 0x21835]
0x9bab: mov      dword ptr [rip + 0x21807], eax
0x9bb1: movabs   rax, 0xffffffffffffffa3c
0x9bbb: and      rax, rdx
0x9bbe: mov      rdx, r12
0x9bc1: movaps   xmmword ptr [rip + 0x217c8], xmm1
0x9bc8: mov      qword ptr [rip + 0x217b1], rax
0x9bcf: mov      eax, dword ptr [rip + 0x217b7]
0x9bd5: movaps   xmmword ptr [rip + 0x217c4], xmm2
0x9bdc: and      eax, 0xffffffff5
0x9bdf: mov      byte ptr [rip + 0x217bb], 0
0x9be6: mov      dword ptr [rip + 0x217a1], 7
0x9bf0: or       al, 0x81
0x9bf2: mov      dword ptr [rip + 0x21794], eax
0x9bf8: mov      eax, 0x100
0x9bfd: mov      word ptr [rip + 0x21792], ax
0x9c04: mov      qword ptr [rip + 0x2178a], 0x1a0000
0x9c0f: call     0x4d70
0x9c14: mov      rdi, r12
0x9c17: mov      r12, rsp
0x9c1a: call     0x4d10
0x9c1f: mov      edx, 1
0x9c24: mov      esi, 0x540a
0x9c29: mov      edi, 1

```

```

0x9c2e: mov     dword ptr [rip + 0x20cfc], eax
0x9c34: xor      eax, eax
0x9c36: call     0x4a70
0x9c3b: mov     esi, 1
0x9c40: mov     edi, 1
0x9c45: call     0x4db0
0x9c4a: mov     ebp, dword ptr [rip + 0x200d4]
0x9c50: mov     rsi, r12
0x9c53: mov     edi, 1
0x9c58: call     0x4d60
0x9c5d: mov     rdx, r12
0x9c60: mov     esi, 1
0x9c65: mov     edi, 1
0x9c6a: mov     byte ptr [rsp + 0x11], bpl
0x9c6f: mov     byte ptr [rsp + 0x12], 0
0x9c74: call     0x4d70
0x9c79: jmp      0x9b39
0x9b39:
0x9c7e: call     0x49d0
0x9c83: nop      word ptr cs:[rax + rax]
0x9c8e: nop
0x9c90: endbr64
0x9c94: mov     eax, dword ptr [rip + 0x20cbe]
0x9c9a: ret
0x9c9b: nop      dword ptr [rax + rax]
0x9ca0: endbr64
0x9ca4: push    r12
0x9ca6: push    rbx
0x9ca7: mov     ebx, edi
0x9ca9: mov     edi, 1
0x9cae: sub     rsp, 0x58
0x9cb2: mov     rax, qword ptr fs:[0x28]
0x9cbb: mov     qword ptr [rsp + 0x48], rax
0x9cc0: xor     eax, eax
0x9cc2: mov     r12, rsp
0x9cc5: mov     rsi, r12
0x9cc8: call     0x4d60
0x9ccd: mov     rdx, r12
0x9cd0: mov     esi, 1
0x9cd5: mov     edi, 1
0x9cda: mov     byte ptr [rsp + 0x11], bl
0x9cde: mov     byte ptr [rsp + 0x12], 0
0x9ce3: call     0x4d70
0x9ce8: mov     dword ptr [rip + 0x20036], ebx
0x9cee: mov     rax, qword ptr [rsp + 0x48]
0x9cf3: sub     rax, qword ptr fs:[0x28]
0x9cfc: jne     0x9d06
0x9cfe: add     rsp, 0x58

```

```

0x9d02: pop      rbx
0x9d03: pop      r12
0x9d05: ret
0x9d06: call     0x49d0
0x9d0b: nop      dword ptr [rax + rax]
0x9d10: endbr64
0x9d14: mov      edx, dword ptr [rip + 0x211ee]
0x9d1a: test     edx, edx
0x9d1c: jle      0x9d66
0x9d1e: push     rbp
0x9d1f: xor      ebp, ebp
0x9d21: push     rbx
0x9d22: lea      rbx, [rip + 0x211f7]
0x9d29: sub      rsp, 8
0x9d2d: jmp      0x9d3c
0x9d3c:
0x9d2f: nop
0x9d30: mov      edx, dword ptr [rip + 0x211d2]
0x9d36: add      ebp, eax
0x9d38: cmp      edx, ebp
0x9d3a: jle      0x9d55
0x9d3c: sub      edx, ebp
0x9d3e: movsxd   rsi, ebp
0x9d41: mov      edi, 1
0x9d46: add      rsi, rbx
0x9d49: movsxd   rdx, edx
0x9d4c: call     0x18da0
0x9d51: test     eax, eax
0x9d53: jns      0x9d30
0x9d55: mov      dword ptr [rip + 0x211a9], 0
0x9d5f: add      rsp, 8
0x9d63: pop      rbx
0x9d64: pop      rbp
0x9d65: ret
0x9d66: mov      dword ptr [rip + 0x21198], 0
0x9d70: ret
0x9d71: nop      word ptr cs:[rax + rax]
0x9d7c: nop      dword ptr [rax]
0x9d80: endbr64
0x9d84: push     rbp
0x9d85: lea      rdx, [rip - 0x31c]
0x9d8c: push     rbx
0x9d8d: sub      rsp, 8
0x9d91: mov      esi, dword ptr [rip + 0x20c05]
0x9d97: mov      rdi, qword ptr [rip + 0x2004a]
0x9d9e: call     0x4820
0x9da3: mov      edx, dword ptr [rip + 0x2115f]
0x9da9: test     edx, edx

```

```

0x9dab: jle      0x9de5
0x9dad: xor      ebp, ebp
0x9daf: lea      rbx, [rip + 0x2116a]
0x9db6: jmp      0x9dcc
0x9dcc:
0x9db8: nop      dword ptr [rax + rax]
0x9dc0: mov      edx, dword ptr [rip + 0x21142]
0x9dc6: add      ebp, eax
0x9dc8: cmp      edx, ebp
0x9dca: jle      0x9de5
0x9dcc: sub      edx, ebp
0x9dce: movsxd   rsi, ebp
0x9dd1: mov      edi, 1
0x9dd6: add      rsi, rbx
0x9dd9: movsxd   rdx, edx
0x9ddc: call     0x18da0
0x9de1: test     eax, eax
0x9de3: jns      0x9dc0
0x9de5: mov      dword ptr [rip + 0x21119], 0
0x9def: add      rsp, 8
0x9df3: pop      rbx
0x9df4: pop      rbp
0x9df5: ret
0x9df6: nop      word ptr cs:[rax + rax]
0x9e00: endbr64
0x9e04: push     rbp
0x9e05: lea      rdx, [rip - 0x39c]
0x9e0c: push     rbx
0x9e0d: sub      rsp, 8
0x9e11: mov      eax, dword ptr [rip + 0x20b85]
0x9e17: mov      rdi, qword ptr [rip + 0x2010a]
0x9e1e: lea      esi, [rax - 1]
0x9e21: call     0x4820
0x9e26: mov      edx, dword ptr [rip + 0x210dc]
0x9e2c: test     edx, edx
0x9e2e: jle      0x9e65
0x9e30: xor      ebp, ebp
0x9e32: lea      rbx, [rip + 0x210e7]
0x9e39: jmp      0x9e4c
0x9e4c:
0x9e3b: nop      dword ptr [rax + rax]
0x9e40: mov      edx, dword ptr [rip + 0x210c2]
0x9e46: add      ebp, eax
0x9e48: cmp      edx, ebp
0x9e4a: jle      0x9e65
0x9e4c: sub      edx, ebp
0x9e4e: movsxd   rsi, ebp
0x9e51: mov      edi, 1

```

```

0x9e56: add    rsi, rbx
0x9e59: movsxd  rdx, edx
0x9e5c: call    0x18da0
0x9e61: test    eax, eax
0x9e63: jns     0x9e40
0x9e65: mov     dword ptr [rip + 0x21099], 0
0x9e6f: add     rsp, 8
0x9e73: pop     rbx
0x9e74: pop     rbp
0x9e75: ret
0x9e76: nop     word ptr cs:[rax + rax]
0x9e80: endbr64
0x9e84: push    rbp
0x9e85: lea     rdx, [rip - 0x41c]
0x9e8c: push    rbx
0x9e8d: sub     rsp, 8
0x9e91: mov     eax, dword ptr [rip + 0x20b05]
0x9e97: mov     rdi, qword ptr [rip + 0x200aa]
0x9e9e: lea     esi, [rax - 1]
0x9ea1: call    0x4820
0x9ea6: mov     edx, dword ptr [rip + 0x2105c]
0x9eac: test    edx, edx
0x9eae: jle     0x9ee5
0x9eb0: xor     ebp, ebp
0x9eb2: lea     rbx, [rip + 0x21067]
0x9eb9: jmp     0x9ecc
0x9ecc:
0x9ebb: nop     dword ptr [rax + rax]
0x9ec0: mov     edx, dword ptr [rip + 0x21042]
0x9ec6: add     ebp, eax
0x9ec8: cmp     edx, ebp
0x9eca: jle     0x9ee5
0x9ecc: sub     edx, ebp
0x9ece: movsxd  rsi, ebp
0x9ed1: mov     edi, 1
0x9ed6: add     rsi, rbx
0x9ed9: movsxd  rdx, edx
0x9edc: call    0x18da0
0x9ee1: test    eax, eax
0x9ee3: jns     0x9ec0
0x9ee5: mov     dword ptr [rip + 0x21019], 0
0x9eef: add     rsp, 8
0x9ef3: pop     rbx
0x9ef4: pop     rbp
0x9ef5: ret
0x9ef6: nop     word ptr cs:[rax + rax]
0x9f00: endbr64
0x9f04: mov     edx, dword ptr [rip + 0x214f2]

```

```

0x9f0a: movzx    eax, byte ptr [rdi]
0x9f0d: push     r14
0x9f0f: push     r13
0x9f11: mov      r13, rdi
0x9f14: push     r12
0x9f16: push     rbp
0x9f17: push     rbx
0x9f18: test     edx, edx
0x9f1a: jne      0x9fc8
0x9f20: cmp      al, 0x5e
0x9f22: jne      0x9fb8
0x9f28: lea      rbx, [r13 + 1]
0x9f2c: mov      rbp, r13
0x9f2f: lea      r14, [rip + 0x1fd6a]
0x9f36: jmp      0x9f56
0x9f56:
0x9f38: nop      dword ptr [rax + rax]
0x9f40: add      rbp, 1
0x9f44: mov      byte ptr [rbx - 1], al
0x9f47: mov      rdx, rbx
0x9f4a: add      rbx, 1
0x9f4e: movzx    eax, byte ptr [rbp]
0x9f52: test     al, al
0x9f54: je       0x9fa6
0x9f56: cmp      al, 0x5e
0x9f58: jne      0x9f40
0x9f5a: movzx    r12d, byte ptr [rbp + 1]
0x9f5f: test     r12b, r12b
0x9f62: je       0x9fb8
0x9f64: call     0x47c0
0x9f69: mov      r8, rax
0x9f6c: movzx    eax, r12b
0x9f70: mov      rdx, qword ptr [r8]
0x9f73: mov      eax, dword ptr [rdx + rax*4]
0x9f76: cmp      eax, 0x47
0x9f79: je       0x9fb8
0x9f7b: cmp      eax, 0x5a
0x9f7e: je       0x9fb8
0x9f80: and      r12d, 0x7f
0x9f84: add      rbp, 2
0x9f88: mov      rdx, rbx
0x9f8b: add      rbx, 1
0x9f8f: sub      r12d, 0x20
0x9f93: movsxd   r12, r12d
0x9f96: movzx    eax, byte ptr [r14 + r12]
0x9f9b: mov      byte ptr [rbx - 2], al
0x9f9e: movzx    eax, byte ptr [rbp]
0x9fa2: test     al, al

```

```

0x9fa4: jne      0x9f56
0x9fa6: mov      byte ptr [rdx], 0
0x9fa9: mov      rax, r13
0x9fac: pop      rbx
0x9fad: pop      rbp
0x9fae: pop      r12
0x9fb0: pop      r13
0x9fb2: pop      r14
0x9fb4: ret
0x9fb5: nop      dword ptr [rax]
0x9fb8: pop      rbx
0x9fb9: xor      eax, eax
0x9fbb: pop      rbp
0x9fbc: pop      r12
0x9fbe: pop      r13
0x9fc0: pop      r14
0x9fc2: ret
0x9fc3: nop      dword ptr [rax + rax]
0x9fc8: test     al, al
0x9fca: jne      0x9f28
0x9fd0: mov      rdx, rdi
0x9fd3: jmp      0x9fa6
0x9fa6
0x9fd5: nop      word ptr cs:[rax + rax]
0x9fe0: endbr64
0x9fe4: push     r15
0x9fe6: push     r14
0x9fe8: mov      r14, rdi
0x9feb: push     r13
0x9fed: lea      r13, [rip + 0x16c12]
0x9ff4: push     r12
0x9ff6: lea      r12, [rip + 0x209c3]
0x9ffd: push     rbp
0x9ffe: lea      rbp, [rip + 0x16bef]
0xa005: push     rbx
0xa006: mov      rbx, rdi
0xa009: sub      rsp, 0x18
0xa00d: mov      rax, qword ptr fs:[0x28]
0xa016: mov      qword ptr [rsp + 8], rax
0xa01b: xor      eax, eax
0xa01d: mov      byte ptr [rip + 0x2099c], 0
0xa024: movzx    r15d, byte ptr [rdi]
0xa028: test     r15b, r15b
0xa02b: jne      0xa0c2
0xa031: jmp      0xa0f0
0xa0f0
0xa036: nop      word ptr cs:[rax + rax]
0xa040: cmp      r15b, 0x20

```

```

0xa044: je      0xa148
0xa04a: cmp     r15b, 0x7f
0xa04e: je      0xa160
0xa054: cmp     r15b, 0xff
0xa058: je      0xa180
0xa05e: call    0x4f00
0xa063: mov     r8, rax
0xa066: movzx   eax, r15b
0xa06a: mov     rsi, qword ptr [r8]
0xa06d: test    byte ptr [rsi + rax*2], 2
0xa071: je      0xa120
0xa077: lea     edx, [r15 + 0x40]
0xa07b: lea     rsi, [rsp + 5]
0xa080: mov     rdi, r12
0xa083: mov     byte ptr [rsp + 5], 0x5e
0xa088: mov     byte ptr [rsp + 6], dl
0xa08c: mov     edx, 0x12c
0xa091: mov     byte ptr [rsp + 7], 0
0xa096: call    0x47d0
0xa09b: add     r14, 1
0xa09f: cmp     byte ptr [r14], 0
0xa0a3: je      0xa0f0
0xa0a5: cmp     rbx, r14
0xa0a8: je      0xa0be
0xa0aa: mov     edx, 0x12c
0xa0af: lea     rsi, [rip + 0x1668f]
0xa0b6: mov     rdi, r12
0xa0b9: call    0x47d0
0xa0be: movzx   r15d, byte ptr [r14]
0xa0c2: cmp     r15b, 0x1b
0xa0c6: jne     0xa040
0xa0cc: mov     edx, 0x12c
0xa0d1: mov     rsi, rbp
0xa0d4: mov     rdi, r12
0xa0d7: add     r14, 1
0xa0db: call    0x47d0
0xa0e0: cmp     byte ptr [r14], 0
0xa0e4: jne     0xa0a5
0xa0e6: nop     word ptr cs:[rax + rax]
0xa0f0: mov     rax, qword ptr [rsp + 8]
0xa0f5: sub     rax, qword ptr fs:[0x28]
0xa0fe: jne     0xa199
0xa104: add     rsp, 0x18
0xa108: mov     rax, r12
0xa10b: pop     rbx
0xa10c: pop     rbp
0xa10d: pop     r12
0xa10f: pop     r13

```



```

0xa111: pop      r14
0xa113: pop      r15
0xa115: ret
0xa116: nop      word ptr cs:[rax + rax]
0xa120: lea      rsi, [rsp + 5]
0xa125: mov      edx, 0x12c
0xa12a: mov      rdi, r12
0xa12d: mov      byte ptr [rsp + 5], r15b
0xa132: mov      byte ptr [rsp + 6], 0
0xa137: call     0x47d0
0xa13c: jmp      0xa09b
0xa09b:
0xa141: nop      dword ptr [rax]
0xa148: mov      edx, 0x12c
0xa14d: mov      rsi, r13
0xa150: mov      rdi, r12
0xa153: call     0x47d0
0xa158: jmp      0xa09b
0xa09b:
0xa15d: nop      dword ptr [rax]
0xa160: mov      edx, 0x12c
0xa165: lea      rsi, [rip + 0x16a8f]
0xa16c: mov      rdi, r12
0xa16f: call     0x47d0
0xa174: jmp      0xa09b
0xa09b:
0xa179: nop      dword ptr [rax]
0xa180: mov      edx, 0x12c
0xa185: lea      rsi, [rip + 0x16a79]
0xa18c: mov      rdi, r12
0xa18f: call     0x47d0
0xa194: jmp      0xa09b
0xa09b:
0xa199: call     0x49d0
0xa19e: nop
0xa1a0: endbr64
0xa1a4: mov      eax, dword ptr [rip + 0x211ae]
0xa1aa: mov      r8, rdi
0xa1ad: mov      edi, dword ptr [rip + 0x207e5]
0xa1b3: push     r13
0xa1b5: push     r12
0xa1b7: xor      r12d, r12d
0xa1ba: lea      edx, [rax + rsi]
0xa1bd: push     rbx
0xa1be: mov      dword ptr [rip + 0x21194], edx
0xa1c4: cmp      edi, eax
0xa1c6: jle      0xa21c
0xa1c8: mov      ecx, dword ptr [rip + 0x21186]

```

```

0xa1ce: cmp      ecx, dword ptr [rip + 0x207c8]
0xa1d4: jge      0xa21c
0xa1d6: mov      r12d, edi
0xa1d9: sub      r12d, eax
0xa1dc: cmp      edx, edi
0xa1de: cmovle   r12d, esi
0xa1e2: imul     edi, ecx
0xa1e5: mov      rsi, r8
0xa1e8: movsxd   r13, r12d
0xa1eb: lea      ebx, [rdi + rax]
0xa1ee: mov      rdi, qword ptr [rip + 0x21143]
0xa1f5: mov      rdx, r13
0xa1f8: movsxd   rbx, ebx
0xa1fb: add      rdi, rbx
0xa1fe: call     0x4bb0
0xa203: add      rbx, qword ptr [rip + 0x21126]
0xa20a: movzx    esi, byte ptr [rip + 0x21140]
0xa211: mov      rdx, r13
0xa214: mov      rdi, rbx
0xa217: call     0x4a50
0xa21c: mov      eax, r12d
0xa21f: pop      rbx
0xa220: pop      r12
0xa222: pop      r13
0xa224: ret
0xa225: nop      word ptr cs:[rax + rax]
0xa230: endbr64
0xa234: push     r13
0xa236: push     r12
0xa238: xor      r12d, r12d
0xa23b: push     rbx
0xa23c: sub      rsp, 0x10
0xa240: mov      rax, qword ptr fs:[0x28]
0xa249: mov      qword ptr [rsp + 8], rax
0xa24e: xor      eax, eax
0xa250: mov      eax, dword ptr [rip + 0x21102]
0xa256: mov      byte ptr [rsp + 7], dil
0xa25b: mov      edi, dword ptr [rip + 0x20737]
0xa261: lea      edx, [rax + 1]
0xa264: mov      dword ptr [rip + 0x210ee], edx
0xa26a: cmp      eax, edi
0xa26c: jge      0xa2c2
0xa26e: mov      ecx, dword ptr [rip + 0x210e0]
0xa274: cmp      ecx, dword ptr [rip + 0x20722]
0xa27a: jge      0xa2c2
0xa27c: cmp      edx, edi
0xa27e: jle      0xa2e0
0xa280: mov      r12d, edi

```

```

0xa283: sub     r12d, eax
0xa286: movsxd  r13, r12d
0xa289: imul    edi, ecx
0xa28c: lea     rsi, [rsp + 7]
0xa291: mov     rdx, r13
0xa294: lea     ebx, [rdi + rax]
0xa297: mov     rdi, qword ptr [rip + 0x2109a]
0xa29e: movsxd  rbx, ebx
0xa2a1: add     rdi, rbx
0xa2a4: call    0x4bb0
0xa2a9: add     rbx, qword ptr [rip + 0x21080]
0xa2b0: movzx   esi, byte ptr [rip + 0x2109a]
0xa2b7: mov     rdx, r13
0xa2ba: mov     rdi, rbx
0xa2bd: call    0x4a50
0xa2c2: mov     rax, qword ptr [rsp + 8]
0xa2c7: sub     rax, qword ptr fs:[0x28]
0xa2d0: jne     0xa2ee
0xa2d2: add     rsp, 0x10
0xa2d6: mov     eax, r12d
0xa2d9: pop     rbx
0xa2da: pop     r12
0xa2dc: pop     r13
0xa2de: ret
0xa2df: nop
0xa2e0: mov     r13d, 1
0xa2e6: mov     r12d, 1
0xa2ec: jmp     0xa289
0xa289
0xa2ee: call    0x49d0
0xa2f3: nop     word ptr cs:[rax + rax]
0xa2fe: nop
0xa300: endbr64
0xa304: push    rbp
0xa305: lea     rdx, [rip - 0x89c]
0xa30c: push    rbx
0xa30d: sub     rsp, 8
0xa311: mov     esi, dword ptr [rip + 0x20685]
0xa317: mov     rdi, qword ptr [rip + 0x1faca]
0xa31e: call    0x4820
0xa323: mov     edx, dword ptr [rip + 0x20bdf]
0xa329: test    edx, edx
0xa32b: jle     0xa365
0xa32d: xor     ebp, ebp
0xa32f: lea     rbx, [rip + 0x20bea]
0xa336: jmp     0xa34c
0xa34c
0xa338: nop     dword ptr [rax + rax]

```

```

0xa340: mov     edx, dword ptr [rip + 0x20bc2]
0xa346: add     ebp, eax
0xa348: cmp     edx, ebp
0xa34a: jle     0xa365
0xa34c: sub     edx, ebp
0xa34e: movsxd  rsi, ebp
0xa351: mov     edi, 1
0xa356: add     rsi, rbx
0xa359: movsxd  rdx, edx
0xa35c: call    0x18da0
0xa361: test    eax, eax
0xa363: jns     0xa340
0xa365: mov     edx, dword ptr [rip + 0x20631]
0xa36b: xor     esi, esi
0xa36d: imul    edx, dword ptr [rip + 0x20624]
0xa374: mov     dword ptr [rip + 0x20b8a], 0
0xa37e: mov     rdi, qword ptr [rip + 0x20fb3]
0xa385: movsxd  rdx, edx
0xa388: call    0x4a50
0xa38d: mov     edx, dword ptr [rip + 0x20609]
0xa393: imul    edx, dword ptr [rip + 0x205fe]
0xa39a: xor     esi, esi
0xa39c: mov     rdi, qword ptr [rip + 0x20f8d]
0xa3a3: movsxd  rdx, edx
0xa3a6: call    0x4a50
0xa3ab: mov     edx, dword ptr [rip + 0x205eb]
0xa3b1: imul    edx, dword ptr [rip + 0x205e0]
0xa3b8: xor     esi, esi
0xa3ba: mov     rdi, qword ptr [rip + 0x20f67]
0xa3c1: movsxd  rdx, edx
0xa3c4: call    0x4a50
0xa3c9: mov     edx, dword ptr [rip + 0x205cd]
0xa3cf: imul    edx, dword ptr [rip + 0x205c2]
0xa3d6: xor     esi, esi
0xa3d8: mov     rdi, qword ptr [rip + 0x20f41]
0xa3df: movsxd  rdx, edx
0xa3e2: call    0x4a50
0xa3e7: mov     dword ptr [rip + 0x20f67], 0
0xa3f1: mov     dword ptr [rip + 0x20f59], 0
0xa3fb: add     rsp, 8
0xa3ff: pop     rbx
0xa400: pop     rbp
0xa401: ret
0xa402: nop     word ptr cs:[rax + rax]
0xa40d: nop     dword ptr [rax]
0xa410: endbr64
0xa414: sub     rsp, 8
0xa418: mov     edx, dword ptr [rip + 0x2057e]

```

```

0xa41e: imul    edx, dword ptr [rip + 0x20573]
0xa425: mov     esi, 0x20
0xa42a: mov     rdi, qword ptr [rip + 0x20f07]
0xa431: movsxd  rdx, edx
0xa434: call    0x4a50
0xa439: mov     edx, dword ptr [rip + 0x2055d]
0xa43f: imul    edx, dword ptr [rip + 0x20552]
0xa446: movzx   esi, byte ptr [rip + 0x20f04]
0xa44d: mov     rdi, qword ptr [rip + 0x20edc]
0xa454: movsxd  rdx, edx
0xa457: call    0x4a50
0xa45c: mov     edx, dword ptr [rip + 0x2053a]
0xa462: imul    edx, dword ptr [rip + 0x2052f]
0xa469: xor     esi, esi
0xa46b: mov     rdi, qword ptr [rip + 0x20eb6]
0xa472: add     rsp, 8
0xa476: movsxd  rdx, edx
0xa479: jmp     0x4a50
0x4a50:
0xa47e: nop
0xa480: endbr64
0xa484: mov     edx, dword ptr [rip + 0x20512]
0xa48a: imul    edx, dword ptr [rip + 0x20507]
0xa491: xor     esi, esi
0xa493: mov     rdi, qword ptr [rip + 0x20e8e]
0xa49a: movsxd  rdx, edx
0xa49d: jmp     0x4a50
0x4a50:
0xa4a2: nop     word ptr cs:[rax + rax]
0xa4ad: nop     dword ptr [rax]
0xa4b0: endbr64
0xa4b4: mov     dword ptr [rip + 0x20e9a], edi
0xa4ba: mov     dword ptr [rip + 0x20e98], esi
0xa4c0: ret
0xa4c1: nop     word ptr cs:[rax + rax]
0xa4cc: nop     dword ptr [rax]
0xa4d0: endbr64
0xa4d4: mov     eax, dword ptr [rip + 0x20e7a]
0xa4da: mov     dword ptr [rdi], eax
0xa4dc: mov     eax, dword ptr [rip + 0x20e76]
0xa4e2: mov     dword ptr [rsi], eax
0xa4e4: ret
0xa4e5: nop     word ptr cs:[rax + rax]
0xa4f0: endbr64
0xa4f4: movzx   eax, byte ptr [rip + 0x20e56]
0xa4fb: and     edi, 7
0xa4fe: and     eax, 0xffffffff8
0xa501: or      eax, edi

```

```

0xa503: mov      byte ptr [rip + 0x20e48], al
0xa509: ret
0xa50a: nop      word ptr [rax + rax]
0xa510: endbr64
0xa514: movzx    eax, byte ptr [rip + 0x20e36]
0xa51b: shl      edi, 3
0xa51e: and      edi, 0x38
0xa521: and      eax, 0xffffffffc7
0xa524: or       eax, edi
0xa526: mov      byte ptr [rip + 0x20e25], al
0xa52c: ret
0xa52d: nop      dword ptr [rax]
0xa530: endbr64
0xa534: movzx    eax, byte ptr [rip + 0x20e16]
0xa53b: shl      edi, 6
0xa53e: and      edi, 0x40
0xa541: and      eax, 0xffffffffbf
0xa544: or       eax, edi
0xa546: mov      byte ptr [rip + 0x20e05], al
0xa54c: ret
0xa54d: nop      dword ptr [rax]
0xa550: endbr64
0xa554: movzx    eax, byte ptr [rip + 0x20df6]
0xa55b: shl      edi, 7
0xa55e: and      eax, 0x7f
0xa561: or       eax, edi
0xa563: mov      byte ptr [rip + 0x20de8], al
0xa569: ret
0xa56a: nop      word ptr [rax + rax]
0xa570: endbr64
0xa574: shl      edi, 6
0xa577: and      esi, 7
0xa57a: shl      edx, 3
0xa57d: mov      eax, edi
0xa57f: movzx    edi, byte ptr [rip + 0x20dcb]
0xa586: and      edx, 0x38
0xa589: and      eax, 0x40
0xa58c: and      edi, 0xffffffffb8
0xa58f: or       edi, eax
0xa591: or       edi, esi
0xa593: and      edi, 0xffffffffc7
0xa596: or       edi, edx
0xa598: mov      byte ptr [rip + 0x20db2], dil
0xa59f: ret
0xa5a0: endbr64
0xa5a4: push     rbp
0xa5a5: push     rbx
0xa5a6: sub      rsp, 8

```

```

0xa5aa: movsxd  rax, dword ptr [rip + 0x20957]
0xa5b1: cmp      eax, 0x400
0xa5b6: je       0xa610
0xa5b8: lea      edx, [rax + 1]
0xa5bb: lea      rbx, [rip + 0x2095e]
0xa5c2: mov      dword ptr [rip + 0x20940], edx
0xa5c8: mov      byte ptr [rbx + rax], 7
0xa5cc: test     edx, edx
0xa5ce: jle      0xa5fd
0xa5d0: xor      ebp, ebp
0xa5d2: jmp      0xa5e4
0xa5e4:
0xa5d4: nop      dword ptr [rax]
0xa5d8: mov      edx, dword ptr [rip + 0x2092a]
0xa5de: add      ebp, eax
0xa5e0: cmp      edx, ebp
0xa5e2: jle      0xa5fd
0xa5e4: sub      edx, ebp
0xa5e6: movsxd   rsi, ebp
0xa5e9: mov      edi, 1
0xa5ee: add      rsi, rbx
0xa5f1: movsxd   rdx, edx
0xa5f4: call     0x18da0
0xa5f9: test     eax, eax
0xa5fb: jns      0xa5d8
0xa5fd: mov      dword ptr [rip + 0x20901], 0
0xa607: add      rsp, 8
0xa60b: pop      rbx
0xa60c: pop      rbp
0xa60d: ret
0xa60e: nop
0xa610: xor      ebp, ebp
0xa612: lea      rbx, [rip + 0x20907]
0xa619: jmp      0xa62c
0xa62c:
0xa61b: nop      dword ptr [rax + rax]
0xa620: add      ebp, eax
0xa622: mov      eax, dword ptr [rip + 0x208e0]
0xa628: cmp      eax, ebp
0xa62a: jle      0xa645
0xa62c: sub      eax, ebp
0xa62e: movsxd   rsi, ebp
0xa631: mov      edi, 1
0xa636: add      rsi, rbx
0xa639: movsxd   rdx, eax
0xa63c: call     0x18da0
0xa641: test     eax, eax
0xa643: jns      0xa620

```

```

0xa645: mov     dword ptr [rip + 0x208b9], 1
0xa64f: mov     edx, 1
0xa654: mov     byte ptr [rip + 0x208c5], 7
0xa65b: jmp     0xa5d0
0xa5d0
0xa660: endbr64
0xa664: test    edi, edi
0xa666: je      0xa688
0xa668: mov     rdi, qword ptr [rip + 0x1f759]
0xa66f: test    rdi, rdi
0xa672: je      0xa694
0xa674: lea     rdx, [rip - 0xc0b]
0xa67b: mov     esi, 1
0xa680: jmp     0x4820
0x4820
0xa685: nop     dword ptr [rax]
0xa688: mov     rdi, qword ptr [rip + 0x1f719]
0xa68f: test    rdi, rdi
0xa692: jne     0xa674
0xa694: ret
0xa695: nop     word ptr cs:[rax + rax]
0xa6a0: endbr64
0xa6a4: movzx   eax, byte ptr [rip + 0x20ca6]
0xa6ab: mov     byte ptr [rdi], al
0xa6ad: ret
0xa6ae: nop
0xa6b0: endbr64
0xa6b4: movzx   eax, byte ptr [rdi]
0xa6b7: mov     byte ptr [rip + 0x20c94], al
0xa6bd: ret
0xa6be: nop
0xa6c0: endbr64
0xa6c4: push    rbp
0xa6c5: push    rbx
0xa6c6: sub     rsp, 8
0xa6ca: cmp     dword ptr [rip + 0x1f58f], 1
0xa6d1: je      0xa720
0xa6d3: mov     rdi, qword ptr [rip + 0x1f66e]
0xa6da: test    rdi, rdi
0xa6dd: je      0xa6f0
0xa6df: lea     rdx, [rip - 0xc76]
0xa6e6: mov     esi, 1
0xa6eb: call    0x4820
0xa6f0: mov     dword ptr [rip + 0x20c52], 0
0xa6fa: mov     dword ptr [rip + 0x20c44], 0
0xa704: mov     dword ptr [rip + 0x20c36], 0
0xa70e: mov     dword ptr [rip + 0x20c28], 0
0xa718: add     rsp, 8

```



```

0xa71c: pop      rbx
0xa71d: pop      rbp
0xa71e: ret
0xa71f: nop
0xa720: mov      edx, dword ptr [rip + 0x207e2]
0xa726: cmp      edx, 0x3fb
0xa72c: jg       0xa750
0xa72e: movsxd   rax, edx
0xa731: lea      rcx, [rip + 0x207e8]
0xa738: add      edx, 4
0xa73b: add      rax, rcx
0xa73e: mov      ecx, dword ptr [rip + 0x1f5db]
0xa744: mov      dword ptr [rip + 0x207be], edx
0xa74a: mov      dword ptr [rax], ecx
0xa74c: jmp      0xa6d3
0xa6d3:
0xa74e: nop
0xa750: xor      ebx, ebx
0xa752: lea      rbp, [rip + 0x207c7]
0xa759: jmp      0xa76c
0xa76c:
0xa75b: nop      dword ptr [rax + rax]
0xa760: mov      edx, dword ptr [rip + 0x207a2]
0xa766: add      ebx, eax
0xa768: cmp      edx, ebx
0xa76a: jle      0xa785
0xa76c: sub      edx, ebx
0xa76e: movsxd   rsi, ebx
0xa771: mov      edi, 1
0xa776: add      rsi, rbp
0xa779: movsxd   rdx, edx
0xa77c: call     0x18da0
0xa781: test     eax, eax
0xa783: jns      0xa760
0xa785: mov      edx, 4
0xa78a: lea      rax, [rip + 0x2078f]
0xa791: jmp      0xa73e
0xa73e:
0xa793: nop      word ptr cs:[rax + rax]
0xa79e: nop
0xa7a0: push     r12
0xa7a2: mov      r12d, edi
0xa7a5: push     rbp
0xa7a6: mov      ebp, esi
0xa7a8: sub      rsp, 8
0xa7ac: mov      eax, dword ptr [rip + 0x1f71e]
0xa7b2: test     eax, eax
0xa7b4: je       0xa800

```

```

0xa7b6: mov     eax, ebp
0xa7b8: mov     esi, dword ptr [rip + 0x201da]
0xa7be: mov     edx, dword ptr [rip + 0x201d8]
0xa7c4: or      eax, r12d
0xa7c7: mov     rdi, qword ptr [rip + 0x1f63a]
0xa7ce: js      0xa7d4
0xa7d0: cmp     ebp, esi
0xa7d2: jl      0xa810
0xa7d4: sub     edx, 1
0xa7d7: sub     esi, 1
0xa7da: call    0x4960
0xa7df: add     rsp, 8
0xa7e3: mov     esi, 1
0xa7e8: lea     rdx, [rip - 0xd7f]
0xa7ef: pop     rbp
0xa7f0: mov     rdi, rax
0xa7f3: pop     r12
0xa7f5: jmp     0x4820
0x4820:
0xa7fa: nop     word ptr [rax + rax]
0xa800: xor     eax, eax
0xa802: call    0xa6c0
0xa807: jmp     0xa7b6
0xa7b6:
0xa809: nop     dword ptr [rax]
0xa810: cmp     r12d, edx
0xa813: jge     0xa7d4
0xa815: mov     edx, r12d
0xa818: mov     esi, ebp
0xa81a: jmp     0xa7da
0xa7da:
0xa81c: nop     dword ptr [rax]
0xa820: endbr64
0xa824: push    r15
0xa826: push    r14
0xa828: push    r13
0xa82a: push    r12
0xa82c: push    rbp
0xa82d: push    rbx
0xa82e: sub     rsp, 0x38
0xa832: mov     ebx, dword ptr [rip + 0x20160]
0xa838: imul    ebx, dword ptr [rip + 0x2015d]
0xa83f: mov     rax, qword ptr fs:[0x28]
0xa848: mov     qword ptr [rsp + 0x28], rax
0xa84d: xor     eax, eax
0xa84f: mov     rdi, qword ptr [rip + 0x1f552]
0xa856: cmp     dword ptr [rip + 0x20aff], 1
0xa85d: sbb     ebx, 0

```

```

0xa860: test    rdi, rdi
0xa863: je      0xa876
0xa865: lea     rdx, [rip - 0xdfc]
0xa86c: mov     esi, 1
0xa871: call    0x4820
0xa876: test    ebx, ebx
0xa878: jle     0xab70
0xa87e: movsxd  rax, ebx
0xa881: mov     dword ptr [rsp + 4], 0xffffffff
0xa889: xor     ebp, ebp
0xa88b: mov     r13d, 0xffffffff
0xa891: mov     qword ptr [rsp + 8], rax
0xa896: lea     r12, [rip + 0x20683]
0xa89d: jmp     0xa9ce
0xa9ce
0xa8a2: nop     word ptr [rax + rax]
0xa8a8: cdq
0xa8a9: idiv    dword ptr [rip + 0x200e9]
0xa8af: mov     r14d, edx
0xa8b2: cmp     edx, r13d
0xa8b5: jne     0xaa20
0xa8bb: cmp     eax, dword ptr [rsp + 4]
0xa8bf: jne     0xaa20
0xa8c5: mov     rax, qword ptr [rip + 0x20a64]
0xa8cc: movzx   ebx, byte ptr [rax + rbp]
0xa8d0: mov     edi, ebx
0xa8d2: mov     r14d, ebx
0xa8d5: sar     edi, 6
0xa8d8: and     r14d, 7
0xa8dc: and     edi, 1
0xa8df: call    0x9870
0xa8e4: mov     eax, ebx
0xa8e6: and     eax, 7
0xa8e9: cmp     dword ptr [rip + 0x20a5c], 1
0xa8f0: je      0xaa40
0xa8f6: cmp     dword ptr [rip + 0x1f363], 1
0xa8fd: je      0xaa5f
0xa903: xor     edi, edi
0xa905: cmp     eax, 7
0xa908: setne   dil
0xa90c: call    0x9790
0xa911: movzx   eax, byte ptr [rip + 0x20a38]
0xa918: mov     dword ptr [rip + 0x20a2a], 1
0xa922: and     eax, 0xffffffff8
0xa925: or      eax, r14d
0xa928: mov     byte ptr [rip + 0x20a22], al
0xa92e: sar     ebx, 3
0xa931: and     ebx, 7

```

```

0xa934: cmp     dword ptr [rip + 0x20a0d], 1
0xa93b: je      0xaab0
0xa941: cmp     dword ptr [rip + 0x1f318], 1
0xa948: je      0xaad8
0xa94e: xor     edi, edi
0xa950: test    ebx, ebx
0xa952: setne   dil
0xa956: call    0x9790
0xa95b: mov     edx, dword ptr [rip + 0x205a7]
0xa961: movzx   eax, byte ptr [rip + 0x209e8]
0xa968: shl     ebx, 3
0xa96b: mov     dword ptr [rip + 0x209d3], 1
0xa975: and     eax, 0xffffffffc7
0xa978: or      ebx, eax
0xa97a: mov     byte ptr [rip + 0x209d0], bl
0xa980: mov     rax, qword ptr [rip + 0x209b1]
0xa987: movzx   ebx, byte ptr [rax + rbp]
0xa98b: cmp     edx, 0x400
0xa991: je      0xab30
0xa997: lea     eax, [rdx + 1]
0xa99a: movsxd  rdx, edx
0xa99d: add     r13d, 1
0xa9a1: cmp     dword ptr [rip + 0x1fff0], r13d
0xa9a8: mov     dword ptr [rip + 0x2055a], eax
0xa9ae: mov     byte ptr [r12 + rdx], bl
0xa9b2: jne     0xa9bf
0xa9b4: mov     dword ptr [rsp + 4], 0xffffffff
0xa9bc: xor     r13d, r13d
0xa9bf: add     rbp, 1
0xa9c3: cmp     rbp, qword ptr [rsp + 8]
0xa9c8: je      0xab70
0xa9ce: mov     rdx, qword ptr [rip + 0x20953]
0xa9d5: mov     rcx, qword ptr [rip + 0x2095c]
0xa9dc: mov     eax, ebp
0xa9de: movzx   ebx, byte ptr [rdx + rbp]
0xa9e2: cmp     byte ptr [rcx + rbp], bl
0xa9e5: jne     0xa8a8
0xa9eb: mov     rdx, qword ptr [rip + 0x2092e]
0xa9f2: mov     rcx, qword ptr [rip + 0x20937]
0xa9f9: movzx   edi, byte ptr [rdx + rbp]
0xa9fd: cmp     byte ptr [rcx + rbp], dil
0xaa01: je      0xa9bf
0xaa03: cdq
0xaa04: idiv    dword ptr [rip + 0x1ff8e]
0xaa0a: mov     r14d, edx
0xaa0d: cmp     edx, r13d
0xaa10: je      0xa8bb
0xaa16: nop     word ptr cs:[rax + rax]

```

```

0xaa20: mov     esi, r14d
0xaa23: mov     edi, eax
0xaa25: mov     dword ptr [rsp + 4], eax
0xaa29: mov     r13d, r14d
0xaa2c: call    0xa7a0
0xaa31: jmp     0xa8c5
0xa8c5:
0xaa36: nop     word ptr cs:[rax + rax]
0xaa40: movzx   edx, byte ptr [rip + 0x20909]
0xaa47: and     edx, 7
0xaa4a: cmp     eax, edx
0xaa4c: je      0xa92e
0xaa52: cmp     dword ptr [rip + 0x1f207], 1
0xaa59: jne     0xa903
0xaa5f: mov     eax, dword ptr [rip + 0x164b4]
0xaa65: mov     edx, dword ptr [rip + 0x2049d]
0xaa6b: mov     dword ptr [rsp + 0x10], eax
0xaa6f: movzx   eax, byte ptr [rip + 0x164a7]
0xaa76: mov     byte ptr [rsp + 0x14], al
0xaa7a: lea     eax, [r14 + 0x30]
0xaa7e: mov     byte ptr [rsp + 0x13], al
0xaa82: cmp     edx, 0x3fa
0xaa88: jg      0xac18
0xaa8e: movsxd  rax, edx
0xaa91: add     edx, 5
0xaa94: add     rax, r12
0xaa97: mov     ecx, dword ptr [rsp + 0x10]
0xaa9b: mov     dword ptr [rip + 0x20467], edx
0xaaa1: mov     dword ptr [rax], ecx
0xaaa3: movzx   ecx, byte ptr [rsp + 0x14]
0xaaa8: mov     byte ptr [rax + 4], cl
0xaaab: jmp     0xa911
0xa911:
0xaab0: movzx   eax, byte ptr [rip + 0x20899]
0xaab7: mov     edx, dword ptr [rip + 0x2044b]
0xaabd: sar     eax, 3
0xaac0: and     eax, 7
0xaac3: cmp     ebx, eax
0xaac5: je      0xa980
0xaacb: cmp     dword ptr [rip + 0x1f18e], 1
0xaad2: jne     0xa94e
0xaad8: mov     eax, dword ptr [rip + 0x16436]
0xaade: mov     edx, dword ptr [rip + 0x20424]
0xaae4: mov     dword ptr [rsp + 0x10], eax
0xaae8: movzx   eax, byte ptr [rip + 0x16429]
0xaaef: mov     byte ptr [rsp + 0x14], al
0xaaf3: lea     eax, [rbx + 0x30]
0xaaf6: mov     byte ptr [rsp + 0x13], al

```

```

0xaafa: cmp     edx, 0x3fa
0xab00: jg         0xac58
0xab06: movsxd    rax, edx
0xab09: add       edx, 5
0xab0c: add       rax, r12
0xab0f: mov       ecx, dword ptr [rsp + 0x10]
0xab13: mov       dword ptr [rip + 0x203ef], edx
0xab19: mov       dword ptr [rax], ecx
0xab1b: movzx     ecx, byte ptr [rsp + 0x14]
0xab20: mov       byte ptr [rax + 4], cl
0xab23: jmp       0xa961
0xa961:
0xab28: nop       dword ptr [rax + rax]
0xab30: xor       r14d, r14d
0xab33: jmp       0xab46
0xab46:
0xab35: nop       dword ptr [rax]
0xab38: mov       edx, dword ptr [rip + 0x203ca]
0xab3e: add       r14d, eax
0xab41: cmp       edx, r14d
0xab44: jle       0xab60
0xab46: sub       edx, r14d
0xab49: movsxd    rsi, r14d
0xab4c: mov       edi, 1
0xab51: add       rsi, r12
0xab54: movsxd    rdx, edx
0xab57: call     0x18da0
0xab5c: test     eax, eax
0xab5e: jns       0xab38
0xab60: mov       eax, 1
0xab65: xor       edx, edx
0xab67: jmp       0xa99a
0xa99a:
0xab6c: nop       dword ptr [rax]
0xab70: mov       edi, dword ptr [rip + 0x207de]
0xab76: mov       esi, dword ptr [rip + 0x207dc]
0xab7c: call     0xa7a0
0xab81: mov       rdi, qword ptr [rip + 0x1f240]
0xab88: test     rdi, rdi
0xab8b: je        0xab9e
0xab8d: lea       rdx, [rip - 0x1124]
0xab94: mov       esi, 1
0xab99: call     0x4820
0xab9e: mov       edx, dword ptr [rip + 0x20364]
0xaba4: test     edx, edx
0xaba6: jne       0xac98
0xabac: mov       edx, dword ptr [rip + 0x1fde6]
0xabb2: imul     edx, dword ptr [rip + 0x1fde3]

```

```

0xab9: mov     rsi, qword ptr [rip + 0x20778]
0xabc0: mov     rdi, qword ptr [rip + 0x20761]
0xabc7: movsxd   rdx, edx
0xabca: call    0x4bb0
0xabcf: mov     edx, dword ptr [rip + 0x1fdc3]
0xabd5: imul    edx, dword ptr [rip + 0x1fdc0]
0xabdc: mov     rdi, qword ptr [rip + 0x2073d]
0xabe3: mov     rsi, qword ptr [rip + 0x20746]
0xabea: movsxd   rdx, edx
0xabed: mov     rax, qword ptr [rsp + 0x28]
0xabf2: sub     rax, qword ptr fs:[0x28]
0xabfb: jne     0xace4
0xac01: add     rsp, 0x38
0xac05: pop     rbx
0xac06: pop     rbp
0xac07: pop     r12
0xac09: pop     r13
0xac0b: pop     r14
0xac0d: pop     r15
0xac0f: jmp     0x4bb0
0x4bb0
0xac14: nop     dword ptr [rax]
0xac18: xor     r15d, r15d
0xac1b: jmp     0xac2e
0xac2e
0xac1d: nop     dword ptr [rax]
0xac20: mov     edx, dword ptr [rip + 0x202e2]
0xac26: add     r15d, eax
0xac29: cmp     edx, r15d
0xac2c: jle     0xac48
0xac2e: sub     edx, r15d
0xac31: movsxd   rsi, r15d
0xac34: mov     edi, 1
0xac39: add     rsi, r12
0xac3c: movsxd   rdx, edx
0xac3f: call    0x18da0
0xac44: test    eax, eax
0xac46: jns     0xac20
0xac48: mov     edx, 5
0xac4d: mov     rax, r12
0xac50: jmp     0xaa97
0xaa97
0xac55: nop     dword ptr [rax]
0xac58: xor     r14d, r14d
0xac5b: jmp     0xac6e
0xac6e
0xac5d: nop     dword ptr [rax]
0xac60: mov     edx, dword ptr [rip + 0x202a2]

```

```

0xac66: add     r14d, eax
0xac69: cmp     edx, r14d
0xac6c: jle     0xac88
0xac6e: sub     edx, r14d
0xac71: movsxd  rsi, r14d
0xac74: mov     edi, 1
0xac79: add     rsi, r12
0xac7c: movsxd  rdx, edx
0xac7f: call    0x18da0
0xac84: test    eax, eax
0xac86: jns     0xac60
0xac88: mov     edx, 5
0xac8d: mov     rax, r12
0xac90: jmp     0xab0f
0xab0f:
0xac95: nop     dword ptr [rax]
0xac98: mov     ebx, 0
0xac9d: lea     rbp, [rip + 0x2027c]
0xaca4: jg      0xacbc
0xaca6: jmp     0xacd5
0xacd5:
0xaca8: nop     dword ptr [rax + rax]
0xacb0: mov     edx, dword ptr [rip + 0x20252]
0xacb6: add     ebx, eax
0xacb8: cmp     edx, ebx
0xacba: jle     0xacd5
0xacbc: sub     edx, ebx
0xacbe: movsxd  rsi, ebx
0xacc1: mov     edi, 1
0xacc6: add     rsi, rbp
0xacc9: movsxd  rdx, edx
0xaccc: call    0x18da0
0xacd1: test    eax, eax
0xacd3: jns     0xacb0
0xacd5: mov     dword ptr [rip + 0x20229], 0
0xacdf: jmp     0xabac
0xabac:
0xace4: call    0x49d0
0xace9: nop     dword ptr [rax]
0xacf0: endbr64
0xacf4: push    r12
0xacf6: xor     eax, eax
0xacf8: mov     r12, rdi
0xacfb: push    rbx
0xacfc: mov     ebx, esi
0xacfe: sub     rsp, 8
0xad02: call    0xa820
0xad07: mov     rdx, qword ptr [rip + 0x1fc1a]

```



```

0xad0e: test    rdx, rdx
0xad11: je      0xad17
0xad13: xor     eax, eax
0xad15: call    rdx
0xad17: movsxd  rdx, ebx
0xad1a: mov     rsi, r12
0xad1d: xor     edi, edi
0xad1f: call    0x18d30
0xad24: mov     rdx, qword ptr [rip + 0x1fbf5]
0xad2b: mov     r12d, eax
0xad2e: test    rdx, rdx
0xad31: je      0xad37
0xad33: xor     eax, eax
0xad35: call    rdx
0xad37: add     rsp, 8
0xad3b: mov     eax, r12d
0xad3e: pop     rbx
0xad3f: pop     r12
0xad41: ret
0xad42: nop     word ptr cs:[rax + rax]
0xad4d: nop     dword ptr [rax]
0xad50: push    r12
0xad52: mov     edi, 1
0xad57: push    rbp
0xad58: lea     rbp, [rip + 0x1fda1]
0xad5f: push    rbx
0xad60: sub     rsp, 0x50
0xad64: mov     rax, qword ptr fs:[0x28]
0xad6d: mov     qword ptr [rsp + 0x48], rax
0xad72: xor     eax, eax
0xad74: mov     r12, rsp
0xad77: mov     rsi, r12
0xad7a: call    0x4d60
0xad7f: mov     edx, 0xff
0xad84: mov     esi, 1
0xad89: mov     edi, 1
0xad8e: mov     word ptr [rsp + 0x11], dx
0xad93: mov     rdx, r12
0xad96: call    0x4d70
0xad9b: mov     edi, 1
0xada0: mov     dword ptr [rip + 0x1ef7a], 0xffffffff
0xadaa: call    0x18b70
0xadaf: mov     dword ptr [rip + 0x20147], 0
0xadb9: nop     dword ptr [rax]
0xadc0: xor     eax, eax
0xadc2: call    0xa820
0xadc7: mov     rdx, qword ptr [rip + 0x1fb5a]
0xadce: test    rdx, rdx

```

```

0xadd1: je      0xadd7
0xadd3: xor      eax, eax
0xadd5: call     rdx
0xadd7: mov      edx, 0x400
0xaddc: mov      rsi, rbp
0xaddf: xor      edi, edi
0xade1: call     0x18d30
0xade6: mov      rdx, qword ptr [rip + 0x1fb33]
0xaded: mov      ebx, eax
0xadeF: test     rdx, rdx
0xadf2: je      0xadf8
0xadf4: xor      eax, eax
0xadf6: call     rdx
0xadf8: mov      dword ptr [rip + 0x20106], ebx
0xadfe: test     ebx, ebx
0xae00: js      0xadC0
0xae02: xor      edi, edi
0xae04: call     0x18b70
0xae09: mov      rsi, r12
0xae0c: mov      edi, 1
0xae11: call     0x4d60
0xae16: mov      eax, 7
0xae1b: mov      rdx, r12
0xae1e: mov      esi, 1
0xae23: mov      edi, 1
0xae28: mov      word ptr [rsp + 0x11], ax
0xae2d: call     0x4d70
0xae32: mov      eax, dword ptr [rip + 0x200cc]
0xae38: mov      dword ptr [rip + 0x1eee2], 7
0xae42: test     eax, eax
0xae44: je      0xae9f
0xae46: movsxd   rsi, dword ptr [rip + 0x200b3]
0xae4d: movsx    r8d, byte ptr [rbp + rsi]
0xae53: mov      rdx, rsi
0xae56: mov      ecx, r8d
0xae59: test     r8b, r8b
0xae5c: js      0xae90
0xae5e: sub      eax, 1
0xae61: add      edx, 1
0xae64: mov      dword ptr [rip + 0x2009a], eax
0xae6a: mov      dword ptr [rip + 0x20090], edx
0xae70: mov      rax, qword ptr [rsp + 0x48]
0xae75: sub      rax, qword ptr fs:[0x28]
0xae7e: jne     0xaea7
0xae80: add      rsp, 0x50
0xae84: mov      eax, r8d
0xae87: pop      rbx
0xae88: pop      rbp

```

```

0xae89: pop     r12
0xae8b: ret
0xae8c: nop     dword ptr [rax]
0xae90: and     ecx, 0x7f
0xae93: mov     r8d, 0x1b
0xae99: mov     byte ptr [rbp + rsi], cl
0xae9d: jmp     0xae70
0xae70:
0xae9f: mov     r8d, 0xffffffff
0xaea5: jmp     0xae70
0xae70:
0xaea7: call    0x49d0
0xaeac: nop     dword ptr [rax]
0xaeb0: endbr64
0xaeb4: xor     eax, eax
0xaeb6: sub     rsp, 8
0xaeba: call    0x18ab0
0xaebf: mov     eax, dword ptr [rip + 0x2003f]
0xaec5: test    eax, eax
0xaec7: jne     0xaed8
0xaec9: xor     eax, eax
0xaecb: add     rsp, 8
0xaecf: jmp     0xad50
0xad50:
0xaed4: nop     dword ptr [rax]
0xaed8: movsxd  rsi, dword ptr [rip + 0x20021]
0xaedf: lea     rdi, [rip + 0x1fc1a]
0xae6: movsx  r8d, byte ptr [rdi + rsi]
0xaeeb: mov     rdx, rsi
0xaeee: mov     ecx, r8d
0xae1: test   r8b, r8b
0xae4: js     0xaf10
0xae6: sub     eax, 1
0xae9: add     edx, 1
0xae1c: mov     dword ptr [rip + 0x20002], eax
0xaf02: mov     dword ptr [rip + 0x1fff8], edx
0xaf08: mov     eax, r8d
0xaf0b: add     rsp, 8
0xaf0f: ret
0xaf10: and     ecx, 0x7f
0xaf13: mov     r8d, 0x1b
0xaf19: mov     byte ptr [rdi + rsi], cl
0xaf1c: jmp     0xaf08
0xaf08:
0xaf1e: nop
0xaf20: endbr64
0xaf24: push    r13
0xaf26: mov     r13, rsi

```

```

0xaf29: push    r12
0xaf2b: mov     r12, rdx
0xaf2e: push    rbp
0xaf2f: mov     rbp, rdi
0xaf32: mov     edi, 0x18
0xaf37: push    rbx
0xaf38: sub     rsp, 8
0xaf3c: call    0x1d670
0xaf41: mov     rdi, r13
0xaf44: mov     rbx, rax
0xaf47: call    0x1d7a0
0xaf4c: mov     qword ptr [rbx + 0x10], r12
0xaf50: mov     qword ptr [rbx], rax
0xaf53: mov     rax, qword ptr [rbp]
0xaf57: mov     qword ptr [rbx + 8], rax
0xaf5b: mov     qword ptr [rbp], rbx
0xaf5f: add     rsp, 8
0xaf63: pop     rbx
0xaf64: pop     rbp
0xaf65: pop     r12
0xaf67: pop     r13
0xaf69: ret
0xaf6a: nop     word ptr [rax + rax]
0xaf70: endbr64
0xaf74: cmp     byte ptr [rdi], 0
0xaf77: je      0xb030
0xaf7d: push    r13
0xaf7f: push    r12
0xaf81: mov     r12, rsi
0xaf84: push    rbp
0xaf85: mov     rbp, rdi
0xaf88: push    rbx
0xaf89: sub     rsp, 8
0xaf8d: mov     r13, qword ptr [rip + 0x1fa0c]
0xaf94: test    r13, r13
0xaf97: je      0xb010
0xaf99: mov     rbx, qword ptr [r13]
0xaf9d: mov     rsi, qword ptr [rbx]
0xafa0: call    0x4b30
0xafa5: test    eax, eax
0xafa7: jle     0xb010
0xafa9: nop     dword ptr [rax]
0xafb0: lea     r13, [rbx + 8]
0xafb4: mov     rbx, qword ptr [rbx + 8]
0xafb8: mov     qword ptr [rip + 0x1f9e1], r13
0xafbf: test    rbx, rbx
0xafc2: je      0xafd3
0xafc4: mov     rsi, qword ptr [rbx]

```

```

0xafc7: mov     rdi, rbp
0xafca: call    0x4b30
0xafcf: test    eax, eax
0xafd1: jg      0xafb0
0xafd3: mov     edi, 0x18
0xafd8: call    0x1d670
0xafdd: mov     rdi, rbp
0xafe0: mov     rbx, rax
0xafe3: call    0x1d7a0
0xafe8: mov     qword ptr [rbx + 0x10], r12
0xafec: mov     qword ptr [rbx], rax
0xafef: mov     rax, qword ptr [r13]
0xaff3: mov     qword ptr [rbx + 8], rax
0xaff7: mov     qword ptr [r13], rbx
0xaffb: add     rsp, 8
0xafff: pop     rbx
0xb000: pop     rbp
0xb001: pop     r12
0xb003: pop     r13
0xb005: ret
0xb006: nop     word ptr cs:[rax + rax]
0xb010: mov     rbx, qword ptr [rip + 0x1f969]
0xb017: lea     r13, [rip + 0x1f962]
0xb01e: mov     qword ptr [rip + 0x1f97b], r13
0xb025: test    rbx, rbx
0xb028: jne     0xafc4
0xb02a: jmp     0xafd3
0xafd3:
0xb02c: nop     dword ptr [rax]
0xb030: ret
0xb031: nop     word ptr cs:[rax + rax]
0xb03c: nop     dword ptr [rax]
0xb040: endbr64
0xb044: mov     rax, qword ptr [rip + 0x1f935]
0xb04b: mov     qword ptr [rip + 0x1f926], rax
0xb052: ret
0xb053: nop     word ptr cs:[rax + rax]
0xb05e: nop
0xb060: endbr64
0xb064: cmp     qword ptr [rip + 0x1f90c], 0
0xb06c: je      0xb073
0xb06e: jmp     0x99e0
0x99e0:
0xb073: xor     eax, eax
0xb075: ret
0xb076: nop     word ptr cs:[rax + rax]
0xb080: endbr64
0xb084: mov     eax, dword ptr [rip + 0x1f912]

```

```

0xb08a: push    r12
0xb08c: mov     r12, rdi
0xb08f: push    rbp
0xb090: push    rbx
0xb091: sub     eax, 1
0xb094: movzx   ebx, byte ptr [rip + 0x202b6]
0xb09b: mov     dword ptr [rip + 0x202b3], eax
0xb0a1: mov     dword ptr [rip + 0x202ad], 0
0xb0ab: mov     eax, ebx
0xb0ad: and     eax, 0xffffffffc0
0xb0b0: or      eax, 0x38
0xb0b3: mov     byte ptr [rip + 0x20298], al
0xb0b9: mov     eax, dword ptr [rip + 0x1f8d9]
0xb0bf: lea     edi, [rax + 1]
0xb0c2: movsxd  rdi, edi
0xb0c5: call    0x1d670
0xb0ca: movsxd  rdx, dword ptr [rip + 0x1f8c7]
0xb0d1: mov     esi, 0x20
0xb0d6: mov     rdi, rax
0xb0d9: mov     rbp, rax
0xb0dc: call    0x4a50
0xb0e1: movsxd  rax, dword ptr [rip + 0x1f8b0]
0xb0e8: mov     rdi, rbp
0xb0eb: mov     byte ptr [rbp + rax], 0
0xb0f0: mov     esi, dword ptr [rip + 0x1f8a2]
0xb0f6: call    0xa1a0
0xb0fb: mov     rdi, rbp
0xb0fe: lea     rbp, [rip + 0x1f8bb]
0xb105: call    0x18d10
0xb10a: mov     eax, dword ptr [rip + 0x1f88c]
0xb110: mov     rdi, r12
0xb113: mov     dword ptr [rip + 0x2023b], 0
0xb11d: sub     eax, 1
0xb120: mov     dword ptr [rip + 0x2022e], eax
0xb126: call    0x9fe0
0xb12b: mov     esi, 0x13
0xb130: lea     rdi, [rip + 0x15ad5]
0xb137: call    0xa1a0
0xb13c: mov     rdi, rbp
0xb13f: call    0x49b0
0xb144: mov     rdi, rbp
0xb147: mov     esi, eax
0xb149: call    0xa1a0
0xb14e: mov     esi, 1
0xb153: lea     rdi, [rip + 0x155eb]
0xb15a: call    0xa1a0
0xb15f: xor     eax, eax
0xb161: call    0xa820

```

```

0xb166: mov      byte ptr [rip + 0x201e5], bl
0xb16c: pop      rbx
0xb16d: pop      rbp
0xb16e: pop      r12
0xb170: ret
0xb171: nop      word ptr cs:[rax + rax]
0xb17c: nop      dword ptr [rax]
0xb180: endbr64
0xb184: push     r15
0xb186: push     r14
0xb188: mov      r14, rdi
0xb18b: push     r13
0xb18d: push     r12
0xb18f: push     rbp
0xb190: lea      rbp, [rip + 0x1f969]
0xb197: push     rbx
0xb198: sub      rsp, 0x28
0xb19c: jmp      0xb1ae
0xb1ae:
0xb19e: nop
0xb1a0: xor      eax, eax
0xb1a2: call     0xad50
0xb1a7: mov      ebx, eax
0xb1a9: cmp      ebx, -1
0xb1ac: jne      0xb1ef
0xb1ae: xor      eax, eax
0xb1b0: call     0x18ab0
0xb1b5: mov      eax, dword ptr [rip + 0x1fd49]
0xb1bb: test     eax, eax
0xb1bd: je       0xb1a0
0xb1bf: movsxd   rsi, dword ptr [rip + 0x1fd3a]
0xb1c6: movsx    ebx, byte ptr [rbp + rsi]
0xb1cb: mov      rdx, rsi
0xb1ce: mov      ecx, ebx
0xb1d0: test     bl, bl
0xb1d2: js       0xb4a0
0xb1d8: sub      eax, 1
0xb1db: add      edx, 1
0xb1de: mov      dword ptr [rip + 0x1fd20], eax
0xb1e4: mov      dword ptr [rip + 0x1fd16], edx
0xb1ea: cmp      ebx, -1
0xb1ed: je       0xb1ae
0xb1ef: test     r14, r14
0xb1f2: je       0xb1fb
0xb1f4: mov      dword ptr [r14], 1
0xb1fb: mov      eax, dword ptr [rip + 0x201fb]
0xb201: test     ebx, ebx
0xb203: jne      0xb445

```

```

0xb209: mov     ebx, 0xff
0xb20e: test    eax, eax
0xb210: jne     0xb240
0xb212: xor     r12d, r12d
0xb215: mov     ebp, 0x1fe
0xb21a: mov     ebx, 0xff
0xb21f: call    0x4f00
0xb224: mov     rax, qword ptr [rax]
0xb227: movzx   eax, word ptr [rax + rbp]
0xb22b: test    ah, 0x40
0xb22e: jne     0xb47d
0xb234: test    r12b, r12b
0xb237: jne     0xb47d
0xb23d: nop
0xb240: mov     rax, qword ptr [rip + 0x1f739]
0xb247: xor     r12d, r12d
0xb24a: lea     r13, [rip + 0x1f8af]
0xb251: mov     dword ptr [rip + 0x1f751], 0
0xb25b: mov     dword ptr [rip + 0x1f743], 0
0xb265: mov     qword ptr [rip + 0x1f70c], rax
0xb26c: test    ebx, ebx
0xb26e: mov     eax, 0xffffffff
0xb273: movsxd  rbp, r12d
0xb276: cmovne  eax, ebx
0xb279: mov     rbx, qword ptr [rip + 0x200e0]
0xb280: mov     byte ptr [rbx + rbp], al
0xb283: add     rbp, 1
0xb287: mov     byte ptr [rbx + rbp], 0
0xb28b: cmp     qword ptr [rip + 0x1f6e5], 0
0xb293: je      0xb2b3
0xb295: mov     rdi, rbx
0xb298: call    0x99e0
0xb29d: mov     r15, rax
0xb2a0: cmp     rax, -1
0xb2a4: je      0xb33a
0xb2aa: test    rax, rax
0xb2ad: jne     0xb38f
0xb2b3: mov     ecx, dword ptr [rip + 0x1fc4b]
0xb2b9: test    ecx, ecx
0xb2bb: jne     0xb2ee
0xb2bd: mov     edx, dword ptr [rip + 0x1f6e5]
0xb2c3: test    edx, edx
0xb2c5: je      0xb380
0xb2cb: mov     rdi, rbx
0xb2ce: call    0xb080
0xb2d3: mov     dword ptr [rip + 0x1f6cf], 1
0xb2dd: jmp     0xb2ee
0xb2ee:

```



```

0xb2df: nop
0xb2e0: xor     eax, eax
0xb2e2: call    0xad50
0xb2e7: mov     ebx, eax
0xb2e9: cmp     ebx, -1
0xb2ec: jne     0xb32c
0xb2ee: xor     eax, eax
0xb2f0: call    0x18ab0
0xb2f5: mov     eax, dword ptr [rip + 0x1fc09]
0xb2fb: test    eax, eax
0xb2fd: je      0xb2e0
0xb2ff: movsxd  rsi, dword ptr [rip + 0x1fbfa]
0xb306: movsx   ebx, byte ptr [r13 + rsi]
0xb30c: mov     rdx, rsi
0xb30f: mov     ecx, ebx
0xb311: test    bl, bl
0xb313: js      0xb360
0xb315: sub     eax, 1
0xb318: add     edx, 1
0xb31b: mov     dword ptr [rip + 0x1fbe3], eax
0xb321: mov     dword ptr [rip + 0x1fbd9], edx
0xb327: cmp     ebx, -1
0xb32a: je      0xb2ee
0xb32c: add     r12d, 1
0xb330: cmp     r12d, 0xf
0xb334: jne     0xb26c
0xb33a: mov     edi, 1
0xb33f: xor     r15d, r15d
0xb342: call    0x4aa0
0xb347: mov     dword ptr [rip + 0x1f65b], 0
0xb351: jmp     0xb433
0xb433:
0xb356: nop     word ptr cs:[rax + rax]
0xb360: and     ecx, 0x7f
0xb363: add     r12d, 1
0xb367: mov     byte ptr [r13 + rsi], cl
0xb36c: cmp     r12d, 0xf
0xb370: je      0xb33a
0xb372: mov     eax, 0x1b
0xb377: jmp     0xb279
0xb279:
0xb37c: nop     dword ptr [rax]
0xb380: mov     edi, 1
0xb385: call    0x4aa0
0xb38a: jmp     0xb2d3
0xb2d3:
0xb38f: test    r14, r14
0xb392: je      0xb41f

```

```

0xb398: mov     ecx, dword ptr [rip + 0x1fb66]
0xb39e: cmp     r12d, ecx
0xb3a1: jge     0xb41f
0xb3a3: movsxd  rax, dword ptr [rip + 0x1fb56]
0xb3aa: lea     r9d, [r12 + 1]
0xb3af: lea     rdx, [rip + 0x1f74a]
0xb3b6: movsxd  rdi, r9d
0xb3b9: mov     qword ptr [rsp + 0x18], rdi
0xb3be: lea     r8d, [r9 + rax]
0xb3c2: lea     r13, [rax + rdx]
0xb3c6: jmp     0xb3f1
0xb3f1:
0xb3c8: nop     dword ptr [rax + rax]
0xb3d0: sub     ecx, r9d
0xb3d3: add     dword ptr [r14], 1
0xb3d7: add     r13, qword ptr [rsp + 0x18]
0xb3dc: mov     dword ptr [rip + 0x1fb1d], r8d
0xb3e3: add     r8d, r9d
0xb3e6: mov     dword ptr [rip + 0x1fb18], ecx
0xb3ec: cmp     r12d, ecx
0xb3ef: jge     0xb41f
0xb3f1: mov     rdx, rbp
0xb3f4: mov     rsi, r13
0xb3f7: mov     rdi, rbx
0xb3fa: mov     dword ptr [rsp + 0x14], r8d
0xb3ff: mov     dword ptr [rsp + 0x10], r9d
0xb404: mov     dword ptr [rsp + 0xc], ecx
0xb408: call    0x4af0
0xb40d: mov     ecx, dword ptr [rsp + 0xc]
0xb411: mov     r9d, dword ptr [rsp + 0x10]
0xb416: test    eax, eax
0xb418: mov     r8d, dword ptr [rsp + 0x14]
0xb41d: je     0xb3d0
0xb41f: mov     edi, 1
0xb424: call    0x4aa0
0xb429: mov     dword ptr [rip + 0x1f579], 0
0xb433: add     rsp, 0x28
0xb437: mov     rax, r15
0xb43a: pop     rbx
0xb43b: pop     rbp
0xb43c: pop     r12
0xb43e: pop     r13
0xb440: pop     r14
0xb442: pop     r15
0xb444: ret
0xb445: test    eax, eax
0xb447: jne     0xb240
0xb44d: cmp     ebx, 0xa

```

```

0xb450: sete    bpl
0xb454: cmp      ebx, 0xd
0xb457: sete      al
0xb45a: or        ebp, eax
0xb45c: call     0x4f00
0xb461: mov      rax, qword ptr [rax]
0xb464: test     bpl, bpl
0xb467: je       0xb4c6
0xb469: mov      ebx, 0xa
0xb46e: test     byte ptr [rax + 0x15], 0x40
0xb472: je       0xb240
0xb478: mov      ebx, 0xa
0xb47d: mov      rax, qword ptr [rip + 0x1f4dc]
0xb484: lea      r15, [rip + 0x1f4d5]
0xb48b: mov      byte ptr [rax], bl
0xb48d: mov      rax, qword ptr [rip + 0x1f4cc]
0xb494: mov      byte ptr [rax + 1], 0
0xb498: jmp      0xb433
0xb433:
0xb49a: nop      word ptr [rax + rax]
0xb4a0: and      ecx, 0x7f
0xb4a3: mov      byte ptr [rbp + rsi], cl
0xb4a7: test     r14, r14
0xb4aa: je       0xb4b3
0xb4ac: mov      dword ptr [r14], 1
0xb4b3: mov      eax, dword ptr [rip + 0x1ff43]
0xb4b9: mov      ebx, 0x1b
0xb4be: test     eax, eax
0xb4c0: jne      0xb240
0xb4c6: movsxd   rbp, ebx
0xb4c9: add      rbp, rbp
0xb4cc: cmp      ebx, 7
0xb4cf: sete     r12b
0xb4d3: jmp      0xb21f
0xb21f:
0xb4d8: nop      dword ptr [rax + rax]
0xb4e0: endbr64
0xb4e4: mov      eax, dword ptr [rip + 0x1f4c2]
0xb4ea: test     eax, eax
0xb4ec: jne      0xb4f0
0xb4ee: ret
0xb4ef: nop
0xb4f0: sub      rsp, 8
0xb4f4: mov      rdi, qword ptr [rip + 0x1fe65]
0xb4fb: call     0xb080
0xb500: mov      dword ptr [rip + 0x1f49e], 1
0xb50a: add      rsp, 8
0xb50e: ret

```

```

0xb50f: nop
0xb510: endbr64
0xb514: mov     rax, qword ptr [rip + 0x1fe45]
0xb51b: ret
0xb51c: nop     dword ptr [rax]
0xb520: endbr64
0xb524: push    r15
0xb526: mov     esi, 0x5413
0xb52b: mov     edi, 1
0xb530: push    r14
0xb532: push    r13
0xb534: push    r12
0xb536: push    rbp
0xb537: push    rbx
0xb538: sub     rsp, 0x58
0xb53c: mov     rax, qword ptr fs:[0x28]
0xb545: mov     qword ptr [rsp + 0x48], rax
0xb54a: xor     eax, eax
0xb54c: lea     rdx, [rsp + 0x18]
0xb551: mov     dword ptr [rsp + 0x10], 0
0xb559: mov     dword ptr [rsp + 0x14], 0
0xb561: call    0x4a70
0xb566: test    eax, eax
0xb568: jne     0xb5a0
0xb56a: mov     ebx, eax
0xb56c: movzx   eax, word ptr [rsp + 0x1a]
0xb571: test    ax, ax
0xb574: je      0xb790
0xb57a: movzx   ebp, word ptr [rsp + 0x18]
0xb57f: test    bp, bp
0xb582: je      0xb790
0xb588: movzx   ebx, ax
0xb58b: lea     r12d, [rbp - 1]
0xb58f: lea     eax, [rbx - 1]
0xb592: mov     dword ptr [rsp + 0xc], eax
0xb596: jmp     0xb5b2
0xb5b2:
0xb598: nop     dword ptr [rax + rax]
0xb5a0: mov     dword ptr [rsp + 0xc], 0xffffffff
0xb5a8: mov     r12d, 0xffffffff
0xb5ae: xor     ebx, ebx
0xb5b0: xor     ebp, ebp
0xb5b2: lea     r13, [rip + 0x15667]
0xb5b9: lea     r15, [rip + 0x15a13]
0xb5c0: mov     rdi, r13
0xb5c3: call    0x47e0
0xb5c8: mov     rdi, rax
0xb5cb: test    rax, rax

```

```

0xb5ce: je      0xb5df
0xb5d0: lea      rdx, [rsp + 0x14]
0xb5d5: mov      rsi, r15
0xb5d8: xor      eax, eax
0xb5da: call     0x4c60
0xb5df: lea      r14, [rip + 0x15642]
0xb5e6: mov      rdi, r14
0xb5e9: call     0x47e0
0xb5ee: mov      rdi, rax
0xb5f1: test     rax, rax
0xb5f4: je      0xb605
0xb5f6: lea      rdx, [rsp + 0x10]
0xb5fb: mov      rsi, r15
0xb5fe: xor      eax, eax
0xb600: call     0x4c60
0xb605: cmp      dword ptr [rsp + 0xc], 0x3ff
0xb60d: mov      eax, dword ptr [rip + 0x1e8fd]
0xb613: jbe      0xb628
0xb615: mov      ebx, dword ptr [rsp + 0x14]
0xb619: lea      edx, [rbx - 1]
0xb61c: cmp      edx, 0x3ff
0xb622: ja      0xb7c0
0xb628: mov      dword ptr [rip + 0x1f36a], ebx
0xb62e: cmp      r12d, 0x3ff
0xb635: jbe      0xb64a
0xb637: mov      ebp, dword ptr [rsp + 0x10]
0xb63b: lea      edx, [rbp - 1]
0xb63e: cmp      edx, 0x3ff
0xb644: ja      0xb7a8
0xb64a: lea      r12, [rsp + 0x20]
0xb64f: mov      r8d, ebp
0xb652: mov      rcx, r15
0xb655: mov      edx, 0x20
0xb65a: mov      esi, 1
0xb65f: mov      rdi, r12
0xb662: xor      eax, eax
0xb664: mov      dword ptr [rip + 0x1f332], ebp
0xb66a: call     0x4f10
0xb66f: mov      rsi, r12
0xb672: mov      rdi, r14
0xb675: call     0x18380
0xb67a: mov      rcx, r15
0xb67d: mov      edx, 0x20
0xb682: mov      rdi, r12
0xb685: mov      r8d, dword ptr [rip + 0x1f30c]
0xb68c: mov      esi, 1
0xb691: xor      eax, eax
0xb693: call     0x4f10

```

```

0xb698: mov     rdi, r13
0xb69b: mov     rsi, r12
0xb69e: call    0x18380
0xb6a3: mov     rdi, qword ptr [rip + 0x1fc8e]
0xb6aa: test    rdi, rdi
0xb6ad: je      0xb6b4
0xb6af: call    0x18d10
0xb6b4: mov     rdi, qword ptr [rip + 0x1fc75]
0xb6bb: test    rdi, rdi
0xb6be: je      0xb6c5
0xb6c0: call    0x18d10
0xb6c5: mov     rdi, qword ptr [rip + 0x1fc5c]
0xb6cc: test    rdi, rdi
0xb6cf: je      0xb6d6
0xb6d1: call    0x18d10
0xb6d6: mov     rdi, qword ptr [rip + 0x1fc43]
0xb6dd: test    rdi, rdi
0xb6e0: je      0xb6e7
0xb6e2: call    0x18d10
0xb6e7: mov     edi, dword ptr [rip + 0x1f2ab]
0xb6ed: imul    edi, dword ptr [rip + 0x1f2a8]
0xb6f4: mov     esi, 1
0xb6f9: movsxd  rdi, edi
0xb6fc: call    0x1d740
0xb701: mov     edi, dword ptr [rip + 0x1f291]
0xb707: mov     esi, 1
0xb70c: imul    edi, dword ptr [rip + 0x1f289]
0xb713: mov     qword ptr [rip + 0x1fc1e], rax
0xb71a: movsxd  rdi, edi
0xb71d: call    0x1d740
0xb722: mov     edi, dword ptr [rip + 0x1f270]
0xb728: mov     esi, 1
0xb72d: imul    edi, dword ptr [rip + 0x1f268]
0xb734: mov     qword ptr [rip + 0x1fbf5], rax
0xb73b: movsxd  rdi, edi
0xb73e: call    0x1d740
0xb743: mov     edi, dword ptr [rip + 0x1f24f]
0xb749: mov     esi, 1
0xb74e: imul    edi, dword ptr [rip + 0x1f247]
0xb755: mov     qword ptr [rip + 0x1fbcc], rax
0xb75c: movsxd  rdi, edi
0xb75f: call    0x1d740
0xb764: mov     qword ptr [rip + 0x1fbb5], rax
0xb76b: mov     rax, qword ptr [rsp + 0x48]
0xb770: sub     rax, qword ptr fs:[0x28]
0xb779: jne     0xb7dc
0xb77b: add     rsp, 0x58
0xb77f: pop     rbx

```

```

0xb780: pop    rbp
0xb781: pop    r12
0xb783: pop    r13
0xb785: pop    r14
0xb787: pop    r15
0xb789: ret
0xb78a: nop    word ptr [rax + rax]
0xb790: mov     dword ptr [rsp + 0xc], 0xffffffff
0xb798: xor     ebp, ebp
0xb79a: mov     r12d, 0xffffffff
0xb7a0: jmp     0xb5b2
0xb5b2
0xb7a5: nop    dword ptr [rax]
0xb7a8: lea     edx, [rax - 1]
0xb7ab: mov     ebp, 0x18
0xb7b0: cmp     edx, 0x400
0xb7b6: cmovb   ebp, eax
0xb7b9: jmp     0xb64a
0xb64a
0xb7be: nop
0xb7c0: mov     ebx, dword ptr [rip + 0x1e72a]
0xb7c6: lea     edx, [rbx - 1]
0xb7c9: cmp     edx, 0x3ff
0xb7cf: mov     edx, 0x50
0xb7d4: cmova   ebx, edx
0xb7d7: jmp     0xb628
0xb628
0xb7dc: call    0x49d0
0xb7e1: nop    word ptr cs:[rax + rax]
0xb7ec: nop    dword ptr [rax]
0xb7f0: endbr64
0xb7f4: push    r15
0xb7f6: push    r14
0xb7f8: push    r13
0xb7fa: push    r12
0xb7fc: mov     r12, rdi
0xb7ff: mov     edi, 0x5858
0xb804: push    rbp
0xb805: push    rbx
0xb806: mov     ebx, esi
0xb808: sub     rsp, 0x38
0xb80c: mov     rax, qword ptr fs:[0x28]
0xb815: mov     qword ptr [rsp + 0x28], rax
0xb81a: movabs  rax, 0x7363762f7665642f
0xb824: mov     word ptr [rsp + 8], di
0xb829: mov     byte ptr [rsp + 0xa], 0
0xb82e: mov     dword ptr [rsp + 0x18], 0x585861
0xb836: mov     qword ptr [rsp], rax

```

```

0xb83a: mov     qword ptr [rsp + 0x10], rax
0xb83f: cmp     esi, 1
0xb842: je      0xb978
0xb848: mov     esi, dword ptr [rip + 0x1fbb2]
0xb84e: mov     r13, qword ptr [rip + 0x1d783]
0xb855: mov     r14d, 1
0xb85b: test    esi, esi
0xb85d: jne     0xb90c
0xb863: mov     rax, qword ptr [rip + 0x1f126]
0xb86a: lea     r15, [rsp + 0x10]
0xb86f: mov     esi, r14d
0xb872: mov     dword ptr [rip + 0x1fb84], 0
0xb87c: mov     rdi, r15
0xb87f: movzx   edx, byte ptr [rax + 8]
0xb883: mov     byte ptr [rsp + 0x19], dl
0xb887: movzx   eax, byte ptr [rax + 9]
0xb88b: mov     byte ptr [rsp + 0x1a], al
0xb88f: call    0x4930
0xb894: mov     ebp, eax
0xb896: cmp     eax, -1
0xb899: je      0xb8f0
0xb89b: mov     eax, dword ptr [rip + 0x1f0fb]
0xb8a1: imul    eax, dword ptr [rip + 0x1f0f0]
0xb8a8: mov     edi, ebp
0xb8aa: mov     rsi, r12
0xb8ad: lea     edx, [rax + rax + 4]
0xb8b1: xor     eax, eax
0xb8b3: call    r13
0xb8b6: mov     edi, ebp
0xb8b8: call    0x4ab0
0xb8bd: cmp     ebx, 2
0xb8c0: je      0xb990
0xb8c6: mov     eax, 1
0xb8cb: mov     rdx, qword ptr [rsp + 0x28]
0xb8d0: sub     rdx, qword ptr fs:[0x28]
0xb8d9: jne     0xbb07
0xb8df: add     rsp, 0x38
0xb8e3: pop     rbx
0xb8e4: pop     rbp
0xb8e5: pop     r12
0xb8e7: pop     r13
0xb8e9: pop     r14
0xb8eb: pop     r15
0xb8ed: ret
0xb8ee: nop
0xb8f0: mov     ecx, 0x30
0xb8f5: mov     esi, r14d
0xb8f8: mov     rdi, r15

```



```

0xb8fb: mov     word ptr [rsp + 0x19], cx
0xb900: call    0x4930
0xb905: mov     ebp, eax
0xb907: cmp     eax, -1
0xb90a: jne     0xb89b
0xb90c: mov     dword ptr [rip + 0x1faea], 1
0xb916: cmp     ebx, 2
0xb919: je      0xba30
0xb91f: mov     rax, qword ptr [rip + 0x1f06a]
0xb926: mov     r15, rsp
0xb929: mov     esi, r14d
0xb92c: mov     rdi, r15
0xb92f: movzx   edx, byte ptr [rax + 8]
0xb933: movzx   eax, byte ptr [rax + 9]
0xb937: mov     byte ptr [rsp + 8], dl
0xb93b: mov     byte ptr [rsp + 9], al
0xb93f: call    0x4930
0xb944: mov     ebp, eax
0xb946: cmp     eax, -1
0xb949: je      0xba00
0xb94f: mov     edx, dword ptr [rip + 0x1f043]
0xb955: mov     edi, ebp
0xb957: mov     rsi, r12
0xb95a: xor     eax, eax
0xb95c: imul    edx, dword ptr [rip + 0x1f039]
0xb963: add     edx, 4
0xb966: call    r13
0xb969: mov     edi, ebp
0xb96b: call    0x4ab0
0xb970: jmp     0xb8c6
0xb8c6
0xb975: nop     dword ptr [rax]
0xb978: mov     r13, qword ptr [rip + 0x1d661]
0xb97f: xor     r14d, r14d
0xb982: jmp     0xb863
0xb863
0xb987: nop     word ptr [rax + rax]
0xb990: movsx   esi, byte ptr [r12 + 2]
0xb996: movsx   edi, byte ptr [r12 + 3]
0xb99c: call    0xa7a0
0xb9a1: mov     edx, dword ptr [rip + 0x1f561]
0xb9a7: test    edx, edx
0xb9a9: jle     0xb9e5
0xb9ab: xor     ebp, ebp
0xb9ad: lea     rbx, [rip + 0x1f56c]
0xb9b4: jmp     0xb9cc
0xb9cc
0xb9b6: nop     word ptr cs:[rax + rax]

```

```

0xb9c0: mov     edx, dword ptr [rip + 0x1f542]
0xb9c6: add     ebp, eax
0xb9c8: cmp     edx, ebp
0xb9ca: jle     0xb9e5
0xb9cc: sub     edx, ebp
0xb9ce: movsxd  rsi, ebp
0xb9d1: mov     edi, 1
0xb9d6: add     rsi, rbx
0xb9d9: movsxd  rdx, edx
0xb9dc: call    0x18da0
0xb9e1: test    eax, eax
0xb9e3: jns     0xb9c0
0xb9e5: mov     dword ptr [rip + 0x1f519], 0
0xb9ef: mov     eax, 1
0xb9f4: jmp     0xb8cb
0xb8cb
0xb9f9: nop     dword ptr [rax]
0xba00: mov     edx, 0x30
0xba05: mov     esi, r14d
0xba08: mov     rdi, r15
0xba0b: mov     word ptr [rsp + 8], dx
0xba10: call    0x4930
0xba15: mov     ebp, eax
0xba17: cmp     eax, -1
0xba1a: jne     0xb94f
0xba20: xor     eax, eax
0xba22: jmp     0xb8cb
0xb8cb
0xba27: nop     word ptr [rax + rax]
0xba30: xor     eax, eax
0xba32: mov     r15, rsp
0xba35: call    0xa300
0xba3a: mov     rax, qword ptr [rip + 0x1ef4f]
0xba41: mov     esi, r14d
0xba44: mov     rdi, r15
0xba47: movzx   edx, byte ptr [rax + 8]
0xba4b: movzx   eax, byte ptr [rax + 9]
0xba4f: mov     byte ptr [rsp + 8], dl
0xba53: mov     byte ptr [rsp + 9], al
0xba57: call    0x4930
0xba5c: mov     ebp, eax
0xba5e: cmp     eax, -1
0xba61: je      0xbae0
0xba63: mov     edx, dword ptr [rip + 0x1ef2f]
0xba69: mov     rsi, r12
0xba6c: mov     edi, ebp
0xba6e: xor     eax, eax
0xba70: imul    edx, dword ptr [rip + 0x1ef25]

```

```

0xba77: add     edx, 4
0xba7a: call    r13
0xba7d: mov     edi, ebp
0xba7f: call    0x4ab0
0xba84: mov     eax, dword ptr [rip + 0x1ef12]
0xba8a: xor     esi, esi
0xba8c: lea     edi, [rax - 1]
0xba8f: call    0xa7a0
0xba94: mov     edx, dword ptr [rip + 0x1f46e]
0xba9a: test    edx, edx
0xba9c: jle     0xb9e5
0xbaa2: xor     ebp, ebp
0xbaa4: lea     rbx, [rip + 0x1f475]
0xbaab: jmp     0xbac0
0xbac0:
0xbaad: nop     dword ptr [rax]
0xbab0: mov     edx, dword ptr [rip + 0x1f452]
0xbab6: add     ebp, eax
0xbab8: cmp     edx, ebp
0xbaba: jle     0xb9e5
0xbac0: sub     edx, ebp
0xbac2: movsxd  rsi, ebp
0xbac5: mov     edi, 1
0xbaca: add     rsi, rbx
0xbacd: movsxd  rdx, edx
0xbad0: call    0x18da0
0xbad5: test    eax, eax
0xbad7: jns     0xbab0
0xbad9: jmp     0xb9e5
0xb9e5:
0xbade: nop
0xbae0: mov     eax, 0x30
0xbae5: mov     esi, r14d
0xbae8: mov     rdi, r15
0xbaeb: mov     word ptr [rsp + 8], ax
0xbaf0: call    0x4930
0xbaf5: mov     ebp, eax
0xbaf7: cmp     eax, -1
0xbafa: jne     0xba63
0xbb00: xor     eax, eax
0xbb02: jmp     0xb8cb
0xb8cb:
0xbb07: call    0x49d0
0xbb0c: nop     dword ptr [rax]
0xbb10: endbr64
0xbb14: mov     eax, dword ptr [rip + 0x1ee6e]
0xbb1a: test    eax, eax
0xbb1c: jne     0xbb20

```

```

0xbb1e: ret
0xbb1f: nop
0xbb20: sub     rsp, 8
0xbb24: mov     esi, 1
0xbb29: xor     eax, eax
0xbb2b: call    0xb7f0
0xbb30: mov     dword ptr [rip + 0x1f8ce], eax
0xbb36: add     rsp, 8
0xbb3a: ret
0xbb3b: nop     dword ptr [rax + rax]
0xbb40: endbr64
0xbb44: push    rbp
0xbb45: xor     eax, eax
0xbb47: mov     rbp, rdi
0xbb4a: call    0xa6c0
0xbb4f: mov     edx, dword ptr [rip + 0x1ee33]
0xbb55: test    edx, edx
0xbb57: je      0xbb94
0xbb59: mov     eax, dword ptr [rip + 0x1f8a5]
0xbb5f: test    eax, eax
0xbb61: je      0xbb94
0xbb63: mov     edx, dword ptr [rip + 0x1ee33]
0xbb69: imul    edx, dword ptr [rip + 0x1ee28]
0xbb70: xor     esi, esi
0xbb72: mov     rdi, qword ptr [rip + 0x1f7af]
0xbb79: movsxd  rdx, edx
0xbb7c: call    0x4a50
0xbb81: xor     eax, eax
0xbb83: mov     esi, 2
0xbb88: mov     rdi, rbp
0xbb8b: call    0xb7f0
0xbb90: test    eax, eax
0xbb92: jne     0xbba0
0xbb94: xor     eax, eax
0xbb96: pop     rbp
0xbb97: jmp     0xa300
0xa300
0xbb9c: nop     dword ptr [rax]
0xbba0: mov     edx, dword ptr [rip + 0x1edf6]
0xbba6: imul    edx, dword ptr [rip + 0x1edeb]
0xbbad: xor     esi, esi
0xbbae: mov     rdi, qword ptr [rip + 0x1f782]
0xbbbb: pop     rbp
0xbbbb7: movsxd  rdx, edx
0xbbbbba: jmp     0x4a50
0x4a50
0xbbbbff: nop
0xbbbc0: endbr64

```

```

0xbbc4: push    rbp
0xbbc5: mov     rbp, rdi
0xbbc8: push    rbx
0xbbc9: sub     rsp, 8
0xbbcd: cmp     dword ptr [rip + 0x1ed84], 1
0xbbdd4: je      0xbcd0
0xbbdda: xor     eax, eax
0xbbddc: call    0xa6c0
0xbbe1: test    rbp, rbp
0xbbe4: je      0xbbf4
0xbbe6: mov     eax, dword ptr [rip + 0x1ed9c]
0xbbec: test    eax, eax
0xbbee: jne     0xbcc0
0xbbf4: mov     eax, dword ptr [rip + 0x1eda2]
0xbbfa: mov     rdi, qword ptr [rip + 0x1e347]
0xbc01: lea     rdx, [rip - 0x2198]
0xbc08: lea     esi, [rax - 1]
0xbc0b: call    0x4820
0xbc10: mov     edx, dword ptr [rip + 0x1f2f2]
0xbc16: test    edx, edx
0xbc18: jle     0xbc4d
0xbc1a: xor     ebp, ebp
0xbc1c: lea     rbx, [rip + 0x1f2fd]
0xbc23: jmp     0xbc34
0xbc34:
0xbc25: nop     dword ptr [rax]
0xbc28: mov     edx, dword ptr [rip + 0x1f2da]
0xbc2e: add     ebp, eax
0xbc30: cmp     edx, ebp
0xbc32: jle     0xbc4d
0xbc34: sub     edx, ebp
0xbc36: movsxd  rsi, ebp
0xbc39: mov     edi, 1
0xbc3e: add     rsi, rbx
0xbc41: movsxd  rdx, edx
0xbc44: call    0x18da0
0xbc49: test    eax, eax
0xbc4b: jns     0xbc28
0xbc4d: mov     edi, dword ptr [rip + 0x1ed49]
0xbc53: xor     esi, esi
0xbc55: mov     dword ptr [rip + 0x1f2a9], 0
0xbc5f: call    0xa7a0
0xbc64: mov     edx, dword ptr [rip + 0x1f29e]
0xbc6a: test    edx, edx
0xbc6c: jle     0xbca5
0xbc6e: xor     ebp, ebp
0xbc70: lea     rbx, [rip + 0x1f2a9]
0xbc77: jmp     0xbc8c

```

```

0xbc8c
0xbc79: nop      dword ptr [rax]
0xbc80: mov      edx, dword ptr [rip + 0x1f282]
0xbc86: add      ebp, eax
0xbc88: cmp      edx, ebp
0xbc8a: jle      0xbca5
0xbc8c: sub      edx, ebp
0xbc8e: movsxd   rsi, ebp
0xbc91: mov      edi, 1
0xbc96: add      rsi, rbx
0xbc99: movsxd   rdx, edx
0xbc9c: call     0x18da0
0xbca1: test     eax, eax
0xbca3: jns      0xbc80
0xbca5: mov      dword ptr [rip + 0x1f259], 0
0xbcaf: add      rsp, 8
0xbcb3: mov      edi, 0xa
0xbcb8: pop      rbx
0xbcb9: pop      rbp
0xbcba: jmp      0x4810
0x4810
0xbcbf: nop
0xbcc0: mov      rdi, rbp
0xbcc3: call     0xbb40
0xbcc8: jmp      0xbbf4
0xbbf4
0xbccd: nop      dword ptr [rax]
0xbcd0: lea      rdx, [rip + 0x1f6e9]
0xbcd7: mov      esi, 1
0xbcdc: mov      edi, 1
0xbce1: call     0x4d70
0xbce6: mov      dword ptr [rip + 0x1ec68], 0
0xbcf0: jmp      0xbbda
0xbbda
0xbcf5: nop      word ptr cs:[rax + rax]
0xbd00: endbr64
0xbd04: push     r13
0xbd06: lea      r13, [rip + 0x1cc33]
0xbd0d: push     r12
0xbd0f: push     rbp
0xbd10: mov      rbp, rdi
0xbd13: lea      rdi, [rip + 0x14f14]
0xbd1a: push     rbx
0xbd1b: xor      ebx, ebx
0xbd1d: sub      rsp, 8
0xbd21: jmp      0xbd2d
0xbd2d
0xbd23: nop      dword ptr [rax + rax]

```

```

0xbd28: mov     rdi, qword ptr [r13 + rbx*8]
0xbd2d: mov     rsi, rbp
0xbd30: mov     r12d, ebx
0xbd33: call    0x4b30
0xbd38: test    eax, eax
0xbd3a: je      0xbd60
0xbd3c: add     rbx, 1
0xbd40: cmp     rbx, 0xa
0xbd44: jne     0xbd28
0xbd46: add     rsp, 8
0xbd4a: mov     r12d, 0xffffffff
0xbd50: pop     rbx
0xbd51: mov     eax, r12d
0xbd54: pop     rbp
0xbd55: pop     r12
0xbd57: pop     r13
0xbd59: ret
0xbd5a: nop     word ptr [rax + rax]
0xbd60: lea     eax, [rbx - 8]
0xbd63: cmp     ebx, 7
0xbd66: cmovg   r12d, eax
0xbd6a: add     rsp, 8
0xbd6e: pop     rbx
0xbd6f: pop     rbp
0xbd70: mov     eax, r12d
0xbd73: pop     r12
0xbd75: pop     r13
0xbd77: ret
0xbd78: nop     dword ptr [rax + rax]
0xbd80: endbr64
0xbd84: push    r12
0xbd86: mov     r12, rdi
0xbd89: push    rbp
0xbd8a: lea     rbp, [rip + 0x1e1e7]
0xbd91: push    rbx
0xbd92: mov     ebx, 0x11
0xbd97: jmp     0xbdac
0xbdac:
0xbd99: nop     dword ptr [rax]
0xbda0: add     ebx, 1
0xbda3: add     rbp, 0x20
0xbda7: cmp     ebx, 0x26
0xbdaa: je      0xbdd8
0xbdac: mov     rdi, qword ptr [rbp]
0xbdb0: mov     rsi, r12
0xbdb3: call    0x4b30
0xbdb8: test    eax, eax
0xbdba: jne     0xbda0

```

```

0xbdbc: movsxd  rbx, ebx
0xbdbf: lea      rax, [rip + 0x1df7a]
0xbdc6: shl      rbx, 5
0xbdca: mov      rax, qword ptr [rax + rbx + 8]
0xbdcf: pop      rbx
0xbdd0: pop      rbp
0xbdd1: pop      r12
0xbdd3: ret
0xbdd4: nop      dword ptr [rax]
0xbdd8: pop      rbx
0xbdd9: xor      eax, eax
0xbddb: pop      rbp
0xbddc: pop      r12
0xbdde: ret
0xbddf: nop
0xbde0: endbr64
0xbde4: push    r14
0xbde6: lea      rdi, [rip + 0x1479d]
0xbded: push    r13
0xbdef: push    r12
0xbdf1: push    rbp
0xbdf2: push    rbx
0xbdf3: sub      rsp, 0xb0
0xbdfa: mov      rax, qword ptr fs:[0x28]
0xbe03: mov      qword ptr [rsp + 0xa8], rax
0xbe0b: xor      eax, eax
0xbe0d: call     0x47e0
0xbe12: lea      rsi, [rsp + 0x10]
0xbe17: mov      edi, 1
0xbe1c: mov      rbp, rax
0xbe1f: call     0x4eb0
0xbe24: mov      rax, qword ptr [rsp + 0x38]
0xbe29: shr      rax, 8
0xbe2d: cmp      rax, 4
0xbe31: sete     al
0xbe34: movzx    eax, al
0xbe37: mov      dword ptr [rip + 0x1eb4b], eax
0xbe3d: test     rbp, rbp
0xbe40: je       0xc36d
0xbe46: mov      rdi, rbp
0xbe49: call     0x49b0
0xbe4e: cmp      rax, 0x3f
0xbe52: ja       0xc308
0xbe58: lea      r12, [rip + 0x14ff1]
0xbe5f: lea      rbx, [rip + 0x14d6f]
0xbe66: lea      r14, [rip + 0x14e02]
0xbe6d: lea      r13, [rip + 0x14e07]
0xbe74: nop      dword ptr [rax]

```



```

0xbe78: xor     edi, edi
0xbe7a: mov     rsi, rbp
0xbe7d: call    0x4e90
0xbe82: cmp     eax, -1
0xbe85: je      0xbf38
0xbe8b: test    eax, eax
0xbe8d: jne     0xbff8
0xbe93: mov     rdi, qword ptr [rip + 0x1e9e6]
0xbe9a: mov     esi, 1
0xbe9f: mov     r8, rbp
0xbea2: mov     rdx, r12
0xbea5: mov     rcx, qword ptr [rip + 0x1ea3c]
0xbeac: lea     r9, [rip + 0x1de5d]
0xeb3: call    0x4e60
0xeb8: mov     rsi, rbx
0xebb: mov     rdi, rbp
0xebe: call    0x4b30
0xbec3: test    eax, eax
0xbec5: je      0xbf98
0xbecb: mov     rdi, rbp
0xbece: call    0x9940
0xbed3: test    eax, eax
0xbed5: jne     0xbfc8
0xbedb: mov     rsi, r14
0xbede: mov     rdi, rbp
0xbee1: call    0x4b30
0xbee6: test    eax, eax
0xbee8: je      0xbf01
0xbeea: lea     rsi, [rip + 0x14d84]
0xbef1: mov     rdi, rbp
0xbef4: call    0x4b30
0xbef9: test    eax, eax
0xbefb: jne     0xc3b4
0xbf01: mov     rdi, qword ptr [rip + 0x1e978]
0xbf08: mov     rcx, qword ptr [rip + 0x1e9d9]
0xbf0f: mov     esi, 1
0xbf14: xor     eax, eax
0xbf16: mov     rdx, r13
0xbf19: lea     rbp, [rip + 0x14cba]
0xbf20: call    0x4e60
0xbf25: xor     edi, edi
0xbf27: mov     rsi, rbp
0xbf2a: call    0x4e90
0xbf2f: cmp     eax, -1
0xbf32: jne     0xbe8b
0xbf38: mov     rcx, qword ptr [rip + 0x1e9a9]
0xbf3f: mov     rdi, qword ptr [rip + 0x1e93a]
0xbf46: xor     eax, eax

```

```

0xbf48: lea      r8, [rip + 0x1ddc1]
0xbf4f: lea      rdx, [rip + 0x14ea2]
0xbf56: mov      esi, 1
0xbf5b: call     0x4e60
0xbf60: mov      rdi, qword ptr [rip + 0x1e919]
0xbf67: xor      eax, eax
0xbf69: mov      rcx, qword ptr [rip + 0x1e978]
0xbf70: lea      r8, [rip + 0x1dd89]
0xbf77: lea      rdx, [rip + 0x14ea2]
0xbf7e: mov      esi, 1
0xbf83: call     0x4e60
0xbf88: mov      edi, 1
0xbf8d: call     0x4e20
0xbf92: nop      word ptr [rax + rax]
0xbf98: mov      rcx, qword ptr [rip + 0x1e949]
0xbf9f: mov      rdi, qword ptr [rip + 0x1e8da]
0xbfa6: lea      rdx, [rip + 0x14c99]
0xbfad: mov      esi, 1
0xbfb2: lea      rbp, [rip + 0x14c21]
0xbfb9: call     0x4e60
0xbfbe: jmp      0xbe78
0xbe78:
0xbfc3: nop      dword ptr [rax + rax]
0xbfc8: mov      rcx, qword ptr [rip + 0x1e919]
0xbfcf: mov      rdi, qword ptr [rip + 0x1e8aa]
0xbfd6: lea      rdx, [rip + 0x14c7d]
0xbfdd: xor      eax, eax
0xbfdf: mov      esi, 1
0xbfe4: lea      rbp, [rip + 0x14bf6]
0xbfeb: call     0x4e60
0xbff0: jmp      0xbe78
0xbe78:
0xbff5: nop      dword ptr [rax]
0xbff8: mov      rdi, rbp
0xbffb: lea      r13, [rsp + 8]
0xc000: call     0x1d7a0
0xc005: mov      edi, 0x800
0xc00a: mov      qword ptr [rip + 0x1e93f], rax
0xc011: call     0x1d670
0xc016: mov      rdi, qword ptr [rip + 0x1de03]
0xc01d: mov      rsi, r13
0xc020: mov      qword ptr [rsp + 8], rax
0xc025: call     0x4ca0
0xc02a: xor      edx, edx
0xc02c: test     rax, rax
0xc02f: je       0xc034
0xc031: movzx    edx, byte ptr [rax]
0xc034: mov      rdi, qword ptr [rip + 0x1de25]

```

```

0xc03b: mov     rsi, r13
0xc03e: mov     byte ptr [rip + 0x1e904], dl
0xc044: call    0x4ca0
0xc049: mov     rdi, qword ptr [rip + 0x1ddf0]
0xc050: mov     rsi, r13
0xc053: mov     qword ptr [rip + 0x1e8e6], rax
0xc05a: call    0x4ca0
0xc05f: cmp     qword ptr [rip + 0x1e8d9], 0
0xc067: mov     qword ptr [rip + 0x1e8ca], rax
0xc06e: je      0xc2c0
0xc074: test    rax, rax
0xc077: je      0xc2c0
0xc07d: mov     rdi, qword ptr [rip + 0x1dcbc]
0xc084: mov     rsi, r13
0xc087: call    0x4ca0
0xc08c: mov     rdi, qword ptr [rip + 0x1dced]
0xc093: mov     rsi, r13
0xc096: mov     qword ptr [rip + 0x1dcab], rax
0xc09d: call    0x4ca0
0xc0a2: mov     rdi, qword ptr [rip + 0x1dcb7]
0xc0a9: mov     rsi, r13
0xc0ac: mov     qword ptr [rip + 0x1dcd5], rax
0xc0b3: call    0x4ca0
0xc0b8: cmp     qword ptr [rip + 0x1dc88], 0
0xc0c0: mov     qword ptr [rip + 0x1dca1], rax
0xc0c7: je      0xc352
0xc0cd: mov     rdi, qword ptr [rip + 0x1ddac]
0xc0d4: mov     rsi, r13
0xc0d7: call    0x4ca0
0xc0dc: xor     edx, edx
0xc0de: mov     qword ptr [rip + 0x1dda3], rax
0xc0e5: test    rax, rax
0xc0e8: je      0xc0ff
0xc0ea: mov     rdi, qword ptr [rip + 0x1ddcf]
0xc0f1: mov     qword ptr [rip + 0x1dc70], rax
0xc0f8: call    0x4c90
0xc0fd: mov     edx, eax
0xc0ff: mov     rdi, qword ptr [rip + 0x1dd9a]
0xc106: mov     dword ptr [rip + 0x1ddc4], edx
0xc10c: call    0x4c50
0xc111: mov     dword ptr [rip + 0x1dd99], eax
0xc117: test    eax, eax
0xc119: js      0xc13c
0xc11b: mov     qword ptr [rip + 0x1dc62], 0
0xc126: mov     qword ptr [rip + 0x1dc37], 0
0xc131: mov     qword ptr [rip + 0x1dc0c], 0
0xc13c: mov     rdi, qword ptr [rip + 0x1dd9d]
0xc143: call    0x4c50

```

```

0xc148: mov     rdi, qword ptr [rip + 0x1ddb1]
0xc14f: mov     dword ptr [rip + 0x1dd9b], eax
0xc155: call    0x4c50
0xc15a: mov     rdi, qword ptr [rip + 0x1dc3f]
0xc161: mov     rsi, r13
0xc164: mov     dword ptr [rip + 0x1dda6], eax
0xc16a: call    0x4ca0
0xc16f: mov     rdi, qword ptr [rip + 0x1dc4a]
0xc176: mov     rsi, r13
0xc179: mov     qword ptr [rip + 0x1dc28], rax
0xc180: call    0x4ca0
0xc185: cmp     qword ptr [rip + 0x1dc1b], 0
0xc18d: mov     qword ptr [rip + 0x1dc34], rax
0xc194: je      0xc2a0
0xc19a: test    rax, rax
0xc19d: je      0xc2a0
0xc1a3: mov     rdi, qword ptr [rip + 0x1dc36]
0xc1aa: mov     rsi, r13
0xc1ad: lea     rbp, [rip + 0x1ddac]
0xc1b4: lea     rbx, [rbp - 0x220]
0xc1bb: call    0x4ca0
0xc1c0: mov     rdi, qword ptr [rip + 0x1dc39]
0xc1c7: mov     rsi, r13
0xc1ca: lea     r12, [rbx + 0x4c0]
0xc1d1: mov     qword ptr [rip + 0x1dc10], rax
0xc1d8: call    0x4ca0
0xc1dd: mov     rdi, qword ptr [rip + 0x1dd3c]
0xc1e4: mov     rsi, r13
0xc1e7: mov     qword ptr [rip + 0x1dc1a], rax
0xc1ee: call    0x4ca0
0xc1f3: mov     rdi, qword ptr [rip + 0x1dd46]
0xc1fa: mov     rsi, r13
0xc1fd: mov     qword ptr [rip + 0x1dd24], rax
0xc204: call    0x4ca0
0xc209: mov     qword ptr [rip + 0x1dd38], rax
0xc210: mov     rdi, qword ptr [rbp]
0xc214: mov     rsi, r13
0xc217: add     rbp, 0x20
0xc21b: call    0x4ca0
0xc220: mov     qword ptr [rbp - 0x18], rax
0xc224: cmp     rbp, r12
0xc227: jne     0xc210
0xc229: xor     ebp, ebp
0xc22b: lea     r13, [rip + 0x161b3]
0xc232: lea     r14, [rip + 0x14c57]
0xc239: jmp     0xc249
0xc249:
0xc23b: nop     dword ptr [rax + rax]

```

```

0xc240: add    rbx, 0x20
0xc244: cmp    rbx, r12
0xc247: je     0xc268
0xc249: cmp    qword ptr [rbx + 8], 0
0xc24e: jne    0xc240
0xc250: mov    eax, dword ptr [rbx + 0x14]
0xc253: test   eax, eax
0xc255: jne    0xc2e0
0xc25b: mov    qword ptr [rbx + 8], r13
0xc25f: add    rbx, 0x20
0xc263: cmp    rbx, r12
0xc266: jne    0xc249
0xc268: test   ebp, ebp
0xc26a: jne    0xc390
0xc270: mov    rax, qword ptr [rsp + 0xa8]
0xc278: sub    rax, qword ptr fs:[0x28]
0xc281: jne    0xc3be
0xc287: add    rsp, 0xb0
0xc28e: pop    rbx
0xc28f: pop    rbp
0xc290: pop    r12
0xc292: pop    r13
0xc294: pop    r14
0xc296: ret
0xc297: nop    word ptr [rax + rax]
0xc2a0: mov    qword ptr [rip + 0x1dafd], 0
0xc2ab: mov    qword ptr [rip + 0x1db12], 0
0xc2b6: jmp    0xc1a3
0xc1a3
0xc2bb: nop    dword ptr [rax + rax]
0xc2c0: mov    qword ptr [rip + 0x1e66d], 0
0xc2cb: mov    qword ptr [rip + 0x1e66a], 0
0xc2d6: jmp    0xc07d
0xc07d
0xc2db: nop    dword ptr [rax + rax]
0xc2e0: mov    r8, qword ptr [rbx]
0xc2e3: mov    rcx, qword ptr [rip + 0x1e5fe]
0xc2ea: mov    rdx, r14
0xc2ed: xor    eax, eax
0xc2ef: mov    rdi, qword ptr [rip + 0x1e58a]
0xc2f6: mov    esi, 1
0xc2fb: add    ebp, 1
0xc2fe: call  0x4e60
0xc303: jmp    0xc240
0xc240
0xc308: mov    rcx, qword ptr [rip + 0x1e5d9]
0xc30f: mov    rdi, qword ptr [rip + 0x1e56a]
0xc316: lea    rdx, [rip + 0x14aab]

```

```

0xc31d: xor     eax, eax
0xc31f: mov     esi, 1
0xc324: call    0x4e60
0xc329: mov     rcx, qword ptr [rip + 0x1e550]
0xc330: mov     edx, 0x11
0xc335: mov     esi, 1
0xc33a: lea     rdi, [rip + 0x148f3]
0xc341: lea     rbp, [rip + 0x1d9d1]
0xc348: call    0x4e40
0xc34d: jmp     0xbe58
0xbe58:
0xc352: mov     qword ptr [rip + 0x1da2b], 0
0xc35d: mov     qword ptr [rip + 0x1da00], 0
0xc368: jmp     0xc0cd
0xc0cd:
0xc36d: mov     rcx, qword ptr [rip + 0x1e574]
0xc374: mov     rdi, qword ptr [rip + 0x1e505]
0xc37b: lea     rdx, [rip + 0x14a16]
0xc382: xor     eax, eax
0xc384: mov     esi, 1
0xc389: call    0x4e60
0xc38e: jmp     0xc329
0xc329:
0xc390: mov     rcx, qword ptr [rip + 0x1e551]
0xc397: mov     rdi, qword ptr [rip + 0x1e4e2]
0xc39e: mov     r8d, ebp
0xc3a1: xor     eax, eax
0xc3a3: lea     rdx, [rip + 0x14b16]
0xc3aa: mov     esi, 1
0xc3af: call    0x4e60
0xc3b4: mov     edi, 1
0xc3b9: call    0x4e20
0xc3be: call    0x49d0
0xc3c3: nop     word ptr cs:[rax + rax]
0xc3ce: nop
0xc3d0: endbr64
0xc3d4: push    rbx
0xc3d5: mov     ebx, edi
0xc3d7: xor     edi, edi
0xc3d9: call    0x48f0
0xc3de: test    eax, eax
0xc3e0: je      0xc453
0xc3e2: mov     edi, 1
0xc3e7: call    0x48f0
0xc3ec: test    eax, eax
0xc3ee: je      0xc453
0xc3f0: mov     edi, 1
0xc3f5: call    0x4cd0

```

```

0xc3fa: mov     qword ptr [rip + 0x1e58f], rax
0xc401: test    rax, rax
0xc404: je      0xc47e
0xc406: lea     rsi, [rip + 0x1efb3]
0xc40d: mov     edi, 1
0xc412: call    0x4d60
0xc417: mov     edi, 0x40
0xc41c: call    0x1d670
0xc421: mov     dword ptr [rip + 0x1efd5], ebx
0xc427: pop     rbx
0xc428: mov     qword ptr [rip + 0x1ef31], rax
0xc42f: mov     qword ptr [rip + 0x1e52a], rax
0xc436: xor     eax, eax
0xc438: mov     qword ptr [rip + 0x1e52d], 0
0xc443: mov     qword ptr [rip + 0x1e51a], 0
0xc44e: jmp     0xbde0
0xbde0
0xc453: mov     rdi, qword ptr [rip + 0x1e426]
0xc45a: mov     rcx, qword ptr [rip + 0x1e487]
0xc461: lea     rdx, [rip + 0x14a88]
0xc468: xor     eax, eax
0xc46a: mov     esi, 1
0xc46f: call    0x4e60
0xc474: mov     edi, 1
0xc479: call    0x4e20
0xc47e: mov     rdi, qword ptr [rip + 0x1e3fb]
0xc485: mov     rcx, qword ptr [rip + 0x1e45c]
0xc48c: mov     esi, 1
0xc491: lea     rdx, [rip + 0x147f8]
0xc498: call    0x4e60
0xc49d: mov     edi, 1
0xc4a2: call    0x4e20
0xc4a7: nop     word ptr [rax + rax]
0xc4b0: endbr64
0xc4b4: push    r12
0xc4b6: push    rbp
0xc4b7: mov     rbp, rdi
0xc4ba: mov     rdi, qword ptr [rip + 0x1e48f]
0xc4c1: push    rbx
0xc4c2: call    0x9940
0xc4c7: test    eax, eax
0xc4c9: jne     0xc4d0
0xc4cb: pop     rbx
0xc4cc: pop     rbp
0xc4cd: pop     r12
0xc4cf: ret
0xc4d0: mov     rdi, rbp
0xc4d3: call    0x49b0

```

```

0xc4d8: lea    rdi, [rax + 0x81]
0xc4df: mov     rbx, rax
0xc4e2: call    0x1d670
0xc4e7: mov     rdi, rbp
0xc4ea: mov     r12, rax
0xc4ed: call    0x1d7a0
0xc4f2: mov     rsi, rbx
0xc4f5: add     ebx, 0x7f
0xc4f8: mov     rbp, rax
0xc4fb: mov     rdi, rax
0xc4fe: call    0x184d0
0xc503: push    7
0xc505: lea     rcx, [rip + 0x147a2]
0xc50c: xor     eax, eax
0xc50e: push    rbp
0xc50f: mov     rdx, -1
0xc516: mov     esi, 1
0xc51b: mov     rdi, r12
0xc51e: lea     r9, [rip + 0x14040]
0xc525: mov     r8d, 0x1b
0xc52b: call    0x4f10
0xc530: mov     eax, dword ptr [rip + 0x1e462]
0xc536: pop     rdx
0xc537: pop     rcx
0xc538: add     eax, eax
0xc53a: cmp     ebx, eax
0xc53c: jl      0xc555
0xc53e: cdqe
0xc540: mov     byte ptr [r12 + rax], 7
0xc545: mov     eax, dword ptr [rip + 0x1e44d]
0xc54b: add     eax, eax
0xc54d: cdqe
0xc54f: mov     byte ptr [r12 + rax + 1], 0
0xc555: mov     rdi, r12
0xc558: call    0x49b0
0xc55d: mov     rsi, r12
0xc560: mov     edi, 1
0xc565: mov     rdx, rax
0xc568: call    0x18da0
0xc56d: mov     rdi, rbp
0xc570: call    0x18d10
0xc575: mov     rdi, r12
0xc578: call    0x18d10
0xc57d: mov     rdi, qword ptr [rip + 0x1e2bc]
0xc584: pop     rbx
0xc585: pop     rbp
0xc586: pop     r12
0xc588: jmp     0x4c40

```



```

0x4c40
0xc58d: nop      dword ptr [rax]
0xc590: endbr64
0xc594: sub      rsp, 8
0xc598: mov      edi, 0x18
0xc59d: call     0x1d670
0xc5a2: mov      qword ptr [rax], 0
0xc5a9: mov      qword ptr [rax + 8], 0
0xc5b1: add      rsp, 8
0xc5b5: ret
0xc5b6: nop      word ptr cs:[rax + rax]
0xc5c0: endbr64
0xc5c4: test     rdi, rdi
0xc5c7: je       0xc5d0
0xc5c9: jmp      0x18d10
0x18d10
0xc5ce: nop
0xc5d0: ret
0xc5d1: nop      word ptr cs:[rax + rax]
0xc5dc: nop      dword ptr [rax]
0xc5e0: endbr64
0xc5e4: mov      dword ptr [rdi], esi
0xc5e6: mov      dword ptr [rdi + 4], edx
0xc5e9: mov      dword ptr [rdi + 8], ecx
0xc5ec: mov      dword ptr [rdi + 0xc], r8d
0xc5f0: ret
0xc5f1: nop      word ptr cs:[rax + rax]
0xc5fc: nop      dword ptr [rax]
0xc600: endbr64
0xc604: mov      rax, rdi
0xc607: mov      rdi, rsi
0xc60a: mov      esi, edx
0xc60c: mov      ecx, dword ptr [rax + 0x10]
0xc60f: lea      r8d, [rcx + rdx]
0xc613: mov      edx, dword ptr [rax + 0xc]
0xc616: mov      dword ptr [rax + 0x10], r8d
0xc61a: cmp      edx, ecx
0xc61c: jle      0xc640
0xc61e: mov      r9d, dword ptr [rax + 8]
0xc622: cmp      dword ptr [rax + 0x14], r9d
0xc626: jge      0xc640
0xc628: cmp      r8d, edx
0xc62b: jle      0xc631
0xc62d: sub      edx, ecx
0xc62f: mov      esi, edx
0xc631: jmp      0xa1a0
0xa1a0
0xc636: nop      word ptr cs:[rax + rax]

```

```

0xc640: xor      eax, eax
0xc642: ret
0xc643: nop      word ptr cs:[rax + rax]
0xc64e: nop
0xc650: endbr64
0xc654: mov      ecx, dword ptr [rdi + 0x10]
0xc657: lea      edx, [rcx + 1]
0xc65a: mov      dword ptr [rdi + 0x10], edx
0xc65d: cmp      edx, dword ptr [rdi + 0xc]
0xc660: jg       0xc66a
0xc662: mov      ecx, dword ptr [rdi + 8]
0xc665: cmp      dword ptr [rdi + 0x14], ecx
0xc668: jl       0xc670
0xc66a: xor      eax, eax
0xc66c: ret
0xc66d: nop      dword ptr [rax]
0xc670: mov      edi, esi
0xc672: jmp      0xa230
0xa230:
0xc677: nop      word ptr [rax + rax]
0xc680: endbr64
0xc684: mov      rax, rdi
0xc687: mov      edi, esi
0xc689: mov      dword ptr [rax + 0x10], edx
0xc68c: add      edx, dword ptr [rax]
0xc68e: mov      dword ptr [rax + 0x14], esi
0xc691: add      edi, dword ptr [rax + 4]
0xc694: mov      esi, edx
0xc696: jmp      0xa4b0
0xa4b0:
0xc69b: nop      dword ptr [rax + rax]
0xc6a0: endbr64
0xc6a4: mov      eax, dword ptr [rdi]
0xc6a6: ret
0xc6a7: nop      word ptr [rax + rax]
0xc6b0: endbr64
0xc6b4: mov      eax, dword ptr [rdi + 4]
0xc6b7: ret
0xc6b8: nop      dword ptr [rax + rax]
0xc6c0: endbr64
0xc6c4: mov      eax, dword ptr [rdi + 8]
0xc6c7: ret
0xc6c8: nop      dword ptr [rax + rax]
0xc6d0: endbr64
0xc6d4: mov      eax, dword ptr [rdi + 0xc]
0xc6d7: ret
0xc6d8: nop      dword ptr [rax + rax]
0xc6e0: push     r14

```

```

0xc6e2: push    r13
0xc6e4: push    r12
0xc6e6: mov     r12, qword ptr [rip + 0x1ed1b]
0xc6ed: push    rbp
0xc6ee: push    rbx
0xc6ef: mov     r13, qword ptr [r12 + 0x20]
0xc6f4: mov     rbx, qword ptr [r12 + 0x18]
0xc6f9: cmp     r13, rbx
0xc6fc: je      0xc761
0xc6fe: mov     ebp, edi
0xc700: jbe     0xc770
0xc702: test    bpl, 1
0xc706: je      0xc75c
0xc708: mov     rdi, qword ptr [r12 + 0x60]
0xc70d: test    rdi, rdi
0xc710: je      0xc71e
0xc712: call    0x18d10
0xc717: mov     r12, qword ptr [rip + 0x1ecea]
0xc71e: mov     r14, r13
0xc721: sub     r14, rbx
0xc724: lea     rdi, [r14 + 1]
0xc728: call    0x1d670
0xc72d: mov     rdx, r14
0xc730: mov     qword ptr [r12 + 0x60], rax
0xc735: mov     rax, qword ptr [rip + 0x1eccc]
0xc73c: mov     rsi, qword ptr [rax + 0x58]
0xc740: mov     rdi, qword ptr [rax + 0x60]
0xc744: add     rsi, rbx
0xc747: call    0x4bb0
0xc74c: mov     rax, qword ptr [rip + 0x1ecb5]
0xc753: mov     rax, qword ptr [rax + 0x60]
0xc757: mov     byte ptr [rax + r14], 0
0xc75c: and     ebp, 2
0xc75f: jne     0xc788
0xc761: pop     rbx
0xc762: pop     rbp
0xc763: pop     r12
0xc765: pop     r13
0xc767: pop     r14
0xc769: ret
0xc76a: nop     word ptr [rax + rax]
0xc770: mov     rax, rbx
0xc773: mov     rbx, r13
0xc776: mov     r13, rax
0xc779: test    bpl, 1
0xc77d: je      0xc75c
0xc77f: jmp     0xc708
0xc708:

```

```

0xc781: nop        dword ptr [rax]
0xc788: mov     rax, qword ptr [rip + 0x1ec79]
0xc78f: mov     rdx, rbx
0xc792: sub     rdx, r13
0xc795: add     qword ptr [rax + 0x48], rdx
0xc799: mov     rdi, qword ptr [rax + 0x58]
0xc79d: add     qword ptr [rax + 0x38], rdx
0xc7a1: mov     qword ptr [rax + 0x18], rbx
0xc7a5: lea     rsi, [rdi + r13]
0xc7a9: add     rdi, rbx
0xc7ac: mov     qword ptr [rax + 0x20], rbx
0xc7b0: call    0x48b0
0xc7b5: mov     rbx, qword ptr [rip + 0x1ec4c]
0xc7bc: mov     rax, qword ptr [rbx + 0x38]
0xc7c0: mov     rdi, qword ptr [rbx + 0x58]
0xc7c4: lea     rsi, [rax + 1]
0xc7c8: mov     qword ptr [rbx + 0x50], rsi
0xc7cc: call    0x1d690
0xc7d1: mov     qword ptr [rbx + 0x58], rax
0xc7d5: pop     rbx
0xc7d6: pop     rbp
0xc7d7: pop     r12
0xc7d9: pop     r13
0xc7db: pop     r14
0xc7dd: ret
0xc7de: nop
0xc7e0: endbr64
0xc7e4: test    rdi, rdi
0xc7e7: je      0xc818
0xc7e9: push    rbp
0xc7ea: mov     rbp, rdi
0xc7ed: mov     rdi, qword ptr [rdi + 0x58]
0xc7f1: test    rdi, rdi
0xc7f4: je      0xc7fb
0xc7f6: call    0x18d10
0xc7fb: mov     rdi, qword ptr [rbp + 0x60]
0xc7ff: test    rdi, rdi
0xc802: je      0xc809
0xc804: call    0x18d10
0xc809: mov     rdi, rbp
0xc80c: pop     rbp
0xc80d: jmp     0x18d10
0x18d10:
0xc812: nop     word ptr [rax + rax]
0xc818: ret
0xc819: nop     dword ptr [rax]
0xc820: endbr64
0xc824: mov     rax, qword ptr [rip + 0x1ebdd]

```

```

0xc82b: push    rbp
0xc82c: mov     rdi, qword ptr [rax + 0x68]
0xc830: xor     eax, eax
0xc832: call    0x1aac0
0xc837: mov     rax, qword ptr [rip + 0x1ebca]
0xc83e: mov     rdi, qword ptr [rax]
0xc841: call    0xc5c0
0xc846: mov     rbp, qword ptr [rip + 0x1ebbb]
0xc84d: test    rbp, rbp
0xc850: je      0xc876
0xc852: mov     rdi, qword ptr [rbp + 0x58]
0xc856: test    rdi, rdi
0xc859: je      0xc860
0xc85b: call    0x18d10
0xc860: mov     rdi, qword ptr [rbp + 0x60]
0xc864: test    rdi, rdi
0xc867: je      0xc86e
0xc869: call    0x18d10
0xc86e: mov     rdi, rbp
0xc871: call    0x18d10
0xc876: mov     qword ptr [rip + 0x1eb87], 0
0xc881: pop     rbp
0xc882: ret
0xc883: nop     word ptr cs:[rax + rax]
0xc88e: nop
0xc890: endbr64
0xc894: mov     rax, qword ptr [rip + 0x1eb6d]
0xc89b: movsxd  rcx, edi
0xc89e: mov     edx, esi
0xc8a0: mov     r8, rcx
0xc8a3: mov     qword ptr [rax + 0x28], rcx
0xc8a7: movsxd  rcx, esi
0xc8aa: mov     rdi, qword ptr [rax]
0xc8ad: xor     esi, esi
0xc8af: mov     qword ptr [rax + 0x30], rcx
0xc8b3: mov     ecx, 1
0xc8b8: jmp     0xc5e0
0xc5e0:
0xc8bd: nop     dword ptr [rax]
0xc8c0: endbr64
0xc8c4: push    r12
0xc8c6: mov     edi, 0x70
0xc8cb: push    rbx
0xc8cc: sub     rsp, 8
0xc8d0: call    0x1d670
0xc8d5: mov     rbx, qword ptr [rip + 0x1eb2c]
0xc8dc: mov     r12, rax
0xc8df: movdqu  xmm0, xmmword ptr [rbx]

```

```

0xc8e3: movups  xmmword ptr [rax], xmm0
0xc8e6: movdqu  xmm1, xmmword ptr [rbx + 0x10]
0xc8eb: movups  xmmword ptr [rax + 0x10], xmm1
0xc8ef: movdqu  xmm2, xmmword ptr [rbx + 0x20]
0xc8f4: movups  xmmword ptr [rax + 0x20], xmm2
0xc8f8: movdqu  xmm3, xmmword ptr [rbx + 0x30]
0xc8fd: movups  xmmword ptr [rax + 0x30], xmm3
0xc901: movdqu  xmm4, xmmword ptr [rbx + 0x40]
0xc906: movups  xmmword ptr [rax + 0x40], xmm4
0xc90a: movdqu  xmm5, xmmword ptr [rbx + 0x50]
0xc90f: movups  xmmword ptr [rax + 0x50], xmm5
0xc913: movdqu  xmm6, xmmword ptr [rbx + 0x60]
0xc918: cmp     qword ptr [rax + 0x58], 0
0xc91d: movups  xmmword ptr [rax + 0x60], xmm6
0xc921: je      0xc94a
0xc923: mov     rdi, qword ptr [rax + 0x50]
0xc927: call    0x1d670
0xc92c: mov     rdx, qword ptr [r12 + 0x50]
0xc931: mov     qword ptr [rbx + 0x58], rax
0xc935: mov     rax, qword ptr [rip + 0x1eacc]
0xc93c: mov     rsi, qword ptr [r12 + 0x58]
0xc941: mov     rdi, qword ptr [rax + 0x58]
0xc945: call    0x4bb0
0xc94a: mov     rdi, qword ptr [r12 + 0x60]
0xc94f: test    rdi, rdi
0xc952: je      0xc964
0xc954: mov     rbx, qword ptr [rip + 0x1eaad]
0xc95b: call    0x1d7a0
0xc960: mov     qword ptr [rbx + 0x60], rax
0xc964: add     rsp, 8
0xc968: mov     rax, r12
0xc96b: pop     rbx
0xc96c: pop     r12
0xc96e: ret
0xc96f: nop
0xc970: endbr64
0xc974: test    rdi, rdi
0xc977: je      0xc9f0
0xc979: push    r13
0xc97b: push    r12
0xc97d: push    rbp
0xc97e: push    rbx
0xc97f: mov     rbx, rdi
0xc982: sub     rsp, 8
0xc986: mov     rbp, qword ptr [rip + 0x1ea7b]
0xc98d: mov     rdi, qword ptr [rbp + 0x58]
0xc991: mov     r13, qword ptr [rbp + 0x28]
0xc995: mov     r12, qword ptr [rbp + 0x30]

```

```

0xc999: test    rdi, rdi
0xc99c: je      0xc9a3
0xc99e: call   0x18d10
0xc9a3: mov     rdi, qword ptr [rbp + 0x60]
0xc9a7: test    rdi, rdi
0xc9aa: je      0xc9b1
0xc9ac: call   0x18d10
0xc9b1: mov     rdi, rbp
0xc9b4: call   0x18d10
0xc9b9: movsxd  rax, r13d
0xc9bc: mov     edx, r12d
0xc9bf: movsxd  r12, r12d
0xc9c2: mov     qword ptr [rbx + 0x28], rax
0xc9c6: mov     rdi, qword ptr [rbx]
0xc9c9: mov     r8d, r13d
0xc9cc: mov     ecx, 1
0xc9d1: mov     qword ptr [rbx + 0x30], r12
0xc9d5: xor     esi, esi
0xc9d7: mov     qword ptr [rip + 0x1ea2a], rbx
0xc9de: add     rsp, 8
0xc9e2: pop     rbx
0xc9e3: pop     rbp
0xc9e4: pop     r12
0xc9e6: pop     r13
0xc9e8: jmp     0xc5e0
0xc5e0:
0xc9ed: nop     dword ptr [rax]
0xc9f0: ret
0xc9f1: nop     word ptr cs:[rax + rax]
0xc9fc: nop     dword ptr [rax]
0xca00: endbr64
0xca04: mov     rdx, qword ptr [rip + 0x1e9fd]
0xca0b: mov     rax, qword ptr [rdx + 0x18]
0xca0f: sub     rax, qword ptr [rdx + 0x40]
0xca13: ret
0xca14: nop     word ptr cs:[rax + rax]
0xca1f: nop
0xca20: endbr64
0xca24: mov     rax, qword ptr [rip + 0x1e9dd]
0xca2b: mov     r8d, dword ptr [rax + 8]
0xca2f: mov     dword ptr [rax + 8], edi
0xca32: mov     eax, r8d
0xca35: ret
0xca36: nop     word ptr cs:[rax + rax]
0xca40: endbr64
0xca44: mov     rax, qword ptr [rip + 0x1e9bd]
0xca4b: cmp     qword ptr [rax + 0x48], 0
0xca50: sete    al

```

```

0xca53: movzx    eax, al
0xca56: ret
0xca57: nop      word ptr [rax + rax]
0xca60: endbr64
0xca64: mov      rax, qword ptr [rip + 0x1e99d]
0xca6b: mov      rdx, qword ptr [rax + 0x18]
0xca6f: mov      qword ptr [rax + 0x20], rdx
0xca73: ret
0xca74: nop      word ptr cs:[rax + rax]
0xca7f: nop
0xca80: endbr64
0xca84: sub      rsp, 8
0xca88: xor      eax, eax
0xca8a: mov      edi, 3
0xca8f: call     0xc6e0
0xca94: mov      rax, qword ptr [rip + 0x1e96d]
0xca9b: mov      dword ptr [rax + 0x10], 0x18
0xcaa2: add      rsp, 8
0xcaa6: ret
0xcaa7: nop      word ptr [rax + rax]
0xcab0: endbr64
0xcab4: sub      rsp, 8
0xcab8: xor      eax, eax
0xcaba: mov      edi, 1
0xcabf: call     0xc6e0
0xcac4: mov      rax, qword ptr [rip + 0x1e93d]
0xcacb: mov      dword ptr [rax + 0x10], 0x19
0xcad2: add      rsp, 8
0xcad6: ret
0xcad7: nop      word ptr [rax + rax]
0xcae0: endbr64
0xcae4: mov      rax, qword ptr [rip + 0x1e91d]
0xcaeb: mov      rdx, qword ptr [rax + 0x18]
0xcaef: mov      rcx, qword ptr [rax + 0x20]
0xcaf3: movsxd   rdx, edx
0xcaf6: mov      qword ptr [rax + 0x18], rcx
0xcafa: mov      qword ptr [rax + 0x20], rdx
0xcafe: ret
0xcaff: nop
0xcb00: endbr64
0xcb04: mov      rax, qword ptr [rip + 0x1e8fd]
0xcb0b: mov      rdx, qword ptr [rax + 0x18]
0xcb0f: cmp      rdx, qword ptr [rax + 0x40]
0xcb13: jbe      0xcb24
0xcb15: sub      rdx, 1
0xcb19: mov      dword ptr [rax + 0x10], 1
0xcb20: mov      qword ptr [rax + 0x18], rdx
0xcb24: ret

```



```

0xcb25: nop      word ptr cs:[rax + rax]
0xcb30: endbr64
0xcb34: mov      rax, qword ptr [rip + 0x1e8cd]
0xcb3b: mov      rdx, qword ptr [rax + 0x18]
0xcb3f: cmp      rdx, qword ptr [rax + 0x38]
0xcb43: jae      0xcb54
0xcb45: add      rdx, 1
0xcb49: mov      dword ptr [rax + 0x10], 2
0xcb50: mov      qword ptr [rax + 0x18], rdx
0xcb54: ret
0xcb55: nop      word ptr cs:[rax + rax]
0xcb60: endbr64
0xcb64: mov      rdx, qword ptr [rip + 0x1e89d]
0xcb6b: mov      rax, qword ptr [rdx + 0x18]
0xcb6f: mov      rdi, qword ptr [rdx + 0x40]
0xcb73: cmp      rax, rdi
0xcb76: jbe      0xcc20
0xcb7c: mov      r8, qword ptr [rdx + 0x58]
0xcb80: jmp      0xcba9
0xcba9
0xcb82: nop      word ptr [rax + rax]
0xcb88: lea      ecx, [rsi - 0x30]
0xcb8b: cmp      cl, 9
0xcb8e: jbe      0xcbbf
0xcb90: sub      esi, 0x24
0xcb93: cmp      sil, 1
0xcb97: jbe      0xcbbf
0xcb99: mov      qword ptr [rdx + 0x18], rax
0xcb9d: mov      dword ptr [rdx + 0x10], 1
0xcba4: cmp      rdi, rax
0xcba7: jae      0xcbef
0xcba9: sub      rax, 1
0xcbad: movzx   esi, byte ptr [r8 + rax]
0xcbb2: mov      ecx, esi
0xcbb4: and      ecx, 0xffffffff
0xcbb7: sub      ecx, 0x41
0xcbb8: cmp      cl, 0x19
0xcbbd: ja       0xcb88
0xcbbf: mov      rax, qword ptr [rdx + 0x18]
0xcbc3: cmp      rdi, rax
0xcbc6: jae      0xcbef
0xcbc8: sub      rax, 1
0xcbcc: movzx   esi, byte ptr [r8 + rax]
0xcbd1: mov      ecx, esi
0xcbd3: and      ecx, 0xffffffff
0xcbd6: sub      ecx, 0x41
0xcbd9: cmp      cl, 0x19
0xcfdc: jbe      0xcc00

```

```

0xcbde: lea     ecx, [rsi - 0x30]
0xcbe1: cmp     cl, 9
0xcbe4: jbe     0xcc00
0xcbe6: sub     esi, 0x24
0xcbe9: cmp     sil, 1
0xcbed: jbe     0xcc00
0xcbef: mov     dword ptr [rdx + 0x10], 3
0xcbf6: ret
0xcbf7: nop     word ptr [rax + rax]
0xcc00: mov     qword ptr [rdx + 0x18], rax
0xcc04: mov     dword ptr [rdx + 0x10], 1
0xcc0b: cmp     rdi, rax
0xcc0e: jb      0xcbc8
0xcc10: mov     dword ptr [rdx + 0x10], 3
0xcc17: ret
0xcc18: nop     dword ptr [rax + rax]
0xcc20: ret
0xcc21: nop     word ptr cs:[rax + rax]
0xcc2c: nop     dword ptr [rax]
0xcc30: endbr64
0xcc34: mov     rdx, qword ptr [rip + 0x1e7cd]
0xcc3b: mov     rax, qword ptr [rdx + 0x18]
0xcc3f: mov     rdi, qword ptr [rdx + 0x38]
0xcc43: cmp     rax, rdi
0xcc46: jae     0xcce8
0xcc4c: mov     r8, qword ptr [rdx + 0x58]
0xcc50: jmp     0xcc7d
0xcc52: nop     word ptr [rax + rax]
0xcc58: lea     ecx, [rsi - 0x30]
0xcc5b: cmp     cl, 9
0xcc5e: jbe     0xcc8f
0xcc60: sub     esi, 0x24
0xcc63: cmp     sil, 1
0xcc67: jbe     0xcc8f
0xcc69: add     rax, 1
0xcc6d: mov     dword ptr [rdx + 0x10], 2
0xcc74: mov     qword ptr [rdx + 0x18], rax
0xcc78: cmp     rdi, rax
0xcc7b: je      0xccbb
0xcc7d: movzx   esi, byte ptr [r8 + rax]
0xcc82: mov     ecx, esi
0xcc84: and     ecx, 0xffffffff
0xcc87: sub     ecx, 0x41
0xcc8a: cmp     cl, 0x19
0xcc8d: ja      0xcc58
0xcc8f: mov     rax, qword ptr [rdx + 0x18]
0xcc93: cmp     rdi, rax

```

```

0xcc96: jbe      0xccbb
0xcc98: movzx   esi, byte ptr [r8 + rax]
0xcc9d: mov     ecx, esi
0xcc9f: and     ecx, 0xffffffff
0xcca2: sub     ecx, 0x41
0xcca5: cmp     cl, 0x19
0xcca8: jbe     0xcc8
0xccaa: lea     ecx, [rsi - 0x30]
0xccad: cmp     cl, 9
0xccb0: jbe     0xcc8
0xccb2: sub     esi, 0x24
0xccb5: cmp     sil, 1
0xccb9: jbe     0xcc8
0xccbb: mov     dword ptr [rdx + 0x10], 4
0xcc2: ret
0xcc3: nop     dword ptr [rax + rax]
0xcc8: add     rax, 1
0xcccc: mov     dword ptr [rdx + 0x10], 2
0xcd3: mov     qword ptr [rdx + 0x18], rax
0xcd7: cmp     rdi, rax
0xcda: jne     0xcc98
0xcdc: mov     dword ptr [rdx + 0x10], 4
0xcce3: ret
0xcce4: nop     dword ptr [rax]
0xcce8: ret
0xcce9: nop     dword ptr [rax]
0xccf0: endbr64
0xccf4: mov     rax, qword ptr [rip + 0x1e70d]
0xccfb: mov     rdx, qword ptr [rax + 0x40]
0xccff: mov     dword ptr [rax + 0x10], 5
0xcd06: mov     qword ptr [rax + 0x18], rdx
0xcd0a: ret
0xcd0b: nop     dword ptr [rax + rax]
0xcd10: endbr64
0xcd14: mov     rax, qword ptr [rip + 0x1e6ed]
0xcd1b: mov     rdx, qword ptr [rax + 0x38]
0xcd1f: mov     dword ptr [rax + 0x10], 6
0xcd26: mov     qword ptr [rax + 0x18], rdx
0xcd2a: ret
0xcd2b: nop     dword ptr [rax + rax]
0xcd30: endbr64
0xcd34: push    rbp
0xcd35: push    rbx
0xcd36: mov     ebx, edi
0xcd38: sub     rsp, 8
0xcd3c: call    0x4f00
0xcd41: movsxd  rdx, ebx
0xcd44: mov     rax, qword ptr [rax]

```

```

0xcd47: test    byte ptr [rax + rdx*2 + 1], 0x40
0xcd4c: je        0xcdab
0xcd4e: mov      rbp, qword ptr [rip + 0x1e6b3]
0xcd55: mov      rsi, qword ptr [rbp + 0x38]
0xcd59: lea      rdx, [rsi + 1]
0xcd5d: cmp      rdx, qword ptr [rbp + 0x50]
0xcd61: jae      0xcdb8
0xcd63: mov      rax, qword ptr [rbp + 0x18]
0xcd67: mov      rcx, qword ptr [rbp + 0x58]
0xcd6b: sub      rdx, rax
0xcd6e: lea      rsi, [rcx + rax]
0xcd72: lea      rdi, [rcx + rax + 1]
0xcd77: call     0x4d40
0xcd7c: mov      rax, qword ptr [rip + 0x1e685]
0xcd83: mov      rdx, qword ptr [rax + 0x58]
0xcd87: mov      rax, qword ptr [rax + 0x18]
0xcd8b: mov      byte ptr [rdx + rax], bl
0xcd8e: mov      rax, qword ptr [rip + 0x1e673]
0xcd95: add      qword ptr [rax + 0x18], 1
0xcd9a: add      qword ptr [rax + 0x38], 1
0xcd9f: add      qword ptr [rax + 0x48], 1
0xcda4: mov      dword ptr [rax + 0x10], 7
0xcdab: add      rsp, 8
0xcdaf: pop      rbx
0xcdb0: pop      rbp
0xcdb1: ret
0xcdb2: nop      word ptr [rax + rax]
0xcdb8: add      rsi, 0x21
0xcdbc: mov      rdi, qword ptr [rbp + 0x58]
0xcdc0: mov      qword ptr [rbp + 0x50], rsi
0xcdc4: call     0x1d690
0xcdc9: mov      qword ptr [rbp + 0x58], rax
0xcdcd: mov      rbp, qword ptr [rip + 0x1e634]
0xcdd4: mov      rax, qword ptr [rbp + 0x38]
0xcdd8: lea      rdx, [rax + 1]
0xcddc: jmp      0xcd63
0xcd63:
0xcdde: nop
0xcde0: endbr64
0xcde4: mov      rax, qword ptr [rip + 0x1e61d]
0xcdeb: mov      rcx, qword ptr [rax + 0x18]
0xcdef: mov      rdx, qword ptr [rax + 0x38]
0xcdf3: cmp      rcx, rdx
0xcdf6: jbe      0xce00
0xcdf8: ret
0xcdf9: nop      dword ptr [rax]
0xce00: push     rbx
0xce01: mov      rax, qword ptr [rax + 0x58]

```

```

0xce05: add     rdx, 1
0xce09: sub     rdx, rcx
0xce0c: lea     rsi, [rax + rcx + 1]
0xce11: lea     rdi, [rax + rcx]
0xce15: call    0x4bb0
0xce1a: mov     rbx, qword ptr [rip + 0x1e5e7]
0xce21: mov     rsi, qword ptr [rbx + 0x38]
0xce25: sub     qword ptr [rbx + 0x48], 1
0xce2a: lea     rax, [rsi - 1]
0xce2e: mov     qword ptr [rbx + 0x38], rax
0xce32: test    al, 0xf
0xce34: je      0xce40
0xce36: mov     dword ptr [rbx + 0x10], 8
0xce3d: pop     rbx
0xce3e: ret
0xce3f: nop
0xce40: mov     qword ptr [rbx + 0x50], rsi
0xce44: mov     rdi, qword ptr [rbx + 0x58]
0xce48: call    0x1d690
0xce4d: mov     qword ptr [rbx + 0x58], rax
0xce51: mov     rbx, qword ptr [rip + 0x1e5b0]
0xce58: mov     dword ptr [rbx + 0x10], 8
0xce5f: pop     rbx
0xce60: ret
0xce61: nop     word ptr cs:[rax + rax]
0xce6c: nop     dword ptr [rax]
0xce70: endbr64
0xce74: mov     rax, qword ptr [rip + 0x1e58d]
0xce7b: mov     rcx, qword ptr [rax + 0x18]
0xce7f: cmp     rcx, qword ptr [rax + 0x40]
0xce83: ja      0xce90
0xce85: ret
0xce86: nop     word ptr cs:[rax + rax]
0xce90: push    rbx
0xce91: mov     rdx, qword ptr [rax + 0x38]
0xce95: mov     rsi, qword ptr [rax + 0x58]
0xce99: add     rdx, 1
0xce9d: lea     rdi, [rsi + rcx - 1]
0xcea2: sub     rdx, rcx
0xcea5: add     rsi, rcx
0xcea8: call    0x4bb0
0xcead: mov     rbx, qword ptr [rip + 0x1e554]
0xceb4: mov     rsi, qword ptr [rbx + 0x38]
0xceb8: sub     qword ptr [rbx + 0x18], 1
0xcebd: sub     qword ptr [rbx + 0x48], 1
0xcec2: lea     rax, [rsi - 1]
0xcec6: mov     qword ptr [rbx + 0x38], rax
0xceca: test    al, 0xf

```

```

0xcecc: je      0xcee0
0xcece: mov     dword ptr [rbx + 0x10], 9
0xcd5: pop     rbx
0xcd6: ret
0xcd7: nop     word ptr [rax + rax]
0xcee0: mov     qword ptr [rbx + 0x50], rsi
0xcee4: mov     rdi, qword ptr [rbx + 0x58]
0xcee8: call    0x1d690
0xceed: mov     qword ptr [rbx + 0x58], rax
0xcef1: mov     rbx, qword ptr [rip + 0x1e510]
0xcef8: mov     dword ptr [rbx + 0x10], 9
0xceff: pop     rbx
0xcf00: ret
0xcf01: nop     word ptr cs:[rax + rax]
0xcf0c: nop     dword ptr [rax]
0xcf10: endbr64
0xcf14: push    rbp
0xcf15: push    rbx
0xcf16: sub     rsp, 8
0xcf1a: mov     r9, qword ptr [rip + 0x1e4e7]
0xcf21: mov     rax, qword ptr [r9 + 0x18]
0xcf25: mov     rbx, qword ptr [r9 + 0x20]
0xcf29: mov     qword ptr [r9 + 0x20], rax
0xcf2d: xor     eax, eax
0xcf2f: call    0xcc30
0xcf34: xor     eax, eax
0xcf36: mov     edi, 3
0xcf3b: mov     rbp, qword ptr [r9 + 0x18]
0xcf3f: call    0xc6e0
0xcf44: mov     rax, qword ptr [rip + 0x1e4bd]
0xcf4b: mov     rdx, qword ptr [rax + 0x18]
0xcf4f: cmp     rdx, rbx
0xcf52: jae     0xcf65
0xcf54: lea     rcx, [rdx + rbx]
0xcf58: sub     rcx, rbp
0xcf5b: cmp     rbx, rbp
0xcf5e: cmovae  rdx, rcx
0xcf62: mov     rbx, rdx
0xcf65: mov     qword ptr [rax + 0x20], rbx
0xcf69: mov     dword ptr [rax + 0x10], 0xa
0xcf70: add     rsp, 8
0xcf74: pop     rbx
0xcf75: pop     rbp
0xcf76: ret
0xcf77: nop     word ptr [rax + rax]
0xcf80: endbr64
0xcf84: push    rbp
0xcf85: push    rbx

```

```

0xcf86: sub     rsp, 8
0xcf8a: mov     rax, qword ptr [rip + 0x1e477]
0xcf91: mov     rbp, qword ptr [rax + 0x18]
0xcf95: mov     rbx, qword ptr [rax + 0x20]
0xcf99: mov     qword ptr [rax + 0x20], rbp
0xcf9d: xor     eax, eax
0xcf9f: call    0xcb60
0xcfa4: xor     eax, eax
0xcfa6: mov     edi, 3
0xcfab: call    0xc6e0
0xcfb0: mov     rax, qword ptr [rip + 0x1e451]
0xcfb7: mov     rdx, qword ptr [rax + 0x18]
0xcfb9: cmp     rdx, rbx
0xcfbe: jae     0xcfd1
0xcfc0: lea     rcx, [rdx + rbx]
0xcfc4: sub     rcx, rbp
0xcfc7: cmp     rbx, rbp
0xcfca: cmovae  rdx, rcx
0xcfce: mov     rbx, rdx
0xcfd1: mov     qword ptr [rax + 0x20], rbx
0xcfd5: mov     dword ptr [rax + 0x10], 0xb
0xcfdc: add     rsp, 8
0xcfe0: pop     rbx
0xcfe1: pop     rbp
0xcfe2: ret
0xcfe3: nop     word ptr cs:[rax + rax]
0xcfee: nop
0xcff0: endbr64
0xcff4: push    rbx
0xcff5: mov     rbx, qword ptr [rip + 0x1e40c]
0xcffc: mov     esi, 1
0xd001: mov     qword ptr [rbx + 0x18], 0
0xd009: mov     rdi, qword ptr [rbx + 0x58]
0xd00d: mov     qword ptr [rbx + 0x20], 0
0xd015: mov     qword ptr [rbx + 0x38], 0
0xd01d: mov     qword ptr [rbx + 0x40], 0
0xd025: mov     qword ptr [rbx + 0x48], 0
0xd02d: mov     qword ptr [rbx + 0x50], 1
0xd035: call    0x1d690
0xd03a: mov     qword ptr [rbx + 0x58], rax
0xd03e: mov     rax, qword ptr [rip + 0x1e3c3]
0xd045: mov     rax, qword ptr [rax + 0x58]
0xd049: mov     byte ptr [rax], 0
0xd04c: mov     rax, qword ptr [rip + 0x1e3b5]
0xd053: mov     dword ptr [rax + 0x10], 0xc
0xd05a: pop     rbx
0xd05b: ret
0xd05c: nop     dword ptr [rax]

```

```

0xd060: endbr64
0xd064: push    rbx
0xd065: mov     edi, 0x70
0xd06a: sub     rsp, 0x10
0xd06e: mov     rax, qword ptr fs:[0x28]
0xd077: mov     qword ptr [rsp + 8], rax
0xd07c: xor     eax, eax
0xd07e: call    0x1d670
0xd083: mov     qword ptr [rax + 8], 1
0xd08b: mov     qword ptr [rax + 0x58], 0
0xd093: mov     qword ptr [rax + 0x60], 0
0xd09b: mov     qword ptr [rax + 0x28], 0
0xd0a3: mov     qword ptr [rax + 0x30], 0
0xd0ab: mov     qword ptr [rip + 0x1e356], rax
0xd0b2: xor     eax, eax
0xd0b4: call    0xcff0
0xd0b9: mov     rbx, qword ptr [rip + 0x1e348]
0xd0c0: xor     eax, eax
0xd0c2: call    0xc590
0xd0c7: lea     rdi, [rip + 0x135e4]
0xd0ce: mov     qword ptr [rbx], rax
0xd0d1: call    0x17c50
0xd0d6: mov     ecx, 1
0xd0db: mov     rsi, rsp
0xd0de: mov     edx, 1
0xd0e3: lea     rdi, [rip + 0x13e42]
0xd0ea: call    0x17320
0xd0ef: mov     rdi, qword ptr [rsp]
0xd0f3: lea     rax, [rip + 0x13e24]
0xd0fa: mov     rbx, qword ptr [rip + 0x1e307]
0xd101: test    rdi, rdi
0xd104: cmove   rdi, rax
0xd108: xor     eax, eax
0xd10a: call    0x19940
0xd10f: lea     rdi, [rip + 0x1cb32]
0xd116: mov     qword ptr [rbx + 0x68], rax
0xd11a: mov     eax, dword ptr [rip + 0x1cb40]
0xd120: test    eax, eax
0xd122: lea     rax, [rip + 0x1cb07]
0xd129: cmove   rdi, rax
0xd12d: call    0x17c50
0xd132: lea     rdi, [rip + 0x1d0c7]
0xd139: mov     edx, 6
0xd13e: lea     rsi, [rip + 0x1d0db]
0xd145: call    0x17f70
0xd14a: xor     eax, eax
0xd14c: call    0x1a310
0xd151: mov     rax, qword ptr [rip + 0x1e2b0]

```



```

0xd158: mov     rdi, qword ptr [rax + 0x68]
0xd15c: xor     eax, eax
0xd15e: call    0x1a900
0xd163: nop     dword ptr [rax + rax]
0xd168: xor     eax, eax
0xd16a: call    0x1ab40
0xd16f: test    rax, rax
0xd172: jne     0xd168
0xd174: mov     edi, 0x200
0xd179: call    0x1a620
0xd17e: mov     rax, qword ptr [rsp + 8]
0xd183: sub     rax, qword ptr fs:[0x28]
0xd18c: jne     0xd194
0xd18e: add     rsp, 0x10
0xd192: pop     rbx
0xd193: ret
0xd194: call    0x49d0
0xd199: nop     dword ptr [rax]
0xd1a0: endbr64
0xd1a4: sub     rsp, 8
0xd1a8: mov     rax, qword ptr [rip + 0x1e259]
0xd1af: or      edi, 2
0xd1b2: mov     rdx, qword ptr [rax + 0x40]
0xd1b6: mov     dword ptr [rax + 0x10], 6
0xd1bd: mov     qword ptr [rax + 0x20], rdx
0xd1c1: mov     rdx, qword ptr [rax + 0x38]
0xd1c5: mov     qword ptr [rax + 0x18], rdx
0xd1c9: xor     eax, eax
0xd1cb: call    0xc6e0
0xd1d0: mov     rax, qword ptr [rip + 0x1e231]
0xd1d7: mov     rdx, qword ptr [rax + 0x18]
0xd1db: mov     dword ptr [rax + 0x10], 0xd
0xd1e2: mov     qword ptr [rax + 0x20], rdx
0xd1e6: add     rsp, 8
0xd1ea: ret
0xd1eb: nop     dword ptr [rax + rax]
0xd1f0: endbr64
0xd1f4: mov     rax, qword ptr [rip + 0x1e20d]
0xd1fb: push    rbx
0xd1fc: mov     edi, 3
0xd201: mov     rdx, qword ptr [rax + 0x18]
0xd205: mov     rsi, qword ptr [rax + 0x20]
0xd209: mov     dword ptr [rax + 0x10], 5
0xd210: mov     rcx, qword ptr [rax + 0x40]
0xd214: mov     qword ptr [rax + 0x20], rdx
0xd218: mov     rbx, rsi
0xd21b: mov     qword ptr [rax + 0x18], rcx
0xd21f: sub     rbx, rdx

```

```

0xd222: cmp     rsi, rdx
0xd225: cmovbe  rbx, rcx
0xd229: xor     eax, eax
0xd22b: call    0xc6e0
0xd230: mov     rdx, qword ptr [rip + 0x1e1d1]
0xd237: mov     rax, qword ptr [rdx + 0x38]
0xd23b: mov     dword ptr [rdx + 0x10], 0xe
0xd242: cmp     rax, rbx
0xd245: cmova   rax, rbx
0xd249: mov     qword ptr [rdx + 0x20], rax
0xd24d: pop     rbx
0xd24e: ret
0xd24f: nop
0xd250: endbr64
0xd254: mov     rax, qword ptr [rip + 0x1e1ad]
0xd25b: push    rbx
0xd25c: mov     edi, 3
0xd261: mov     rdx, qword ptr [rax + 0x18]
0xd265: mov     rbx, qword ptr [rax + 0x20]
0xd269: mov     dword ptr [rax + 0x10], 6
0xd270: mov     qword ptr [rax + 0x20], rdx
0xd274: mov     rdx, qword ptr [rax + 0x38]
0xd278: mov     qword ptr [rax + 0x18], rdx
0xd27c: xor     eax, eax
0xd27e: call    0xc6e0
0xd283: mov     rdx, qword ptr [rip + 0x1e17e]
0xd28a: mov     rax, qword ptr [rdx + 0x38]
0xd28e: mov     dword ptr [rdx + 0x10], 0xf
0xd295: cmp     rax, rbx
0xd298: cmova   rax, rbx
0xd29c: mov     qword ptr [rdx + 0x20], rax
0xd2a0: pop     rbx
0xd2a1: ret
0xd2a2: nop     word ptr cs:[rax + rax]
0xd2ad: nop     dword ptr [rax]
0xd2b0: endbr64
0xd2b4: mov     rax, qword ptr [rip + 0x1e14d]
0xd2bb: mov     rdx, qword ptr [rax + 0x58]
0xd2bf: mov     rax, qword ptr [rax + 0x18]
0xd2c3: cmp     byte ptr [rdx + rax], 0x20
0xd2c7: je      0xd2d0
0xd2c9: ret
0xd2ca: nop     word ptr [rax + rax]
0xd2d0: sub     rsp, 8
0xd2d4: nop     dword ptr [rax]
0xd2d8: xor     eax, eax
0xd2da: call    0xcde0
0xd2df: mov     rax, qword ptr [rip + 0x1e122]

```

```

0xd2e6: mov     rdx, qword ptr [rax + 0x58]
0xd2ea: mov     rcx, qword ptr [rax + 0x18]
0xd2ee: cmp     byte ptr [rdx + rcx], 0x20
0xd2f2: je      0xd2d8
0xd2f4: cmp     qword ptr [rax + 0x48], 0
0xd2f9: jne     0xd31d
0xd2fb: jmp     0xd324
0xd324:
0xd2fd: nop     dword ptr [rax]
0xd300: xor     eax, eax
0xd302: call    0xce70
0xd307: mov     rax, qword ptr [rip + 0x1e0fa]
0xd30e: cmp     qword ptr [rax + 0x48], 0
0xd313: je      0xd324
0xd315: mov     rdx, qword ptr [rax + 0x58]
0xd319: mov     rcx, qword ptr [rax + 0x18]
0xd31d: cmp     byte ptr [rdx + rcx - 1], 0x20
0xd322: je      0xd300
0xd324: mov     dword ptr [rax + 0x10], 0x11
0xd32b: add     rsp, 8
0xd32f: ret
0xd330: endbr64
0xd334: mov     rax, qword ptr [rip + 0x1e0cd]
0xd33b: mov     rdx, qword ptr [rax + 0x58]
0xd33f: mov     rax, qword ptr [rax + 0x18]
0xd343: cmp     byte ptr [rdx + rax], 0x20
0xd347: je      0xd350
0xd349: ret
0xd34a: nop     word ptr [rax + rax]
0xd350: sub     rsp, 8
0xd354: xor     eax, eax
0xd356: call    0xd2b0
0xd35b: mov     edi, 0x20
0xd360: call    0xcd30
0xd365: mov     rax, qword ptr [rip + 0x1e09c]
0xd36c: mov     dword ptr [rax + 0x10], 0x10
0xd373: add     rsp, 8
0xd377: ret
0xd378: nop     dword ptr [rax + rax]
0xd380: endbr64
0xd384: push    rbp
0xd385: push    rbx
0xd386: sub     rsp, 8
0xd38a: mov     rbp, qword ptr [rip + 0x1e077]
0xd391: mov     rbx, qword ptr [rbp + 0x18]
0xd395: cmp     rbx, qword ptr [rbp + 0x38]
0xd399: jb      0xd3a8
0xd39b: add     rsp, 8

```

```

0xd39f: pop      rbx
0xd3a0: pop      rbp
0xd3a1: ret
0xd3a2: nop      word ptr [rax + rax]
0xd3a8: xor      eax, eax
0xd3aa: call     0xcc30
0xd3af: cmp      rbx, qword ptr [rbp + 0x18]
0xd3b3: jae      0xd3e4
0xd3b5: call     0x4ef0
0xd3ba: nop      word ptr [rax + rax]
0xd3c0: mov      rdx, qword ptr [rbp + 0x58]
0xd3c4: mov      rcx, qword ptr [rax]
0xd3c7: add      rdx, rbx
0xd3ca: add      rbx, 1
0xd3ce: movsx    rsi, byte ptr [rdx]
0xd3d2: mov      ecx, dword ptr [rcx + rsi*4]
0xd3d5: mov      byte ptr [rdx], cl
0xd3d7: mov      rbp, qword ptr [rip + 0x1e02a]
0xd3de: cmp      qword ptr [rbp + 0x18], rbx
0xd3e2: ja       0xd3c0
0xd3e4: mov      dword ptr [rbp + 0x10], 0x12
0xd3eb: add      rsp, 8
0xd3ef: pop      rbx
0xd3f0: pop      rbp
0xd3f1: ret
0xd3f2: nop      word ptr cs:[rax + rax]
0xd3fd: nop      dword ptr [rax]
0xd400: endbr64
0xd404: push     rbp
0xd405: push     rbx
0xd406: sub      rsp, 8
0xd40a: mov      rbp, qword ptr [rip + 0x1dff7]
0xd411: mov      rbx, qword ptr [rbp + 0x18]
0xd415: cmp      rbx, qword ptr [rbp + 0x38]
0xd419: jb       0xd428
0xd41b: add      rsp, 8
0xd41f: pop      rbx
0xd420: pop      rbp
0xd421: ret
0xd422: nop      word ptr [rax + rax]
0xd428: xor      eax, eax
0xd42a: call     0xcc30
0xd42f: cmp      rbx, qword ptr [rbp + 0x18]
0xd433: jae      0xd464
0xd435: call     0x47c0
0xd43a: nop      word ptr [rax + rax]
0xd440: mov      rdx, qword ptr [rbp + 0x58]
0xd444: mov      rcx, qword ptr [rax]

```

```

0xd447: add    rdx, rbx
0xd44a: add    rbx, 1
0xd44e: movsx  rsi, byte ptr [rdx]
0xd452: mov    ecx, dword ptr [rcx + rsi*4]
0xd455: mov    byte ptr [rdx], cl
0xd457: mov    rbp, qword ptr [rip + 0x1dfaa]
0xd45e: cmp    qword ptr [rbp + 0x18], rbx
0xd462: ja     0xd440
0xd464: mov    dword ptr [rbp + 0x10], 0x13
0xd46b: add    rsp, 8
0xd46f: pop    rbx
0xd470: pop    rbp
0xd471: ret
0xd472: nop    word ptr cs:[rax + rax]
0xd47d: nop    dword ptr [rax]
0xd480: endbr64
0xd484: push   r14
0xd486: push   r13
0xd488: push   r12
0xd48a: push   rbp
0xd48b: mov    rbp, qword ptr [rip + 0x1df76]
0xd492: push   rbx
0xd493: mov    rbx, qword ptr [rbp + 0x18]
0xd497: cmp    rbx, qword ptr [rbp + 0x38]
0xd49b: jb     0xd4b0
0xd49d: pop    rbx
0xd49e: pop    rbp
0xd49f: pop    r12
0xd4a1: pop    r13
0xd4a3: pop    r14
0xd4a5: ret
0xd4a6: nop    word ptr cs:[rax + rax]
0xd4b0: xor    eax, eax
0xd4b2: call   0xcc30
0xd4b7: cmp    rbx, qword ptr [rbp + 0x18]
0xd4bb: jae    0xd528
0xd4bd: call   0x4f00
0xd4c2: mov    r14d, 1
0xd4c8: mov    r13, rax
0xd4cb: jmp    0xd4f3
0xd4f3:
0xd4cd: nop    dword ptr [rax]
0xd4d0: call   0x47c0
0xd4d5: xor    r14d, r14d
0xd4d8: mov    rax, qword ptr [rax]
0xd4db: mov    eax, dword ptr [rax + r12*4]
0xd4df: mov    byte ptr [rbp], al
0xd4e2: mov    rbp, qword ptr [rip + 0x1df1f]

```

```

0xd4e9: add    rbx, 1
0xd4ed: cmp    qword ptr [rbp + 0x18], rbx
0xd4f1: jbe    0xd528
0xd4f3: mov    rax, qword ptr [rbp + 0x58]
0xd4f7: add    rax, rbx
0xd4fa: movsx  r12, byte ptr [rax]
0xd4fe: mov    rbp, rax
0xd501: mov    rax, qword ptr [r13]
0xd505: test   byte ptr [rax + r12*2], 8
0xd50a: je     0xd4e2
0xd50c: test   r14d, r14d
0xd50f: jne    0xd4d0
0xd511: call   0x4ef0
0xd516: mov    rax, qword ptr [rax]
0xd519: mov    eax, dword ptr [rax + r12*4]
0xd51d: mov    byte ptr [rbp], al
0xd520: jmp    0xd4e2
0xd4e2:
0xd522: nop    word ptr [rax + rax]
0xd528: mov    dword ptr [rbp + 0x10], 0x14
0xd52f: pop    rbx
0xd530: pop    rbp
0xd531: pop    r12
0xd533: pop    r13
0xd535: pop    r14
0xd537: ret
0xd538: nop    dword ptr [rax + rax]
0xd540: endbr64
0xd544: push   r12
0xd546: mov    r12, rdi
0xd549: push   rbp
0xd54a: push   rbx
0xd54b: call   0x49b0
0xd550: mov    rbp, qword ptr [rip + 0x1deb1]
0xd557: mov    rbx, rax
0xd55a: mov    rsi, qword ptr [rbp + 0x40]
0xd55e: sub    rax, rsi
0xd561: add    qword ptr [rbp + 0x18], rax
0xd565: add    qword ptr [rbp + 0x20], rax
0xd569: mov    rax, qword ptr [rbp + 0x48]
0xd56d: lea    rdx, [rax + 1]
0xd571: lea    r8, [rbx + rdx]
0xd575: cmp    r8, qword ptr [rbp + 0x50]
0xd579: ja     0xd5d8
0xd57b: mov    rdi, qword ptr [rbp + 0x58]
0xd57f: add    rsi, rdi
0xd582: add    rdi, rbx
0xd585: call   0x4d40

```

```

0xd58a: mov     rax, qword ptr [rip + 0x1de77]
0xd591: mov     rdx, rbx
0xd594: mov     rsi, r12
0xd597: mov     rdi, qword ptr [rax + 0x58]
0xd59b: call    0x4bb0
0xd5a0: mov     rax, qword ptr [rip + 0x1de61]
0xd5a7: mov     rsi, rbx
0xd5aa: mov     rdi, qword ptr [rax + 0x58]
0xd5ae: call    0x184d0
0xd5b3: mov     rax, qword ptr [rip + 0x1de4e]
0xd5ba: mov     qword ptr [rax + 0x40], rbx
0xd5be: add     rbx, qword ptr [rax + 0x48]
0xd5c2: mov     qword ptr [rax + 0x38], rbx
0xd5c6: mov     dword ptr [rax + 0x10], 0x15
0xd5cd: pop     rbx
0xd5ce: pop     rbp
0xd5cf: pop     r12
0xd5d1: ret
0xd5d2: nop     word ptr [rax + rax]
0xd5d8: mov     qword ptr [rbp + 0x50], r8
0xd5dc: mov     rdi, qword ptr [rbp + 0x58]
0xd5e0: mov     rsi, r8
0xd5e3: call    0x1d690
0xd5e8: mov     qword ptr [rbp + 0x58], rax
0xd5ec: mov     rbp, qword ptr [rip + 0x1de15]
0xd5f3: mov     rax, qword ptr [rbp + 0x48]
0xd5f7: mov     rsi, qword ptr [rbp + 0x40]
0xd5fb: lea     rdx, [rax + 1]
0xd5ff: jmp     0xd57b
0xd57b:
0xd604: nop     word ptr cs:[rax + rax]
0xd60f: nop
0xd610: endbr64
0xd614: test    rdi, rdi
0xd617: je      0xd6f0
0xd61d: push    r12
0xd61f: push    rbp
0xd620: mov     rbp, rdi
0xd623: push    rbx
0xd624: call    0x49b0
0xd629: mov     r12, qword ptr [rip + 0x1ddd8]
0xd630: mov     rbx, rax
0xd633: mov     rax, qword ptr [r12 + 0x38]
0xd638: lea     rdx, [rax + 1]
0xd63c: mov     rax, qword ptr [r12 + 0x50]
0xd641: lea     rcx, [rbx + rdx]
0xd645: cmp     rcx, rax
0xd648: ja      0xd6c0

```

```

0xd64a: mov     rax, qword ptr [r12 + 0x18]
0xd64f: mov     rdi, qword ptr [r12 + 0x58]
0xd654: sub     rdx, rax
0xd657: lea     rsi, [rdi + rax]
0xd65b: add     rax, rbx
0xd65e: add     rdi, rax
0xd661: call    0x4d40
0xd666: mov     rax, qword ptr [rip + 0x1dd9b]
0xd66d: mov     rdx, rbx
0xd670: mov     rsi, rbp
0xd673: mov     rdi, qword ptr [rax + 0x18]
0xd677: add     rdi, qword ptr [rax + 0x58]
0xd67b: call    0x4bb0
0xd680: mov     rax, qword ptr [rip + 0x1dd81]
0xd687: mov     rsi, rbx
0xd68a: mov     rdi, qword ptr [rax + 0x18]
0xd68e: add     rdi, qword ptr [rax + 0x58]
0xd692: call    0x184d0
0xd697: mov     rax, qword ptr [rip + 0x1dd6a]
0xd69e: add     qword ptr [rax + 0x18], rbx
0xd6a2: add     qword ptr [rax + 0x38], rbx
0xd6a6: add     qword ptr [rax + 0x48], rbx
0xd6aa: mov     dword ptr [rax + 0x10], 0x16
0xd6b1: pop     rbx
0xd6b2: pop     rbp
0xd6b3: pop     r12
0xd6b5: ret
0xd6b6: nop     word ptr cs:[rax + rax]
0xd6c0: lea     rsi, [rax + rbx + 0x21]
0xd6c5: mov     rdi, qword ptr [r12 + 0x58]
0xd6ca: mov     qword ptr [r12 + 0x50], rsi
0xd6cf: call    0x1d690
0xd6d4: mov     qword ptr [r12 + 0x58], rax
0xd6d9: mov     r12, qword ptr [rip + 0x1dd28]
0xd6e0: mov     rax, qword ptr [r12 + 0x38]
0xd6e5: lea     rdx, [rax + 1]
0xd6e9: jmp     0xd64a
0xd64a
0xd6ee: nop
0xd6f0: ret
0xd6f1: nop     word ptr cs:[rax + rax]
0xd6fc: nop     dword ptr [rax]
0xd700: endbr64
0xd704: mov     rax, qword ptr [rip + 0x1dcfd]
0xd70b: mov     rdi, qword ptr [rax + 0x60]
0xd70f: test    rdi, rdi
0xd712: je      0xd730
0xd714: sub     rsp, 8

```



```

0xd718: call    0xd610
0xd71d: mov     rax, qword ptr [rip + 0x1dce4]
0xd724: mov     dword ptr [rax + 0x10], 0x1a
0xd72b: add     rsp, 8
0xd72f: ret
0xd730: ret
0xd731: nop     word ptr cs:[rax + rax]
0xd73c: nop     dword ptr [rax]
0xd740: endbr64
0xd744: mov     rax, qword ptr [rip + 0x1dcbd]
0xd74b: push    r12
0xd74d: push    rbp
0xd74e: push    rbx
0xd74f: mov     rbx, qword ptr [rax + 0x40]
0xd753: mov     r12, qword ptr [rax + 0x28]
0xd757: mov     eax, dword ptr [rax + 0xc]
0xd75a: lea     rbp, [rbx + 3]
0xd75e: test    eax, eax
0xd760: je      0xd7f0
0xd766: mov     edx, dword ptr [rip + 0x1caa4]
0xd76c: mov     esi, dword ptr [rip + 0x1ca9a]
0xd772: mov     edi, dword ptr [rip + 0x1ca9c]
0xd778: call    0xa570
0xd77d: mov     rcx, qword ptr [rip + 0x1dc84]
0xd784: cmp     r12, rbp
0xd787: jb      0xd810
0xd78d: mov     rsi, qword ptr [rcx + 0x18]
0xd791: mov     rdx, qword ptr [rcx + 0x40]
0xd795: xor     ebx, ebx
0xd797: mov     rdi, qword ptr [rcx + 0x28]
0xd79b: mov     rax, rdi
0xd79e: sub     rax, rdx
0xd7a1: mov     edx, 1
0xd7a6: mov     r8, rax
0xd7a9: shr     r8, 2
0xd7ad: cmp     rax, 3
0xd7b1: cmova   edx, r8d
0xd7b5: xor     eax, eax
0xd7b7: cmp     rdi, rsi
0xd7ba: ja      0xd7d3
0xd7bc: mov     rax, rsi
0xd7bf: sub     rax, rdi
0xd7c2: movsxd  rdi, edx
0xd7c5: lea     r8d, [rdx + rax]
0xd7c9: xor     edx, edx
0xd7cb: div     rdi
0xd7ce: mov     eax, r8d
0xd7d1: sub     eax, edx

```

```

0xd7d3: mov     edx, esi
0xd7d5: mov     rdi, qword ptr [rcx]
0xd7d8: xor     esi, esi
0xd7da: sub     edx, eax
0xd7dc: call    0xc680
0xd7e1: cmp     r12, rbp
0xd7e4: jnb     0xd840
0xd7e6: pop     rbx
0xd7e7: pop     rbp
0xd7e8: pop     r12
0xd7ea: ret
0xd7eb: nop     dword ptr [rax + rax]
0xd7f0: mov     edx, dword ptr [rip + 0x1ca0e]
0xd7f6: mov     esi, dword ptr [rip + 0x1ca04]
0xd7fc: mov     edi, dword ptr [rip + 0x1ca06]
0xd802: jmp     0xd778
0xd778:
0xd807: nop     word ptr [rax + rax]
0xd810: mov     rbx, qword ptr [rcx + 0x40]
0xd814: mov     rsi, qword ptr [rcx + 0x18]
0xd818: mov     qword ptr [rcx + 0x40], 0
0xd820: xor     edx, edx
0xd822: add     qword ptr [rcx + 0x58], rbx
0xd826: sub     rsi, rbx
0xd829: sub     qword ptr [rcx + 0x38], rbx
0xd82d: sub     qword ptr [rcx + 0x20], rbx
0xd831: movsxd  rbx, ebx
0xd834: mov     qword ptr [rcx + 0x18], rsi
0xd838: jmp     0xd797
0xd797:
0xd83d: nop     dword ptr [rax]
0xd840: mov     rax, qword ptr [rip + 0x1dbc1]
0xd847: movsxd  rbx, ebx
0xd84a: mov     qword ptr [rax + 0x40], rbx
0xd84e: sub     qword ptr [rax + 0x58], rbx
0xd852: add     qword ptr [rax + 0x38], rbx
0xd856: add     qword ptr [rax + 0x18], rbx
0xd85a: add     qword ptr [rax + 0x20], rbx
0xd85e: pop     rbx
0xd85f: pop     rbp
0xd860: pop     r12
0xd862: ret
0xd863: nop     word ptr cs:[rax + rax]
0xd86e: nop
0xd870: endbr64
0xd874: push    r15
0xd876: push    r14
0xd878: push    r13

```

```

0xd87a: push    r12
0xd87c: push    rbp
0xd87d: push    rbx
0xd87e: sub     rsp, 0x28
0xd882: mov     rax, qword ptr fs:[0x28]
0xd88b: mov     qword ptr [rsp + 0x18], rax
0xd890: mov     rax, qword ptr [rip + 0x1db71]
0xd897: lea     r14, [rsp + 0x17]
0xd89c: mov     rdi, r14
0xd89f: mov     rbx, qword ptr [rax + 0x40]
0xd8a3: mov     r13, qword ptr [rax + 0x28]
0xd8a7: call    0xa6a0
0xd8ac: mov     rax, qword ptr [rip + 0x1db55]
0xd8b3: add     rbx, 3
0xd8b7: mov     ecx, dword ptr [rax + 0xc]
0xd8ba: test    ecx, ecx
0xd8bc: je      0xda20
0xd8c2: mov     edx, dword ptr [rip + 0x1c948]
0xd8c8: mov     esi, dword ptr [rip + 0x1c93e]
0xd8ce: mov     edi, dword ptr [rip + 0x1c940]
0xd8d4: call    0xa570
0xd8d9: mov     rdx, qword ptr [rip + 0x1db28]
0xd8e0: cmp     r13, rbx
0xd8e3: jb      0xda90
0xd8e9: mov     qword ptr [rsp + 8], 0
0xd8f2: mov     rax, qword ptr [rdx + 0x18]
0xd8f6: mov     rcx, qword ptr [rdx + 0x40]
0xd8fa: mov     rdi, qword ptr [rdx + 0x28]
0xd8fe: mov     r15, rdi
0xd901: sub     r15, rcx
0xd904: cmp     r15, 3
0xd908: jbe     0xda80
0xd90e: shr     r15, 2
0xd912: movsxd  rcx, r15d
0xd915: cmp     rdi, rax
0xd918: ja      0xda70
0xd91e: mov     edx, eax
0xd920: sub     rax, rdi
0xd923: sub     edx, edi
0xd925: add     r15d, edx
0xd928: xor     edx, edx
0xd92a: div     rcx
0xd92d: sub     r15d, edx
0xd930: mov     r12d, r15d
0xd933: call    0x1d670
0xd938: mov     esi, 0x20
0xd93d: mov     rbp, rax
0xd940: mov     rax, qword ptr [rip + 0x1dac1]

```

```

0xd947: mov     rdi, rbp
0xd94a: mov     rdx, qword ptr [rax + 0x28]
0xd94e: call    0x4a50
0xd953: mov     rax, qword ptr [rip + 0x1daae]
0xd95a: mov     edx, dword ptr [rax + 8]
0xd95d: test    edx, edx
0xd95f: je      0xda40
0xd965: mov     rdx, qword ptr [rax + 0x38]
0xd969: sub     rdx, qword ptr [rax + 0x40]
0xd96d: mov     rdi, rbp
0xd970: sub     rdx, r12
0xd973: mov     rsi, qword ptr [rax + 0x28]
0xd977: sub     rsi, qword ptr [rax + 0x40]
0xd97b: cmp     rdx, rsi
0xd97e: cmova   rdx, rsi
0xd982: mov     rsi, qword ptr [rax + 0x40]
0xd986: add     rsi, r12
0xd989: add     rsi, qword ptr [rax + 0x58]
0xd98d: call    0x4bb0
0xd992: mov     rax, qword ptr [rip + 0x1da6f]
0xd999: xor     edx, edx
0xd99b: xor     esi, esi
0xd99d: mov     rdi, qword ptr [rax]
0xd9a0: call    0xc680
0xd9a5: cmp     r13, rbx
0xd9a8: jae     0xdaf0
0xd9ae: mov     rax, qword ptr [rip + 0x1da53]
0xd9b5: mov     rsi, rbp
0xd9b8: mov     rdi, qword ptr [rax]
0xd9bb: mov     rdx, qword ptr [rax + 0x28]
0xd9bf: sub     edx, dword ptr [rax + 0x40]
0xd9c2: call    0xc600
0xd9c7: mov     rax, qword ptr [rip + 0x1da3a]
0xd9ce: xor     esi, esi
0xd9d0: mov     edx, dword ptr [rax + 0x18]
0xd9d3: mov     rdi, qword ptr [rax]
0xd9d6: sub     edx, r15d
0xd9d9: call    0xc680
0xd9de: cmp     r13, rbx
0xd9e1: jnb     0xdac8
0xd9e7: mov     rdi, rbp
0xd9ea: call    0x18d10
0xd9ef: mov     rdi, r14
0xd9f2: call    0xa6b0
0xd9f7: mov     rax, qword ptr [rsp + 0x18]
0xd9fc: sub     rax, qword ptr fs:[0x28]
0xda05: jne     0xdb0b
0xda0b: add     rsp, 0x28

```

```

0xda0f: pop    rbx
0xda10: pop    rbp
0xda11: pop    r12
0xda13: pop    r13
0xda15: pop    r14
0xda17: pop    r15
0xda19: ret
0xda1a: nop    word ptr [rax + rax]
0xda20: mov    edx, dword ptr [rip + 0x1c7de]
0xda26: mov    esi, dword ptr [rip + 0x1c7d4]
0xda2c: mov    edi, dword ptr [rip + 0x1c7d6]
0xda32: jmp    0xd8d4
0xd8d4
0xda37: nop    word ptr [rax + rax]
0xda40: mov    rsi, qword ptr [rax + 0x40]
0xda44: mov    rdx, qword ptr [rax + 0x38]
0xda48: mov    rdi, rbp
0xda4b: mov    rax, qword ptr [rax + 0x28]
0xda4f: sub    rdx, rsi
0xda52: sub    rdx, r12
0xda55: sub    rax, rsi
0xda58: mov    esi, 0x2a
0xda5d: cmp    rdx, rax
0xda60: cmova  rdx, rax
0xda64: call  0x4a50
0xda69: jmp    0xd992
0xd992
0xda6e: nop
0xda70: xor    r12d, r12d
0xda73: xor    r15d, r15d
0xda76: jmp    0xd933
0xd933
0xda7b: nop    dword ptr [rax + rax]
0xda80: mov    ecx, 1
0xda85: mov    r15d, 1
0xda8b: jmp    0xd915
0xd915
0xda90: mov    r12, qword ptr [rdx + 0x40]
0xda94: mov    rax, qword ptr [rdx + 0x18]
0xda98: mov    qword ptr [rdx + 0x40], 0
0xdaa0: add    qword ptr [rdx + 0x58], r12
0xdaa4: movsxd rcx, r12d
0xdaa7: sub    rax, r12
0xdaaa: sub    qword ptr [rdx + 0x38], r12
0xdaae: mov    qword ptr [rsp + 8], rcx
0xdab3: xor    ecx, ecx
0xdab5: sub    qword ptr [rdx + 0x20], r12
0xdab9: mov    qword ptr [rdx + 0x18], rax

```

```

0xdabd: jmp      0xd8fa
0xd8fa
0xdac2: nop      word ptr [rax + rax]
0xdac8: mov      rax, qword ptr [rip + 0x1d939]
0xdacf: movsxd   r12, dword ptr [rsp + 8]
0xdad4: sub      qword ptr [rax + 0x58], r12
0xdad8: add      qword ptr [rax + 0x38], r12
0xdadc: add      qword ptr [rax + 0x18], r12
0xdae0: add      qword ptr [rax + 0x20], r12
0xdae4: mov      qword ptr [rax + 0x40], r12
0xdae8: jmp      0xd9e7
0xd9e7
0xdaed: nop      dword ptr [rax]
0xdaf0: mov      rax, qword ptr [rip + 0x1d911]
0xdaf7: mov      rsi, qword ptr [rax + 0x58]
0xdafb: mov      edx, dword ptr [rax + 0x40]
0xdafe: mov      rdi, qword ptr [rax]
0xdb01: call     0xc600
0xdb06: jmp      0xd9ae
0xd9ae
0xdb0b: call     0x49d0
0xdb10: endbr64
0xdb14: mov      rax, qword ptr [rip + 0x1d8ed]
0xdb1b: push     rbx
0xdb1c: mov      rbx, rdi
0xdb1f: mov      rdi, qword ptr [rdi]
0xdb22: mov      rsi, qword ptr [rax + 0x48]
0xdb26: add      rsi, 1
0xdb2a: call     0x1d690
0xdb2f: mov      qword ptr [rbx], rax
0xdb32: mov      rdi, rax
0xdb35: mov      rax, qword ptr [rip + 0x1d8cc]
0xdb3c: mov      rcx, qword ptr [rax + 0x48]
0xdb40: mov      rsi, qword ptr [rax + 0x40]
0xdb44: add      rsi, qword ptr [rax + 0x58]
0xdb48: lea      rdx, [rcx + 1]
0xdb4c: call     0x4bb0
0xdb51: mov      rax, qword ptr [rip + 0x1d8b0]
0xdb58: pop      rbx
0xdb59: mov      eax, dword ptr [rax + 0x48]
0xdb5c: ret
0xdb5d: nop      dword ptr [rax]
0xdb60: endbr64
0xdb64: push     rbp
0xdb65: xor      eax, eax
0xdb67: mov      rbp, rdi
0xdb6a: call     0xcff0
0xdb6f: test     rbp, rbp

```

```

0xdb72: lea      rax, [rip + 0x133bf]
0xdb79: cmov     rbp, rax
0xdb7d: mov      rdi, rbp
0xdb80: call     0xd540
0xdb85: xor      eax, eax
0xdb87: call     0xd870
0xdb8c: xor      eax, eax
0xdb8e: pop      rbp
0xdb8f: jmp      0xd740
0xd740:
0xdb94: nop      word ptr cs:[rax + rax]
0xdb9f: nop
0xdba0: endbr64
0xdba4: mov      rax, qword ptr [rip + 0x1d85d]
0xdbab: mov      dword ptr [rax + 0xc], edi
0xdbae: ret
0xdbaf: nop
0xdbb0: endbr64
0xdbb4: push     r12
0xdbb6: push     rbp
0xdbb7: push     rbx
0xdbb8: cmp      edi, 1
0xdbbb: je       0xdbd0
0xdbbd: cmp      edi, 2
0xdbc0: je       0xdc90
0xdbc6: test     edi, edi
0xdbc8: je       0xdc40
0xdbca: pop      rbx
0xdbcb: pop      rbp
0xdbcc: pop      r12
0xdbce: ret
0xdbcf: nop
0xdbd0: mov      ecx, dword ptr [rip + 0x1d83e]
0xdbd6: test     ecx, ecx
0xdbd8: je       0xdd38
0xdbde: xor      eax, eax
0xdbe0: call     0x1ab10
0xdbe5: mov      rbp, rax
0xdbe8: test     rax, rax
0xdbeb: je       0xdbca
0xdbed: mov      r12, qword ptr [rbp]
0xdbf1: mov      rbx, qword ptr [rip + 0x1d810]
0xdbf8: mov      rdi, r12
0xdbfb: call     0x49b0
0xdc00: mov      rdi, qword ptr [rbx + 0x40]
0xdc04: mov      qword ptr [rbx + 0x48], rax
0xdc08: lea      rsi, [rax + rdi]
0xdc0c: mov      qword ptr [rbx + 0x38], rsi

```

```

0xdc10: mov     qword ptr [rbx + 0x18], rsi
0xdc14: add     rsi, 1
0xdc18: cmp     rsi, qword ptr [rbx + 0x50]
0xdc1c: ja      0xdd70
0xdc22: add     rdi, qword ptr [rbx + 0x58]
0xdc26: mov     rsi, r12
0xdc29: call    0x48b0
0xdc2e: pop     rbx
0xdc2f: xor     eax, eax
0xdc31: pop     rbp
0xdc32: pop     r12
0xdc34: jmp     0xd870
0xd870:
0xdc39: nop     dword ptr [rax]
0xdc40: mov     eax, dword ptr [rip + 0x1d7ce]
0xdc46: test    eax, eax
0xdc48: jne     0xdd18
0xdc4e: xor     eax, eax
0xdc50: call    0x1ab10
0xdc55: test    rax, rax
0xdc58: je      0xdd98
0xdc5e: mov     rdx, qword ptr [rip + 0x1d7a3]
0xdc65: mov     rsi, qword ptr [rax]
0xdc68: mov     rbp, qword ptr [rdx + 0x40]
0xdc6c: add     rbp, qword ptr [rdx + 0x58]
0xdc70: mov     rdi, rbp
0xdc73: call    0x4b30
0xdc78: test    eax, eax
0xdc7a: jne     0xdd58
0xdc80: pop     rbx
0xdc81: xor     eax, eax
0xdc83: pop     rbp
0xdc84: pop     r12
0xdc86: jmp     0x1ab40
0x1ab40:
0xdc8b: nop     dword ptr [rax + rax]
0xdc90: mov     edx, dword ptr [rip + 0x1d77e]
0xdc96: test    edx, edx
0xdc98: je      0xdcf8
0xdc9a: xor     eax, eax
0xdc9c: call    0x1ab40
0xdca1: mov     rbp, rax
0xdca4: test    rax, rax
0xdca7: jne     0xdbed
0xdcad: mov     rax, qword ptr [rip + 0x1d754]
0xdcb4: mov     edi, 2
0xdcb9: mov     rdx, qword ptr [rax + 0x40]
0xdcbd: mov     dword ptr [rax + 0x10], 6

```



```

0xdcc4: mov     qword ptr [rax + 0x20], rdx
0xdcc8: mov     rdx, qword ptr [rax + 0x38]
0xdccc: mov     qword ptr [rax + 0x18], rdx
0xdcd0: xor     eax, eax
0xdcd2: call    0xc6e0
0xdcd7: mov     rax, qword ptr [rip + 0x1d72a]
0xdcde: mov     rdx, qword ptr [rax + 0x18]
0xdce2: mov     dword ptr [rax + 0x10], 0xd
0xdce9: mov     qword ptr [rax + 0x20], rdx
0xdc2e: jmp     0xdc2e
0xdc2e
0xdcf2: nop     word ptr [rax + rax]
0xdcf8: mov     dword ptr [rip + 0x1d712], 1
0xdd02: xor     eax, eax
0xdd04: call    0x1a570
0xdd09: mov     dword ptr [rip + 0x1d701], eax
0xdd0f: jmp     0xdc9a
0xdc9a
0xdd11: nop     dword ptr [rax]
0xdd18: mov     edi, dword ptr [rip + 0x1d6f2]
0xdd1e: xor     eax, eax
0xdd20: call    0x1ac30
0xdd25: mov     dword ptr [rip + 0x1d6e5], 0
0xdd2f: jmp     0xdc4e
0xdc4e
0xdd34: nop     dword ptr [rax]
0xdd38: mov     dword ptr [rip + 0x1d6d2], 1
0xdd42: xor     eax, eax
0xdd44: call    0x1a570
0xdd49: mov     dword ptr [rip + 0x1d6c1], eax
0xdd4f: jmp     0xdbde
0xdbde
0xdd54: nop     dword ptr [rax]
0xdd58: xor     eax, eax
0xdd5a: mov     rdi, rbp
0xdd5d: call    0x1a380
0xdd62: xor     eax, eax
0xdd64: call    0x1ab40
0xdd69: jmp     0xdc80
0xdc80
0xdd6e: nop
0xdd70: mov     qword ptr [rbx + 0x50], rsi
0xdd74: mov     rdi, qword ptr [rbx + 0x58]
0xdd78: call    0x1d690
0xdd7d: mov     qword ptr [rbx + 0x58], rax
0xdd81: mov     rbx, qword ptr [rip + 0x1d680]
0xdd88: mov     r12, qword ptr [rbp]
0xdd8c: mov     rdi, qword ptr [rbx + 0x40]

```

```

0xdd90: jmp      0xdc22
0xdc22
0xdd95: nop      dword ptr [rax]
0xdd98: mov      rax, qword ptr [rip + 0x1d669]
0xdd9f: mov      rdi, qword ptr [rax + 0x40]
0xdda3: add      rdi, qword ptr [rax + 0x58]
0xdda7: xor      eax, eax
0xdda9: call     0x1a380
0xddae: jmp      0xdc80
0xdc80
0xddb3: nop      word ptr cs:[rax + rax]
0xddbd: nop      dword ptr [rax]
0xddc0: endbr64
0xddc4: mov      eax, dword ptr [rip + 0x1be96]
0xddca: push     rbp
0xddcb: mov      rbp, rdi
0xddce: lea      rdi, [rip + 0x1be73]
0xddd5: test     eax, eax
0xddd7: lea      rax, [rip + 0x1be52]
0xddde: cmove    rdi, rax
0xdde2: call     0x17c50
0xdde7: mov      edx, 9
0xddec: lea      rsi, [rip + 0x1c4ad]
0xddf3: lea      rdi, [rip + 0x1c466]
0xddfa: call     0x17f70
0xddff: mov      rdi, rbp
0xde02: call     0x1d7a0
0xde07: mov      rdi, rax
0xde0a: mov      rbp, rax
0xde0d: mov      qword ptr [rip + 0x1d794], rax
0xde14: call     0x49b0
0xde19: mov      rdi, rbp
0xde1c: mov      rsi, rax
0xde1f: call     0x184d0
0xde24: xor      eax, eax
0xde26: call     0xc590
0xde2b: lea      rdi, [rip + 0x1d5ee]
0xde32: pop      rbp
0xde33: mov      qword ptr [rip + 0x1d78e], rax
0xde3a: jmp      0x4a30
0x4a30
0xde3f: nop
0xde40: endbr64
0xde44: mov      rdi, qword ptr [rip + 0x1d77d]
0xde4b: jmp      0xc5c0
0xc5c0
0xde50: endbr64
0xde54: push     rbp

```

```

0xde55: mov     rbp, rdi
0xde58: push    rbx
0xde59: mov     rbx, rsi
0xde5c: sub     rsp, 8
0xde60: mov     rdi, qword ptr [rip + 0x1d749]
0xde67: test    rdi, rdi
0xde6a: je      0xde71
0xde6c: call    0x18d10
0xde71: mov     rdi, rbp
0xde74: call    0x1d670
0xde79: mov     rdi, qword ptr [rip + 0x1d748]
0xde80: mov     edx, ebx
0xde82: xor     esi, esi
0xde84: mov     qword ptr [rip + 0x1d725], rax
0xde8b: add     rsp, 8
0xde8f: mov     r8d, ebp
0xde92: mov     ecx, 1
0xde97: pop     rbx
0xde98: pop     rbp
0xde99: jmp     0xc5e0
0xc5e0
0xde9e: nop
0xdea0: endbr64
0xdea4: push    rbp
0xdea5: mov     rbp, rsp
0xdea8: push    r15
0xdeaa: push    r14
0xdeac: push    r13
0xdeae: push    r12
0xdeb0: push    rbx
0xdeb1: sub     rsp, 0x68
0xdeb5: mov     rax, qword ptr fs:[0x28]
0xdeb6: mov     qword ptr [rbp - 0x38], rax
0xdec2: xor     eax, eax
0xdec4: lea     rax, [rbp - 0x61]
0xdec8: mov     rdi, rax
0xdecb: mov     qword ptr [rbp - 0x80], rax
0xdecf: call    0xa6a0
0xded4: mov     rax, qword ptr [rip + 0x1d6ed]
0xdedb: mov     esi, 0x20
0xdee0: mov     rdi, qword ptr [rip + 0x1d6c9]
0xdee7: movsxd  rdx, dword ptr [rax + 0xc]
0xdeeb: call    0x4a50
0xdef0: mov     rdi, qword ptr [rip + 0x1d6c9]
0xdef7: call    0x49b0
0xdefc: lea     r14, [rax + 1]
0xdf00: mov     rdi, r14
0xdf03: call    0x1d670

```

```

0xdf08: mov     rcx, qword ptr [rip + 0x1d6b1]
0xdf0f: mov     r13, rax
0xdf12: movzx   eax, byte ptr [rcx]
0xdf15: test    al, al
0xdf17: je      0xe430
0xdf1d: xor     esi, esi
0xdf1f: xor     ebx, ebx
0xdf21: lea     r15, [rip + 0x13188]
0xdf28: xor     edx, edx
0xdf2a: mov     dword ptr [rbp - 0x74], esi
0xdf2d: jmp     0xdf48
0xdf48:
0xdf2f: nop
0xdf30: cmp     al, 9
0xdf32: je      0xdf80
0xdf34: add     dword ptr [rbp - 0x74], 1
0xdf38: mov     byte ptr [r13 + r12], al
0xdf3d: movsxd  rdx, ebx
0xdf40: movzx   eax, byte ptr [rcx + rdx]
0xdf44: test    al, al
0xdf46: je      0xdfc0
0xdf48: movsxd  r12, dword ptr [rbp - 0x74]
0xdf4c: add     ebx, 1
0xdf4f: cmp     al, 0x5c
0xdf51: jne     0xdf30
0xdf53: lea     rdx, [rcx + rdx + 1]
0xdf58: movzx   eax, byte ptr [rdx]
0xdf5b: test    al, al
0xdf5d: je      0xe320
0xdf63: sub     eax, 0x5c
0xdf66: cmp     al, 0x17
0xdf68: ja      0xe0d8
0xdf6e: movzx   eax, al
0xdf71: movsxd  rax, dword ptr [r15 + rax*4]
0xdf75: add     rax, r15
0xdf78: jmp     rax
rax

```

->->->

indirect JMP

```

0x107e8: mov     rdx, qword ptr [r13 + 0x18]
0x107ec: mov     byte ptr [rsp + 8], 0
0x107f1: cmp     byte ptr [rdx], 0x2f
0x107f4: je      0x10c18
0x107fa: mov     r15, qword ptr [rax]
0x107fd: mov     rdi, r15
0x10800: call    0x49b0
0x10805: movsxd  rdx, ebp
0x10808: mov     rbp, rdx
0x1080b: sub     rbp, r14

```

```

0x1080e:    cmp     rax, rbp
0x10811:    jbe     0x1082b
0x10813:    movsxd  rcx, dword ptr [r13 + 0x74]
0x10817:    lea     rsi, [rbp + rcx]
0x1081c:    cmp     rax, rsi
0x1081f:    jb      0x10a90
0x10825:    sub     rax, rcx
0x10828:    add     r15, rcx
0x1082b:    cmp     rbp, rax
0x1082e:    mov     rsi, r15
0x10831:    cmova   rbp, rax
0x10835:    mov     rax, qword ptr [r13 + 0x20]
0x10839:    mov     rdx, rbp
0x1083c:    lea     rdi, [rax + 1]
0x10840:    call    0x4bb0
0x10845:    mov     rax, qword ptr [r13 + 0x20]
0x10849:    mov     rsi, rbp
0x1084c:    lea     rdi, [rax + 1]
0x10850:    call    0x184d0
0x10855:    movzx   ecx, byte ptr [rsp + 8]
0x1085a:    test    cl, cl
0x1085c:    je      0x10866
0x1085e:    mov     rax, qword ptr [r13 + 0x20]
0x10862:    mov     byte ptr [rax + rbp + 1], cl
0x10866:    mov     eax, dword ptr [r13 + 0x64]
0x1086a:    cmp     eax, 0x27
0x1086d:    jle     0x108a0
0x1086f:    mov     edx, dword ptr [r13 + 0x80]
0x10876:    cmp     edx, 4
0x10879:    je      0x108a0
0x1087b:    jg      0x109e0
0x10881:    cmp     edx, 1
0x10884:    jg      0x109c0
0x1088a:    test    edx, edx
0x1088c:    jns     0x10c30
0x10892:    lea     rdi, [rip + 0x10953]
0x10899:    call    0x8b00
0x1089e:    nop
0x108a0:    mov     rax, qword ptr [r13 + 0x38]
0x108a4:    cmp     byte ptr [rax + r12 + 0x49], 0
0x108aa:    je      0x108b9
0x108ac:    movsxd  rax, dword ptr [r13 + 0x64]
0x108b0:    mov     rdx, qword ptr [r13 + 0x20]
0x108b4:    mov     byte ptr [rdx + rax - 3], 0x2a
0x108b9:    cmp     ebx, dword ptr [r13 + 0x28]
0x108bd:    je      0x10948
0x108c3:    mov     rax, qword ptr [r13 + 0x38]
0x108c7:    mov     ecx, dword ptr [rip + 0x1938f]

```

```

0x108cd:    movzx    edx, byte ptr [rax + r12 + 0x49]
0x108d3:    test     ecx, ecx
0x108d5:    je       0x108e8
0x108d7:    add      rax, r12
0x108da:    cmp      byte ptr [rax + 0x48], 0
0x108de:    movzx    edx, byte ptr [rax + 0x49]
0x108e2:    jne      0x1097e
0x108e8:    test     dl, dl
0x108ea:    jne      0x109a0
0x108f0:    mov      esi, dword ptr [rip + 0x19a46]
0x108f6:    mov      r9d, dword ptr [rip + 0x19a27]
0x108fd:    mov      edi, dword ptr [rip + 0x19a3d]
0x10903:    mov      edx, r9d
0x10906:    call     0xa570
0x1090b:    sub      ebx, dword ptr [r13 + 0x2c]
0x1090f:    mov      rdi, qword ptr [r13 + 8]
0x10913:    mov      edx, 1
0x10918:    lea      esi, [rbx + 1]
0x1091b:    call     0xc680
0x10920:    mov      eax, dword ptr [r13 + 0x64]
0x10924:    mov      rsi, qword ptr [r13 + 0x20]
0x10928:    mov      rdi, qword ptr [r13 + 8]
0x1092c:    add      rsp, 0x28
0x10930:    pop      rbx
0x10931:    lea      edx, [rax - 2]
0x10934:    pop      rbp
0x10935:    pop      r12
0x10937:    pop      r13
0x10939:    pop      r14
0x1093b:    pop      r15
0x1093d:    jmp      0xc600
0xc600
0x10942:    nop      word ptr [rax + rax]
0x10948:    cmp      byte ptr [r13 + 0x6c], 1
0x1094d:    mov      rdx, qword ptr [r13 + 0x20]
0x10951:    sbb      eax, eax
0x10953:    and      eax, 0xffffffffec
0x10956:    add      eax, 0x3e
0x10959:    mov      byte ptr [rdx], al
0x1095b:    mov      rax, qword ptr [r13 + 0x38]
0x1095f:    mov      esi, dword ptr [rip + 0x192f7]
0x10965:    add      rax, r12
0x10968:    test     esi, esi
0x1096a:    je       0x10a50
0x10970:    cmp      byte ptr [rax + 0x48], 0
0x10974:    movzx    edx, byte ptr [rax + 0x49]
0x10978:    je       0x10a50
0x1097e:    test     dl, dl

```

```

0x10980:      je      0x10b10
0x10986:      cmp     ebx, dword ptr [r13 + 0x28]
0x1098a:      jne     0x109a0
0x1098c:      cmp     byte ptr [r13 + 0x6c], 1
0x10991:      je      0x10c08
0x10997:      nop     word ptr [rax + rax]
0x109a0:      mov     esi, dword ptr [rip + 0x1998e]
0x109a6:      mov     r9d, dword ptr [rip + 0x19977]
0x109ad:      mov     edi, dword ptr [rip + 0x19985]
0x109b3:      jmp     0x10903
0x10903:
0x109b8:      nop     dword ptr [rax + rax]
0x109c0:      mov     ecx, 0xb
0x109c5:      mov     esi, ebx
0x109c7:      mov     rdi, r13
0x109ca:      xor     eax, eax
0x109cc:      call    0x10420
0x109d1:      jmp     0x108a0
0x108a0:
0x109d6:      nop     word ptr cs:[rax + rax]
0x109e0:      cmp     edx, 5
0x109e3:      jne     0x10892
0x109e9:      cmp     eax, 0x49
0x109ec:      jle     0x108a0
0x109f2:      xor     edx, edx
0x109f4:      mov     esi, ebx
0x109f6:      mov     rdi, r13
0x109f9:      xor     eax, eax
0x109fb:      mov     ecx, 0x36
0x10a00:      call    0x10420
0x10a05:      mov     esi, ebx
0x10a07:      mov     rdi, r13
0x10a0a:      xor     eax, eax
0x10a0c:      mov     ecx, 0x26
0x10a11:      mov     edx, 1
0x10a16:      call    0x10420
0x10a1b:      mov     esi, ebx
0x10a1d:      mov     rdi, r13
0x10a20:      xor     eax, eax
0x10a22:      mov     ecx, 0x16
0x10a27:      mov     edx, 2
0x10a2c:      call    0x10420
0x10a31:      mov     ecx, 0xb
0x10a36:      mov     esi, ebx
0x10a38:      mov     rdi, r13
0x10a3b:      mov     edx, 3
0x10a40:      xor     eax, eax
0x10a42:      call    0x10420

```

```

0x10a47:      jmp      0x108a0
0x108a0
0x10a4c:      nop      dword ptr [rax]
0x10a50:      movzx   edx, byte ptr [rax + 0x49]
0x10a54:      cmp      ebx, dword ptr [r13 + 0x28]
0x10a58:      jne      0x108e8
0x10a5e:      cmp      byte ptr [r13 + 0x6c], 1
0x10a63:      jne      0x108e8
0x10a69:      mov      esi, dword ptr [rip + 0x198d9]
0x10a6f:      test     dl, dl
0x10a71:      jne      0x10c08
0x10a77:      mov      r9d, dword ptr [rip + 0x198ce]
0x10a7e:      test     dl, dl
0x10a80:      je       0x108fd
0x10a86:      jmp      0x109ad
0x109ad
0x10a8b:      nop      dword ptr [rax + rax]
0x10a90:      add      r14, rax
0x10a93:      mov      rcx, rdx
0x10a96:      sub      rcx, r14
0x10a99:      sub      r14, rdx
0x10a9c:      add      rax, rcx
0x10a9f:      add      r15, r14
0x10aa2:      jmp      0x1082b
0x1082b
0x10aa7:      nop      word ptr [rax + rax]
0x10ab0:      mov      byte ptr [rsp + 8], 0x3d
0x10ab5:      add      r14, 1
0x10ab9:      jmp      0x107fa
0x107fa
0x10abe:      nop
0x10ac0:      mov      byte ptr [rsp + 8], 0x7c
0x10ac5:      jmp      0x10ab5
0x10ab5
0x10ac7:      nop      word ptr [rax + rax]
0x10ad0:      movzx   ecx, byte ptr [rax + 0x4a]
0x10ad4:      mov      byte ptr [rsp + 8], cl
0x10ad8:      test     cl, cl
0x10ada:      je       0x107fa
0x10ae0:      mov      byte ptr [rsp + 8], 0x2a
0x10ae5:      jmp      0x10ab5
0x10ab5
0x10ae7:      nop      word ptr [rax + rax]
0x10af0:      mov      byte ptr [rsp + 8], 0x40
0x10af5:      jmp      0x10ab5
0x10ab5
0x10af7:      nop      word ptr [rax + rax]
0x10b00:      mov      byte ptr [rsp + 8], 0x2f

```



```

0x10b05:      jmp      0x10ab5
0x10ab5
0x10b07:      nop      word ptr [rax + rax]
0x10b10:      mov      rdi, qword ptr [r13 + 8]
0x10b14:      mov      esi, ebx
0x10b16:      sub      esi, dword ptr [r13 + 0x2c]
0x10b1a:      mov      edx, 1
0x10b1f:      add      esi, 1
0x10b22:      call     0xc680
0x10b27:      mov      rax, qword ptr [r13 + 0x38]
0x10b2b:      cmp      byte ptr [rax + r12 + 0x49], 0
0x10b31:      jne      0x10c50
0x10b37:      mov      eax, dword ptr [rip + 0x197ff]
0x10b3d:      mov      r15d, dword ptr [rip + 0x197fc]
0x10b44:      mov      dword ptr [rsp + 0xc], eax
0x10b48:      cmp      ebx, dword ptr [r13 + 0x28]
0x10b4c:      je       0x10c70
0x10b52:      mov      eax, dword ptr [rip + 0x197cc]
0x10b58:      mov      dword ptr [rsp + 8], eax
0x10b5c:      mov      edx, dword ptr [rsp + 8]
0x10b60:      mov      esi, dword ptr [rsp + 0xc]
0x10b64:      mov      edi, r15d
0x10b67:      call     0xa570
0x10b6c:      mov      rax, qword ptr [r13 + 0x20]
0x10b70:      mov      rdi, qword ptr [r13 + 8]
0x10b74:      movsx     esi, byte ptr [rax]
0x10b77:      call     0xc650
0x10b7c:      mov      rax, qword ptr [r13 + 0x38]
0x10b80:      lea      rdx, [rax + r12]
0x10b84:      cmp      byte ptr [rdx + 0x4b], 0
0x10b88:      je       0x10c90
0x10b8e:      movsx     edi, byte ptr [rdx + 0x4e]
0x10b92:      cmp      dil, 2
0x10b96:      je       0x10bb6
0x10b98:      movsx     esi, byte ptr [rdx + 0x4c]
0x10b9c:      cmp      ebx, dword ptr [r13 + 0x28]
0x10ba0:      jne      0x10bad
0x10ba2:      cmp      byte ptr [r13 + 0x6c], 1
0x10ba7:      je       0x10d52
0x10bad:      movsx     edx, byte ptr [rdx + 0x4d]
0x10bb1:      call     0xa570
0x10bb6:      mov      rax, qword ptr [r13 + 0x20]
0x10bba:      mov      rdi, qword ptr [r13 + 8]
0x10bbe:      lea      edx, [rbp + 1]
0x10bc1:      lea      rsi, [rax + 1]
0x10bc5:      call     0xc600
0x10bca:      mov      eax, dword ptr [r13 + 0x64]
0x10bce:      mov      rdi, qword ptr [r13 + 8]

```

```

0x10bd2:    lea     edx, [rax - 4]
0x10bd5:    mov     rax, qword ptr [r13 + 0x20]
0x10bd9:    sub     edx, ebp
0x10bdb:    lea     rsi, [rax + rbp + 2]
0x10be0:    call    0xc600
0x10be5:    mov     edx, dword ptr [rsp + 8]
0x10be9:    mov     esi, dword ptr [rsp + 0xc]
0x10bed:    mov     edi, r15d
0x10bf0:    add     rsp, 0x28
0x10bf4:    pop     rbx
0x10bf5:    pop     rbp
0x10bf6:    pop     r12
0x10bf8:    pop     r13
0x10bfa:    pop     r14
0x10bfc:    pop     r15
0x10bfe:    jmp     0xa570
0xa570:
0x10c03:    nop     dword ptr [rax + rax]
0x10c08:    mov     esi, dword ptr [rip + 0x19736]
0x10c0e:    jmp     0x10a77
0x10a77:
0x10c13:    nop     dword ptr [rax + rax]
0x10c18:    cmp     byte ptr [rdx + 1], 0
0x10c1c:    jne     0x107fa
0x10c22:    jmp     0x107c6
0x107c6:
0x10c27:    nop     word ptr [rax + rax]
0x10c30:    mov     ecx, 0x10
0x10c35:    mov     esi, ebx
0x10c37:    mov     rdi, r13
0x10c3a:    xor     eax, eax
0x10c3c:    call    0x10420
0x10c41:    jmp     0x108a0
0x108a0:
0x10c46:    nop     word ptr cs:[rax + rax]
0x10c50:    mov     eax, dword ptr [rip + 0x196de]
0x10c56:    mov     r15d, dword ptr [rip + 0x196db]
0x10c5d:    mov     dword ptr [rsp + 0xc], eax
0x10c61:    jmp     0x10b48
0x10b48:
0x10c66:    nop     word ptr cs:[rax + rax]
0x10c70:    cmp     byte ptr [r13 + 0x6c], 1
0x10c75:    jne     0x10b52
0x10c7b:    mov     eax, dword ptr [rip + 0x196cb]
0x10c81:    mov     dword ptr [rsp + 8], eax
0x10c85:    jmp     0x10b5c
0x10b5c:
0x10c8a:    nop     word ptr [rax + rax]

```

```

0x10c90:    mov     r14, qword ptr [rip + 0x1adf9]
0x10c97:    mov     byte ptr [rdx + 0x4b], 1
0x10c9b:    test    r14, r14
0x10c9e:    je      0x10d49
0x10ca4:    mov     qword ptr [rsp + 0x18], 0
0x10cad:    mov     qword ptr [rsp + 0x10], 0
0x10cb6:    nop     word ptr cs:[rax + rax]
0x10cc0:    mov     rsi, qword ptr [rax + r12]
0x10cc4:    mov     rdi, qword ptr [r14]
0x10cc7:    mov     edx, 0x15
0x10ccc:    call    0x4a40
0x10cd1:    test    eax, eax
0x10cd3:    jne     0x10cee
0x10cd5:    mov     rdi, qword ptr [r14]
0x10cd8:    call    0x49b0
0x10cdd:    cmp     qword ptr [rsp + 0x10], rax
0x10ce2:    jae     0x10cee
0x10ce4:    mov     qword ptr [rsp + 0x18], r14
0x10ce9:    mov     qword ptr [rsp + 0x10], rax
0x10cee:    mov     r14, qword ptr [r14 + 0x18]
0x10cf2:    mov     rax, qword ptr [r13 + 0x38]
0x10cf6:    test    r14, r14
0x10cf9:    jne     0x10cc0
0x10cfb:    mov     rsi, qword ptr [rsp + 0x18]
0x10d00:    lea     rdx, [rax + r12]
0x10d04:    test    rsi, rsi
0x10d07:    je      0x10d49
0x10d09:    mov     eax, dword ptr [rsi + 0x10]
0x10d0c:    xor     ecx, ecx
0x10d0e:    cmp     eax, -1
0x10d11:    cmovne  eax, ecx
0x10d14:    mov     byte ptr [rdx + 0x4e], al
0x10d17:    movsx   edi, al
0x10d1a:    mov     eax, dword ptr [rsi + 8]
0x10d1d:    mov     ecx, eax
0x10d1f:    cmp     eax, -1
0x10d22:    movzx   eax, byte ptr [rip + 0x19613]
0x10d29:    cmovne  eax, ecx
0x10d2c:    mov     byte ptr [rdx + 0x4c], al
0x10d2f:    mov     eax, dword ptr [rsi + 0xc]
0x10d32:    mov     ecx, eax
0x10d34:    cmp     eax, -1
0x10d37:    movzx   eax, byte ptr [rip + 0x195e6]
0x10d3e:    cmovne  eax, ecx
0x10d41:    mov     byte ptr [rdx + 0x4d], al
0x10d44:    jmp     0x10b92
0x10b92:
0x10d49:    mov     byte ptr [rdx + 0x4e], 2

```

```

0x10d4d:      jmp      0x10bb6
0x10bb6
0x10d52:      mov     edx, dword ptr [rsp + 8]
0x10d56:      mov     edi, r15d
0x10d59:      call    0xa570
0x10d5e:      jmp     0x10bb6
0x10bb6
0x10d63:      nop     word ptr cs:[rax + rax]
0x10d6e:      nop
0x10d70:      push    r13
0x10d72:      push    r12
0x10d74:      push    rbp
0x10d75:      push    rbx
0x10d76:      mov     rbx, rdi
0x10d79:      sub     rsp, 0x18
0x10d7d:      mov     rax, qword ptr fs:[0x28]
0x10d86:      mov     qword ptr [rsp + 8], rax
0x10d8b:      xor     eax, eax
0x10d8d:      lea     r12, [rsp + 7]
0x10d92:      mov     rdi, r12
0x10d95:      call    0xa6a0
0x10d9a:      mov     ebp, dword ptr [rbx + 0x2c]
0x10d9d:      mov     eax, ebp
0x10d9f:      cmp     ebp, dword ptr [rbx + 0x68]
0x10da2:      jl      0x10dc0
0x10da4:      jmp     0x10dd9
0x10dd9
0x10da6:      nop     word ptr cs:[rax + rax]
0x10db0:      cmp     byte ptr [rbx + 0x6d], 0
0x10db4:      je      0x10dc0
0x10db6:      mov     esi, ebp
0x10db8:      mov     rdi, rbx
0x10dbb:      call    0x10760
0x10dc0:      add     ebp, 1
0x10dc3:      cmp     ebp, dword ptr [rbx + 0x68]
0x10dc6:      jge     0x10dd9
0x10dc8:      mov     eax, dword ptr [rbx + 0x2c]
0x10dcb:      mov     edx, ebp
0x10dcd:      sub     edx, eax
0x10dcf:      mov     eax, dword ptr [rbx + 0x60]
0x10dd2:      sub     eax, 2
0x10dd5:      cmp     edx, eax
0x10dd7:      jl      0x10db0
0x10dd9:      mov     edx, dword ptr [rip + 0x19545]
0x10ddf:      xor     edi, edi
0x10de1:      mov     esi, 7
0x10de6:      call    0xa570
0x10deb:      movsxd  rdx, dword ptr [rbx + 0x64]

```

```

0x10def:      mov     rdi, qword ptr [rbx + 0x20]
0x10df3:      mov     esi, 0x20
0x10df8:      call    0x4a50
0x10dfd:      mov     eax, dword ptr [rbx + 0x60]
0x10e00:      lea     r13d, [rax - 2]
0x10e04:      mov     eax, dword ptr [rbx + 0x14]
0x10e07:      cmp     r13d, eax
0x10e0a:      cmovg   r13d, eax
0x10e0e:      cmp     r13d, ebp
0x10e11:      jle     0x10e49
0x10e13:      nop     dword ptr [rax + rax]
0x10e18:      mov     rdi, qword ptr [rbx + 8]
0x10e1c:      mov     esi, ebp
0x10e1e:      sub     esi, dword ptr [rbx + 0x2c]
0x10e21:      mov     edx, 1
0x10e26:      add     esi, 1
0x10e29:      add     ebp, 1
0x10e2c:      call    0xc680
0x10e31:      mov     eax, dword ptr [rbx + 0x64]
0x10e34:      mov     rsi, qword ptr [rbx + 0x20]
0x10e38:      mov     rdi, qword ptr [rbx + 8]
0x10e3c:      lea     edx, [rax - 2]
0x10e3f:      call    0xc600
0x10e44:      cmp     r13d, ebp
0x10e47:      jne     0x10e18
0x10e49:      mov     eax, dword ptr [rbx + 0x68]
0x10e4c:      mov     rdi, r12
0x10e4f:      mov     dword ptr [rbx + 0x14], eax
0x10e52:      call    0xa6b0
0x10e57:      mov     rax, qword ptr [rsp + 8]
0x10e5c:      sub     rax, qword ptr fs:[0x28]
0x10e65:      jne     0x10e72
0x10e67:      add     rsp, 0x18
0x10e6b:      pop     rbx
0x10e6c:      pop     rbp
0x10e6d:      pop     r12
0x10e6f:      pop     r13
0x10e71:      ret
0x10e72:      call    0x49d0
0x10e77:      nop     word ptr [rax + rax]
0x10e80:      endbr64
0x10e84:      cmp     byte ptr [rdi + 0x6d], 0
0x10e88:      jne     0x10e90
0x10e8a:      ret
0x10e8b:      nop     dword ptr [rax + rax]
0x10e90:      xor     eax, eax
0x10e92:      jmp     0x10d70
0x10d70:

```

```

0x10e97:      nop      word ptr [rax + rax]
0x10ea0:      endbr64
0x10ea4:      cmp      byte ptr [rdi + 0x6d], 0
0x10ea8:      jne      0x10eb0
0x10eaa:      ret
0x10eab:      nop      dword ptr [rax + rax]
0x10eb0:      jmp      0x10760
0x10760
0x10eb5:      nop      word ptr cs:[rax + rax]
0x10ec0:      endbr64
0x10ec4:      push     r13
0x10ec6:      push     r12
0x10ec8:      push     rbp
0x10ec9:      push     rbx
0x10eca:      mov      rbx, rdi
0x10ecd:      sub      rsp, 0x18
0x10ed1:      movsxd   rdi, dword ptr [rdi + 0x64]
0x10ed5:      mov      rax, qword ptr fs:[0x28]
0x10ede:      mov      qword ptr [rsp + 8], rax
0x10ee3:      xor      eax, eax
0x10ee5:      call     0x1d670
0x10eea:      cmp      byte ptr [rbx + 0x6d], 0
0x10eee:      jne      0x10f10
0x10ef0:      mov      rax, qword ptr [rsp + 8]
0x10ef5:      sub      rax, qword ptr fs:[0x28]
0x10efe:      jne      0x11006
0x10f04:      add      rsp, 0x18
0x10f08:      pop      rbx
0x10f09:      pop      rbp
0x10f0a:      pop      r12
0x10f0c:      pop      r13
0x10f0e:      ret
0x10f0f:      nop
0x10f10:      lea      r13, [rsp + 7]
0x10f15:      mov      r12, rax
0x10f18:      mov      ebp, 1
0x10f1d:      mov      rdi, r13
0x10f20:      call     0xa6a0
0x10f25:      mov      esi, dword ptr [rip + 0x193f5]
0x10f2b:      xor      edi, edi
0x10f2d:      mov      edx, esi
0x10f2f:      call     0xa570
0x10f34:      cmp      dword ptr [rbx + 0x60], 2
0x10f38:      jle      0x10fa4
0x10f3a:      nop      word ptr [rax + rax]
0x10f40:      mov      rdi, qword ptr [rbx + 8]
0x10f44:      xor      edx, edx
0x10f46:      mov      esi, ebp

```

```

0x10f48:      add     ebp, 1
0x10f4b:      call    0xc680
0x10f50:      mov     rdi, qword ptr [rbx + 8]
0x10f54:      mov     esi, 0x20
0x10f59:      call    0xc650
0x10f5e:      mov     eax, dword ptr [rbx + 0x60]
0x10f61:      lea     edx, [rax - 1]
0x10f64:      cmp     edx, ebp
0x10f66:      jg      0x10f40
0x10f68:      cmp     eax, 2
0x10f6b:      jle     0x10fa4
0x10f6d:      mov     ebp, 1
0x10f72:      nop     word ptr [rax + rax]
0x10f78:      mov     eax, dword ptr [rbx + 0x64]
0x10f7b:      mov     rdi, qword ptr [rbx + 8]
0x10f7f:      mov     esi, ebp
0x10f81:      add     ebp, 1
0x10f84:      lea     edx, [rax - 1]
0x10f87:      call    0xc680
0x10f8c:      mov     rdi, qword ptr [rbx + 8]
0x10f90:      mov     esi, 0x20
0x10f95:      call    0xc650
0x10f9a:      mov     eax, dword ptr [rbx + 0x60]
0x10f9d:      sub     eax, 1
0x10fa0:      cmp     eax, ebp
0x10fa2:      jg      0x10f78
0x10fa4:      movsxd  rdx, dword ptr [rbx + 0x64]
0x10fa8:      mov     rdi, r12
0x10fab:      mov     esi, 0x20
0x10fb0:      call    0x4a50
0x10fb5:      mov     rdi, qword ptr [rbx + 8]
0x10fb9:      xor     edx, edx
0x10fbb:      xor     esi, esi
0x10fbd:      call    0xc680
0x10fc2:      mov     edx, dword ptr [rbx + 0x64]
0x10fc5:      mov     rdi, qword ptr [rbx + 8]
0x10fc9:      mov     rsi, r12
0x10fcc:      call    0xc600
0x10fd1:      mov     eax, dword ptr [rbx + 0x60]
0x10fd4:      mov     rdi, qword ptr [rbx + 8]
0x10fd8:      xor     edx, edx
0x10fda:      lea     esi, [rax - 1]
0x10fdd:      call    0xc680
0x10fe2:      mov     edx, dword ptr [rbx + 0x64]
0x10fe5:      mov     rdi, qword ptr [rbx + 8]
0x10fe9:      mov     rsi, r12
0x10fec:      call    0xc600
0x10ff1:      mov     rdi, r12

```

```

0x10ff4:      call    0x18d10
0x10ff9:      mov     rdi, r13
0x10ffc:      call    0xa6b0
0x11001:      jmp     0x10ef0
0x10ef0
0x11006:      call    0x49d0
0x1100b:      nop     dword ptr [rax + rax]
0x11010:      endbr64
0x11014:      push    rbp
0x11015:      xor     eax, eax
0x11017:      mov     rbp, rdi
0x1101a:      call    0x10ec0
0x1101f:      xor     eax, eax
0x11021:      mov     rdi, rbp
0x11024:      call    0x10160
0x11029:      cmp     byte ptr [rbp + 0x6d], 0
0x1102d:      je      0x11039
0x1102f:      mov     rdi, rbp
0x11032:      xor     eax, eax
0x11034:      call    0xec50
0x11039:      xor     eax, eax
0x1103b:      mov     rdi, rbp
0x1103e:      call    0x10240
0x11043:      cmp     byte ptr [rbp + 0x6d], 0
0x11047:      jne     0x11050
0x11049:      pop     rbp
0x1104a:      ret
0x1104b:      nop     dword ptr [rax + rax]
0x11050:      mov     rdi, rbp
0x11053:      xor     eax, eax
0x11055:      pop     rbp
0x11056:      jmp     0x10d70
0x10d70
0x1105b:      nop     dword ptr [rax + rax]
0x11060:      endbr64
0x11064:      mov     edx, dword ptr [rdi + 0x68]
0x11067:      test    edx, edx
0x11069:      jle     0x110ab
0x1106b:      sub     edx, 1
0x1106e:      mov     rcx, qword ptr [rdi + 0x38]
0x11072:      lea     rdx, [rdx + rdx*4]
0x11076:      shl     rdx, 4
0x1107a:      lea     rax, [rcx + 0x49]
0x1107e:      lea     rdx, [rcx + rdx + 0x99]
0x11086:      nop     word ptr cs:[rax + rax]
0x11090:      cmp     byte ptr [rax], 0
0x11093:      jne     0x110a2
0x11095:      cmp     byte ptr [rax - 1], 0

```



```

0x11099:      je      0x110a2
0x1109b:      mov     byte ptr [rax], 1
0x1109e:      add     dword ptr [rdi + 0x70], 1
0x110a2:      add     rax, 0x50
0x110a6:      cmp     rax, rdx
0x110a9:      jne     0x11090
0x110ab:      ret
0x110ac:      nop     dword ptr [rax]
0x110b0:      endbr64
0x110b4:      mov     edx, dword ptr [rdi + 0x68]
0x110b7:      test    edx, edx
0x110b9:      jle     0x110ec
0x110bb:      sub     edx, 1
0x110be:      mov     rcx, qword ptr [rdi + 0x38]
0x110c2:      lea     rdx, [rdx + rdx*4]
0x110c6:      shl     rdx, 4
0x110ca:      lea     rax, [rcx + 0x49]
0x110ce:      lea     rdx, [rcx + rdx + 0x99]
0x110d6:      nop     word ptr cs:[rax + rax]
0x110e0:      mov     byte ptr [rax], 0
0x110e3:      add     rax, 0x50
0x110e7:      cmp     rax, rdx
0x110ea:      jne     0x110e0
0x110ec:      mov     dword ptr [rdi + 0x70], 0
0x110f3:      ret
0x110f4:      nop     word ptr cs:[rax + rax]
0x110ff:      nop
0x11100:      endbr64
0x11104:      mov     rax, qword ptr [rdi + 0x18]
0x11108:      ret
0x11109:      nop     dword ptr [rax]
0x11110:      endbr64
0x11114:      mov     eax, dword ptr [rip + 0x1a996]
0x1111a:      test    eax, eax
0x1111c:      jne     0x11120
0x1111e:      ret
0x1111f:      nop
0x11120:      push    rbp
0x11121:      xor     eax, eax
0x11123:      push    rbx
0x11124:      sub     rsp, 8
0x11128:      mov     dword ptr [rip + 0x1a97e], 0
0x11132:      call    0xc8c0
0x11137:      mov     edx, 4
0x1113c:      lea     rsi, [rip + 0xf3ef]
0x11143:      lea     rdi, [rip + 0x100af]
0x1114a:      mov     rbp, rax
0x1114d:      call    0x8070

```

```

0x11152:    mov     rdi, rbp
0x11155:    mov     ebx, eax
0x11157:    call    0xc970
0x1115c:    xor     eax, eax
0x1115e:    and     ebx, 0xffffffff
0x11161:    call    0xd870
0x11166:    xor     eax, eax
0x11168:    call    0xd740
0x1116d:    xor     eax, eax
0x1116f:    call    0xa820
0x11174:    xor     eax, eax
0x11176:    cmp     bl, 0x4e
0x11179:    setne   al
0x1117c:    add     rsp, 8
0x11180:    pop     rbx
0x11181:    pop     rbp
0x11182:    ret
0x11183:    nop     word ptr cs:[rax + rax]
0x1118e:    nop
0x11190:    endbr64
0x11194:    push    rbp
0x11195:    mov     rbp, rsi
0x11198:    sub     rsp, 0x130
0x1119f:    mov     rax, qword ptr fs:[0x28]
0x111a8:    mov     qword ptr [rsp + 0x128], rax
0x111b0:    xor     eax, eax
0x111b2:    mov     rsi, rsp
0x111b5:    call    0x18f40
0x111ba:    test    eax, eax
0x111bc:    je      0x111e0
0x111be:    xor     eax, eax
0x111c0:    mov     rdx, qword ptr [rsp + 0x128]
0x111c8:    sub     rdx, qword ptr fs:[0x28]
0x111d1:    jne     0x11217
0x111d3:    add     rsp, 0x130
0x111da:    pop     rbp
0x111db:    ret
0x111dc:    nop     dword ptr [rax]
0x111e0:    lea     rsi, [rsp + 0x90]
0x111e8:    mov     rdi, rbp
0x111eb:    call    0x18f40
0x111f0:    test    eax, eax
0x111f2:    jne     0x111be
0x111f4:    mov     rax, qword ptr [rsp + 0x90]
0x111fc:    cmp     qword ptr [rsp], rax
0x11200:    jne     0x111be
0x11202:    mov     rax, qword ptr [rsp + 0x98]
0x1120a:    cmp     qword ptr [rsp + 8], rax

```

```

0x1120f:      sete    al
0x11212:      movzx   eax, al
0x11215:      jmp     0x111c0
0x111c0:
0x11217:      call    0x49d0
0x1121c:      nop     dword ptr [rax]
0x11220:      endbr64
0x11224:      push    rbx
0x11225:      mov     edx, dword ptr [rdi + 0x70]
0x11228:      mov     rbx, rdi
0x1122b:      mov     ecx, 0x1e
0x11230:      test    edx, edx
0x11232:      je      0x11268
0x11234:      lea     rdx, [rip + 0xffd8]
0x1123b:      lea     rdi, [rip + 0x107de]
0x11242:      xor     eax, eax
0x11244:      call    0xf760
0x11249:      xor     r8d, r8d
0x1124c:      cmp     al, 0x6f
0x1124e:      je      0x1125e
0x11250:      cmp     al, 0x73
0x11252:      je      0x11280
0x11254:      mov     r8d, 2
0x1125a:      cmp     al, 0x61
0x1125c:      je      0x11298
0x1125e:      mov     eax, r8d
0x11261:      pop     rbx
0x11262:      ret
0x11263:      nop     dword ptr [rax + rax]
0x11268:      lea     rdx, [rip + 0xffa9]
0x1126f:      lea     rdi, [rip + 0x107da]
0x11276:      xor     eax, eax
0x11278:      call    0xf760
0x1127d:      jmp     0x11249
0x11249:
0x1127f:      nop
0x11280:      cmp     dword ptr [rbx + 0x70], 1
0x11284:      mov     r8d, 2
0x1128a:      pop     rbx
0x1128b:      sbb     r8d, -1
0x1128f:      mov     eax, r8d
0x11292:      ret
0x11293:      nop     dword ptr [rax + rax]
0x11298:      mov     eax, dword ptr [rbx + 0x70]
0x1129b:      test    eax, eax
0x1129d:      je      0x1125e
0x1129f:      mov     dword ptr [rbx + 0x90], 0
0x112a9:      xor     r8d, r8d

```

```

0x112ac:    pop     rbx
0x112ad:    mov     eax, r8d
0x112b0:    ret
0x112b1:    nop     word ptr cs:[rax + rax]
0x112bc:    nop     dword ptr [rax]
0x112c0:    endbr64
0x112c4:    push    rbp
0x112c5:    mov     rbp, rdi
0x112c8:    sub     rsp, 0xa0
0x112cf:    mov     rax, qword ptr fs:[0x28]
0x112d8:    mov     qword ptr [rsp + 0x98], rax
0x112e0:    xor     eax, eax
0x112e2:    mov     rsi, rsp
0x112e5:    call    0x18f40
0x112ea:    test    eax, eax
0x112ec:    jne     0x112fe
0x112ee:    mov     eax, dword ptr [rsp + 0x18]
0x112f2:    and     eax, 0xf000
0x112f7:    cmp     eax, 0x8000
0x112fc:    je      0x11320
0x112fe:    mov     rax, qword ptr [rsp + 0x98]
0x11306:    sub     rax, qword ptr fs:[0x28]
0x1130f:    jne     0x1132a
0x11311:    add     rsp, 0xa0
0x11318:    pop     rbp
0x11319:    ret
0x1131a:    nop     word ptr [rax + rax]
0x11320:    mov     rdi, rbp
0x11323:    call    0x4870
0x11328:    jmp     0x112fe
0x112fe:
0x1132a:    call    0x49d0
0x1132f:    nop
0x11330:    endbr64
0x11334:    push    r15
0x11336:    push    r14
0x11338:    mov     r14, rdx
0x1133b:    push    r13
0x1133d:    push    r12
0x1133f:    mov     r12d, ecx
0x11342:    push    rbp
0x11343:    and     r12d, 0xf000
0x1134a:    mov     rbp, rdi
0x1134d:    push    rbx
0x1134e:    mov     ebx, ecx
0x11350:    sub     rsp, 0x178
0x11357:    mov     qword ptr [rsp + 8], rsi
0x1135c:    mov     dword ptr [rsp + 0x30], r8d

```

```

0x11361:    mov     dword ptr [rsp + 0x34], r9d
0x11366:    mov     rax, qword ptr fs:[0x28]
0x1136f:    mov     qword ptr [rsp + 0x168], rax
0x11377:    xor     eax, eax
0x11379:    cmp     r12d, 0xa000
0x11380:    je      0x11a10
0x11386:    cmp     r12d, 0x4000
0x1138d:    je      0x11828
0x11393:    mov     rdi, r14
0x11396:    call    0x49b0
0x1139b:    mov     rdi, r14
0x1139e:    mov     r13, rax
0x113a1:    call    0x1d7a0
0x113a6:    lea     rsi, [rsp + 0x40]
0x113ab:    mov     rdi, rax
0x113ae:    mov     qword ptr [rsp + 0x18], rax
0x113b3:    call    0x18f40
0x113b8:    test    eax, eax
0x113ba:    je      0x11630
0x113c0:    mov     rdi, qword ptr [rsp + 0x18]
0x113c5:    call    0x190c0
0x113ca:    cmp     byte ptr [rax], 0
0x113cd:    mov     r13, rax
0x113d0:    je      0x11ae5
0x113d6:    mov     esi, dword ptr [rbp + 0x90]
0x113dc:    test    esi, esi
0x113de:    je      0x113f4
0x113e0:    mov     rdi, qword ptr [rsp + 0x18]
0x113e5:    xor     esi, esi
0x113e7:    call    0x4d90
0x113ec:    test    eax, eax
0x113ee:    je      0x11a60
0x113f4:    mov     rdi, qword ptr [rsp + 8]
0x113f9:    call    0x49b0
0x113fe:    lea     rdi, [rax + 0x20]
0x11402:    call    0x1d670
0x11407:    mov     r8, qword ptr [rsp + 8]
0x1140c:    lea     rcx, [rip + 0xfe34]
0x11413:    mov     r13, rax
0x11416:    cmp     r12d, 0x8000
0x1141d:    je      0x11a48
0x11423:    mov     rdx, -1
0x1142a:    mov     esi, 1
0x1142f:    mov     rdi, r13
0x11432:    xor     eax, eax
0x11434:    call    0x4f10
0x11439:    mov     edx, 1
0x1143e:    mov     esi, 1

```

```

0x11443:    mov     rdi, r13
0x11446:    call   0xe480
0x1144b:    xor     eax, eax
0x1144d:    call   0xa820
0x11452:    mov     rdi, r13
0x11455:    call   0x18d10
0x1145a:    mov     rdi, qword ptr [rsp + 8]
0x1145f:    xor     esi, esi
0x11461:    xor     eax, eax
0x11463:    call   0x4d30
0x11468:    mov     r15d, eax
0x1146b:    cmp     eax, -1
0x1146e:    je      0x11a90
0x11474:    mov     ecx, dword ptr [rbp + 0x10]
0x11477:    test    ecx, ecx
0x11479:    je      0x1147e
0x1147b:    and     ebx, 0xffffffffb6
0x1147e:    mov     rdi, qword ptr [rsp + 0x18]
0x11483:    mov     esi, ebx
0x11485:    call   0x4e00
0x1148a:    mov     edi, r15d
0x1148d:    mov     dword ptr [rsp + 0x28], eax
0x11491:    cmp     eax, -1
0x11494:    je      0x11ae0
0x1149a:    call   0x182f0
0x1149f:    mov     qword ptr [rsp + 0x10], 0x8000
0x114a8:    mov     r13, rax
0x114ab:    cmp     rax, 0x8000
0x114b1:    jle     0x11b38
0x114b7:    mov     eax, ebx
0x114b9:    and     ebx, 0xb000
0x114bf:    and     eax, 0xf000
0x114c4:    mov     dword ptr [rsp + 0x2c], eax
0x114c8:    cmp     ebx, 0x2000
0x114ce:    je      0x11b00
0x114d4:    test    r13, r13
0x114d7:    je      0x11c58
0x114dd:    mov     rdi, qword ptr [rsp + 0x10]
0x114e2:    call   0x1d670
0x114e7:    mov     r14, rax
0x114ea:    test    r13, r13
0x114ed:    jle     0x11aa0
0x114f3:    movabs  rdx, 0xa3d70a3d70a3d70b
0x114fd:    mov     rax, r13
0x11500:    imul    rdx
0x11503:    mov     rax, r13
0x11506:    sar     rax, 0x3f
0x1150a:    add     rdx, r13

```

```

0x1150d:      sar      rdx, 6
0x11511:      sub      rdx, rax
0x11514:      mov      qword ptr [rsp + 0x38], rdx
0x11519:      xor      r12d, r12d
0x1151c:      jmp      0x11530
0x11530:
0x1151e:      nop
0x11520:      add      r12, 0x8000
0x11527:      cmp      r12, r13
0x1152a:      jge      0x11aa0
0x11530:      mov      rax, qword ptr [rsp + 0x10]
0x11535:      mov      rbp, r13
0x11538:      mov      edx, dword ptr [rip + 0x1a572]
0x1153e:      sub      rbp, r12
0x11541:      cmp      rbp, rax
0x11544:      cmovg     rbp, rax
0x11548:      test     edx, edx
0x1154a:      jne      0x11690
0x11550:      mov      edi, 1
0x11555:      call     0x18b70
0x1155a:      mov      edi, r15d
0x1155d:      movsxd    rdx, ebp
0x11560:      mov      rsi, r14
0x11563:      call     0x18d30
0x11568:      xor      edi, edi
0x1156a:      mov      ebx, eax
0x1156c:      call     0x18b70
0x11571:      cmp      ebp, ebx
0x11573:      je       0x11583
0x11575:      test     ebx, ebx
0x11577:      js       0x11bf0
0x1157d:      je       0x11aa0
0x11583:      mov      eax, dword ptr [rip + 0x1a527]
0x11589:      test     eax, eax
0x1158b:      jne      0x11778
0x11591:      movsxd    rbp, ebx
0x11594:      mov      edi, dword ptr [rsp + 0x28]
0x11598:      mov      rsi, r14
0x1159b:      mov      rdx, rbp
0x1159e:      call     0x18da0
0x115a3:      cmp      ebx, eax
0x115a5:      jne      0x11b50
0x115ab:      add      rbp, r12
0x115ae:      cmp      rbp, r13
0x115b1:      jg       0x11520
0x115b7:      mov      rdi, qword ptr [rsp + 8]
0x115bc:      call     0x49b0
0x115c1:      lea      rdi, [rax + 0x20]

```

```

0x115c5:      call    0x1d670
0x115ca:      cmp     dword ptr [rsp + 0x2c], 0x8000
0x115d2:      mov     rbx, rax
0x115d5:      je      0x117e0
0x115db:      mov     r9, qword ptr [rsp + 8]
0x115e0:      mov     rdi, rax
0x115e3:      mov     r8, rbp
0x115e6:      mov     esi, 1
0x115eb:      lea     rcx, [rip + 0xfc7b]
0x115f2:      mov     rdx, -1
0x115f9:      xor     eax, eax
0x115fb:      call    0x4f10
0x11600:      mov     rdi, rbx
0x11603:      mov     edx, 1
0x11608:      mov     esi, 1
0x1160d:      call    0xe480
0x11612:      xor     eax, eax
0x11614:      call    0xa820
0x11619:      mov     rdi, rbx
0x1161c:      call    0x18d10
0x11621:      jmp     0x11520
0x11520:
0x11626:      nop     word ptr cs:[rax + rax]
0x11630:      mov     eax, dword ptr [rsp + 0x58]
0x11634:      and     eax, 0xf000
0x11639:      cmp     eax, 0x4000
0x1163e:      jne     0x113c0
0x11644:      mov     r14, qword ptr [rsp + 8]
0x11649:      mov     rdi, r14
0x1164c:      call    0x49b0
0x11651:      mov     rdi, qword ptr [rsp + 0x18]
0x11656:      lea     rsi, [r13 + rax + 2]
0x1165b:      call    0x1d690
0x11660:      mov     rdi, rax
0x11663:      mov     r15, rax
0x11666:      mov     qword ptr [rsp + 0x18], rax
0x1166b:      call    0x49b0
0x11670:      mov     rsi, r14
0x11673:      mov     byte ptr [r15 + rax], 0x2f
0x11678:      lea     rdi, [r15 + rax + 1]
0x1167d:      call    0x48b0
0x11682:      jmp     0x113c0
0x113c0:
0x11687:      nop     word ptr [rax + rax]
0x11690:      mov     dword ptr [rip + 0x1a416], 0
0x1169a:      xor     eax, eax
0x1169c:      call    0xc8c0
0x116a1:      mov     edx, 4

```



```

0x116a6:    lea     rsi, [rip + 0xee85]
0x116ad:    lea     rdi, [rip + 0xfb45]
0x116b4:    mov     qword ptr [rsp + 0x20], rax
0x116b9:    call    0x8070
0x116be:    mov     rdi, qword ptr [rsp + 0x20]
0x116c3:    mov     ebx, eax
0x116c5:    call    0xc970
0x116ca:    xor     eax, eax
0x116cc:    and     ebx, 0xffffffff
0x116cf:    call    0xd870
0x116d4:    xor     eax, eax
0x116d6:    call    0xd740
0x116db:    xor     eax, eax
0x116dd:    call    0xa820
0x116e2:    cmp     bl, 0x4e
0x116e5:    je      0x11550
0x116eb:    mov     edi, r15d
0x116ee:    call    0x4ab0
0x116f3:    mov     edi, dword ptr [rsp + 0x28]
0x116f7:    call    0x4ab0
0x116fc:    mov     rdi, qword ptr [rsp + 0x18]
0x11701:    lea     rsi, [rsp + 0xd0]
0x11709:    call    0x18f40
0x1170e:    test    eax, eax
0x11710:    jne     0x11729
0x11712:    mov     eax, dword ptr [rsp + 0xe8]
0x11719:    and     eax, 0xf000
0x1171e:    cmp     eax, 0x8000
0x11723:    je      0x11ced
0x11729:    mov     rdi, r14
0x1172c:    mov     r13d, 2
0x11732:    call    0x18d10
0x11737:    mov     rdi, qword ptr [rsp + 0x18]
0x1173c:    call    0x18d10
0x11741:    nop     dword ptr [rax]
0x11748:    mov     rax, qword ptr [rsp + 0x168]
0x11750:    sub     rax, qword ptr fs:[0x28]
0x11759:    jne     0x11cfc
0x1175f:    add     rsp, 0x178
0x11766:    mov     eax, r13d
0x11769:    pop     rbx
0x1176a:    pop     rbp
0x1176b:    pop     r12
0x1176d:    pop     r13
0x1176f:    pop     r14
0x11771:    pop     r15
0x11773:    ret
0x11774:    nop     dword ptr [rax]

```

```

0x11778:    mov     dword ptr [rip + 0x1a32e], 0
0x11782:    xor     eax, eax
0x11784:    call   0xc8c0
0x11789:    mov     edx, 4
0x1178e:    lea     rsi, [rip + 0xed9d]
0x11795:    lea     rdi, [rip + 0xfa5d]
0x1179c:    mov     qword ptr [rsp + 0x20], rax
0x117a1:    call   0x8070
0x117a6:    mov     rdi, qword ptr [rsp + 0x20]
0x117ab:    mov     ebp, eax
0x117ad:    call   0xc970
0x117b2:    xor     eax, eax
0x117b4:    and     ebp, 0xffffffff
0x117b7:    call   0xd870
0x117bc:    xor     eax, eax
0x117be:    call   0xd740
0x117c3:    xor     eax, eax
0x117c5:    call   0xa820
0x117ca:    cmp     bpl, 0x4e
0x117ce:    je      0x11591
0x117d4:    jmp     0x116eb
0x116eb:
0x117d9:    nop     dword ptr [rax]
0x117e0:    cmp     r13, 0x1869f
0x117e7:    jle     0x119f0
0x117ed:    mov     rax, rbp
0x117f0:    cqo
0x117f2:    idiv    qword ptr [rsp + 0x38]
0x117f7:    mov     r8, rax
0x117fa:    mov     r9, qword ptr [rsp + 8]
0x117ff:    mov     esi, 1
0x11804:    mov     rdi, rbx
0x11807:    xor     eax, eax
0x11809:    lea     rcx, [rip + 0xfa4b]
0x11810:    mov     rdx, -1
0x11817:    call   0x4f10
0x1181c:    jmp     0x11600
0x11600:
0x11821:    nop     dword ptr [rax]
0x11828:    lea     rsi, [rsp + 0x40]
0x1182d:    mov     rdi, r14
0x11830:    call   0x18f40
0x11835:    test    eax, eax
0x11837:    jne     0x118c3
0x1183d:    mov     eax, dword ptr [rsp + 0x58]
0x11841:    mov     r13d, 0xb
0x11847:    and     eax, 0xf000
0x1184c:    cmp     eax, 0x4000

```

```

0x11851:    jne     0x11748
0x11857:    mov     rdi, r14
0x1185a:    call    0x49b0
0x1185f:    mov     r15, qword ptr [rsp + 8]
0x11864:    mov     rbx, rax
0x11867:    mov     rdi, r15
0x1186a:    call    0x49b0
0x1186f:    lea     rdi, [rbx + rax + 2]
0x11874:    call    0x1d670
0x11879:    mov     r9, r15
0x1187c:    mov     r8, r14
0x1187f:    mov     esi, 1
0x11884:    mov     rdi, rax
0x11887:    mov     r12, rax
0x1188a:    xor     eax, eax
0x1188c:    mov     rdx, -1
0x11893:    lea     rcx, [rip + 0xee67]
0x1189a:    call    0x4f10
0x1189f:    mov     edi, dword ptr [rbp + 0x90]
0x118a5:    test    edi, edi
0x118a7:    je      0x118bb
0x118a9:    xor     esi, esi
0x118ab:    mov     rdi, r12
0x118ae:    call    0x4d90
0x118b3:    test    eax, eax
0x118b5:    je      0x11cbb
0x118bb:    mov     rdi, r12
0x118be:    call    0x18d10
0x118c3:    mov     r15, qword ptr [rsp + 8]
0x118c8:    mov     r13d, 6
0x118ce:    mov     rdi, r15
0x118d1:    call    0x49b0
0x118d6:    mov     rdi, r14
0x118d9:    mov     rbx, rax
0x118dc:    call    0x49b0
0x118e1:    lea     rdi, [rbx + rax + 0x21]
0x118e6:    call    0x1d670
0x118eb:    mov     rdi, r14
0x118ee:    mov     r12, rax
0x118f1:    call    0x190c0
0x118f6:    cmp     byte ptr [rax], 0
0x118f9:    je      0x11748
0x118ff:    mov     rdi, r15
0x11902:    call    0x49b0
0x11907:    mov     rdi, r14
0x1190a:    mov     rbx, rax
0x1190d:    call    0x49b0
0x11912:    lea     rdi, [rbx + rax + 0x20]

```

```

0x11917:      call    0x1d670
0x1191c:      mov     r9, r14
0x1191f:      mov     r8, r15
0x11922:      mov     esi, 1
0x11927:      lea     rcx, [rip + 0xf8ed]
0x1192e:      mov     r13, rax
0x11931:      mov     rdi, rax
0x11934:      xor     eax, eax
0x11936:      mov     rdx, -1
0x1193d:      call    0x4f10
0x11942:      mov     edx, 1
0x11947:      mov     esi, 1
0x1194c:      mov     rdi, r13
0x1194f:      call    0xe480
0x11954:      xor     eax, eax
0x11956:      call    0xa820
0x1195b:      mov     rdi, r13
0x1195e:      mov     r13d, 0xc
0x11964:      call    0x18d10
0x11969:      mov     r9, r14
0x1196c:      mov     r8, r15
0x1196f:      mov     esi, 1
0x11974:      lea     rcx, [rip + 0xf8a7]
0x1197b:      mov     rdx, -1
0x11982:      mov     rdi, r12
0x11985:      xor     eax, eax
0x11987:      call    0x4f10
0x1198c:      mov     esi, 1
0x11991:      mov     rdi, r12
0x11994:      call    0x193f0
0x11999:      mov     rdi, r12
0x1199c:      mov     ebx, eax
0x1199e:      call    0x18d10
0x119a3:      mov     rdi, qword ptr [rbp + 0x18]
0x119a7:      call    0xc4b0
0x119ac:      mov     eax, ebx
0x119ae:      and     eax, 0x7f
0x119b1:      add     eax, 1
0x119b4:      cmp     al, 1
0x119b6:      jg      0x11748
0x119bc:      mov     r13d, 1
0x119c2:      test    bl, 0x7f
0x119c5:      jne     0x11748
0x119cb:      and     bh, 0xff
0x119ce:      je      0x11748
0x119d4:      lea     rdi, [rip + 0xf857]
0x119db:      mov     r13d, 9
0x119e1:      call    0x195e0

```

```

0x119e6:      jmp      0x11748
0x11748
0x119eb:      nop      dword ptr [rax + rax]
0x119f0:      lea      rax, [rbp + rbp*4]
0x119f5:      lea      r8, [rax + rax*4]
0x119f9:      shl      r8, 2
0x119fd:      mov      rax, r8
0x11a00:      cqo
0x11a02:      idiv     r13
0x11a05:      mov      r8, rax
0x11a08:      jmp      0x117fa
0x117fa
0x11a0d:      nop      dword ptr [rax]
0x11a10:      mov      rdi, qword ptr [rsp + 8]
0x11a15:      lea      rsi, [rsp + 0xd0]
0x11a1d:      call     0x18f40
0x11a22:      cmp      eax, -1
0x11a25:      je       0x11393
0x11a2b:      mov      ebx, dword ptr [rsp + 0xe8]
0x11a32:      mov      r12d, ebx
0x11a35:      and      r12d, 0xf000
0x11a3c:      jmp      0x11386
0x11386
0x11a41:      nop      dword ptr [rax]
0x11a48:      mov      r8, qword ptr [rsp + 8]
0x11a4d:      lea      rcx, [rip + 0xf7e1]
0x11a54:      jmp      0x11423
0x11423
0x11a59:      nop      dword ptr [rax]
0x11a60:      call     0xe4e0
0x11a65:      mov      rsi, r13
0x11a68:      xor      eax, eax
0x11a6a:      mov      rdi, rbp
0x11a6d:      call     0x11220
0x11a72:      mov      r13d, eax
0x11a75:      xor      eax, eax
0x11a77:      call     0xa820
0x11a7c:      test     r13d, r13d
0x11a7f:      je       0x113f4
0x11a85:      jmp      0x11748
0x11748
0x11a8a:      nop      word ptr [rax + rax]
0x11a90:      mov      r13d, 4
0x11a96:      jmp      0x11748
0x11748
0x11a9b:      nop      dword ptr [rax + rax]
0x11aa0:      call     0x49e0
0x11aa5:      test     eax, eax

```

```

0x11aa7:      je      0x11c90
0x11aad:      mov     edi, r15d
0x11ab0:      mov     r13d, 1
0x11ab6:      call    0x4ab0
0x11abb:      mov     edi, dword ptr [rsp + 0x28]
0x11abf:      call    0x4ab0
0x11ac4:      mov     rdi, r14
0x11ac7:      call    0x18d10
0x11acc:      mov     rdi, qword ptr [rsp + 0x18]
0x11ad1:      call    0x18d10
0x11ad6:      jmp      0x11748
0x11748:
0x11adb:      nop      dword ptr [rax + rax]
0x11ae0:      call    0x4ab0
0x11ae5:      mov     rdi, qword ptr [rsp + 0x18]
0x11aea:      mov     r13d, 6
0x11af0:      call    0x18d10
0x11af5:      jmp      0x11748
0x11748:
0x11afa:      nop      word ptr [rax + rax]
0x11b00:      movabs   r13, 0x7fffffffffffffff
0x11b0a:      mov     edi, 0x8000
0x11b0f:      call    0x1d670
0x11b14:      mov     qword ptr [rsp + 0x10], 0x8000
0x11b1d:      mov     r14, rax
0x11b20:      movabs   rax, 0x147ae147ae147ae
0x11b2a:      mov     qword ptr [rsp + 0x38], rax
0x11b2f:      jmp      0x11519
0x11519:
0x11b34:      nop      dword ptr [rax]
0x11b38:      mov     edi, r15d
0x11b3b:      call    0x182f0
0x11b40:      mov     qword ptr [rsp + 0x10], rax
0x11b45:      mov     r13, rax
0x11b48:      jmp      0x114b7
0x114b7:
0x11b4d:      nop      dword ptr [rax]
0x11b50:      mov     dword ptr [rsp + 8], eax
0x11b54:      call    0x4860
0x11b59:      mov     edi, r15d
0x11b5c:      mov     ebx, dword ptr [rax]
0x11b5e:      call    0x4ab0
0x11b63:      mov     edi, dword ptr [rsp + 0x28]
0x11b67:      call    0x4ab0
0x11b6c:      mov     rdi, qword ptr [rsp + 0x18]
0x11b71:      lea     rsi, [rsp + 0xd0]
0x11b79:      call    0x18f40
0x11b7e:      mov     edx, dword ptr [rsp + 8]

```

```

0x11b82:    test    eax, eax
0x11b84:    jne     0x11bb0
0x11b86:    mov     eax, dword ptr [rsp + 0xe8]
0x11b8d:    and     eax, 0xf000
0x11b92:    cmp     eax, 0x8000
0x11b97:    jne     0x11bb0
0x11b99:    mov     rdi, qword ptr [rsp + 0x18]
0x11b9e:    call    0x4870
0x11ba3:    mov     edx, dword ptr [rsp + 8]
0x11ba7:    nop     word ptr [rax + rax]
0x11bb0:    mov     rdi, r14
0x11bb3:    mov     dword ptr [rsp + 8], edx
0x11bb7:    call    0x18d10
0x11bbc:    mov     rdi, qword ptr [rsp + 0x18]
0x11bc1:    call    0x18d10
0x11bc6:    mov     edx, dword ptr [rsp + 8]
0x11bca:    cmp     ebx, 0x1c
0x11bcd:    sete    r13b
0x11bd1:    test    edx, edx
0x11bd3:    setg    al
0x11bd6:    or      r13d, eax
0x11bd9:    movzx   r13d, r13b
0x11bdd:    add     r13d, 7
0x11be1:    jmp     0x11748
0x11748:
0x11be6:    nop     word ptr cs:[rax + rax]
0x11bf0:    mov     edi, r15d
0x11bf3:    call    0x4ab0
0x11bf8:    mov     edi, dword ptr [rsp + 0x28]
0x11bfc:    call    0x4ab0
0x11c01:    mov     rdi, qword ptr [rsp + 0x18]
0x11c06:    lea     rsi, [rsp + 0xd0]
0x11c0e:    call    0x18f40
0x11c13:    test    eax, eax
0x11c15:    jne     0x11c38
0x11c17:    mov     eax, dword ptr [rsp + 0xe8]
0x11c1e:    and     eax, 0xf000
0x11c23:    cmp     eax, 0x8000
0x11c28:    jne     0x11c38
0x11c2a:    mov     rdi, qword ptr [rsp + 0x18]
0x11c2f:    call    0x4870
0x11c34:    nop     dword ptr [rax]
0x11c38:    mov     rdi, r14
0x11c3b:    mov     r13d, 5
0x11c41:    call    0x18d10
0x11c46:    mov     rdi, qword ptr [rsp + 0x18]
0x11c4b:    call    0x18d10
0x11c50:    jmp     0x11748

```

```

0x11748
0x11c55:    nop        dword ptr [rax]
0x11c58:    call       0x49e0
0x11c5d:    test       eax, eax
0x11c5f:    je         0x11ca7
0x11c61:    mov        edi, r15d
0x11c64:    mov        r13d, 1
0x11c6a:    call       0x4ab0
0x11c6f:    mov        edi, dword ptr [rsp + 0x28]
0x11c73:    call       0x4ab0
0x11c78:    mov        rdi, qword ptr [rsp + 0x18]
0x11c7d:    call       0x18d10
0x11c82:    jmp        0x11748
0x11748
0x11c87:    nop        word ptr [rax + rax]
0x11c90:    mov        edx, dword ptr [rsp + 0x34]
0x11c94:    mov        esi, dword ptr [rsp + 0x30]
0x11c98:    mov        rdi, qword ptr [rsp + 0x18]
0x11c9d:    call       0x4cc0
0x11ca2:    jmp        0x11aad
0x11aad
0x11ca7:    mov        edx, dword ptr [rsp + 0x34]
0x11cab:    mov        esi, dword ptr [rsp + 0x30]
0x11caf:    mov        rdi, qword ptr [rsp + 0x18]
0x11cb4:    call       0x4cc0
0x11cb9:    jmp        0x11c61
0x11c61
0x11cbb:    call       0xe4e0
0x11cc0:    mov        rdi, rbp
0x11cc3:    mov        rsi, r12
0x11cc6:    xor        eax, eax
0x11cc8:    call       0x11220
0x11ccd:    mov        r13d, eax
0x11cd0:    xor        eax, eax
0x11cd2:    call       0xa820
0x11cd7:    mov        rdi, r12
0x11cda:    call       0x18d10
0x11cdf:    test       r13d, r13d
0x11ce2:    je         0x118c3
0x11ce8:    jmp        0x11748
0x11748
0x11ced:    mov        rdi, qword ptr [rsp + 0x18]
0x11cf2:    call       0x4870
0x11cf7:    jmp        0x11729
0x11729
0x11cfc:    call       0x49d0
0x11d01:    nop        word ptr cs:[rax + rax]
0x11d0c:    nop        dword ptr [rax]

```



```

0x11d10:      endbr64
0x11d14:      push     r15
0x11d16:      and      ecx, 0xf000
0x11d1c:      push     r14
0x11d1e:      mov      r14, rdi
0x11d21:      push     r13
0x11d23:      push     r12
0x11d25:      mov      r12, rdx
0x11d28:      push     rbp
0x11d29:      mov      rbp, rsi
0x11d2c:      push     rbx
0x11d2d:      sub      rsp, 0x148
0x11d34:      mov      rax, qword ptr fs:[0x28]
0x11d3d:      mov      qword ptr [rsp + 0x138], rax
0x11d45:      xor      eax, eax
0x11d47:      cmp      ecx, 0x4000
0x11d4d:      je       0x11f20
0x11d53:      mov      rdi, rdx
0x11d56:      lea      rbx, [rsp + 0x10]
0x11d5b:      call     0x49b0
0x11d60:      mov      rdi, r12
0x11d63:      mov      r13, rax
0x11d66:      call     0x1d7a0
0x11d6b:      mov      rsi, rbx
0x11d6e:      mov      rdi, rax
0x11d71:      mov      r12, rax
0x11d74:      call     0x18f40
0x11d79:      test     eax, eax
0x11d7b:      je       0x11ed0
0x11d81:      mov      rdi, r12
0x11d84:      call     0x190c0
0x11d89:      cmp      byte ptr [rax], 0
0x11d8c:      mov      r15, rax
0x11d8f:      je       0x12170
0x11d95:      cmp      r12, rax
0x11d98:      je       0x120e8
0x11d9e:      mov      edx, dword ptr [r14 + 0x90]
0x11da5:      test     edx, edx
0x11da7:      je       0x11dbb
0x11da9:      xor      esi, esi
0x11dab:      mov      rdi, r12
0x11dae:      call     0x4d90
0x11db3:      test     eax, eax
0x11db5:      je       0x12140
0x11dbb:      mov      rdi, rbp
0x11dbe:      call     0x49b0
0x11dc3:      lea      rdi, [rax + 0x20]
0x11dc7:      call     0x1d670

```

```

0x11dcc:      mov     r8, rbp
0x11dcf:      mov     esi, 1
0x11dd4:      lea     rcx, [rip + 0xf4c2]
0x11ddb:      mov     r13, rax
0x11dde:      mov     rdi, rax
0x11de1:      mov     rdx, -1
0x11de8:      xor     eax, eax
0x11dea:      call    0x4f10
0x11def:      mov     edx, 1
0x11df4:      mov     esi, 1
0x11df9:      mov     rdi, r13
0x11dfc:      call    0xe480
0x11e01:      xor     eax, eax
0x11e03:      call    0xa820
0x11e08:      mov     rdi, r13
0x11e0b:      mov     r13d, 6
0x11e11:      call    0x18d10
0x11e16:      lea     rsi, [rsp + 0xa0]
0x11e1e:      mov     rdi, rbp
0x11e21:      call    0x18f40
0x11e26:      cmp     eax, -1
0x11e29:      je      0x11ea1
0x11e2b:      mov     rsi, rbx
0x11e2e:      mov     rdi, r12
0x11e31:      call    0x18f40
0x11e36:      cmp     eax, -1
0x11e39:      je      0x121c8
0x11e3f:      mov     rax, qword ptr [rsp + 0xa0]
0x11e47:      cmp     qword ptr [rsp + 0x10], rax
0x11e4c:      jne     0x11e5a
0x11e4e:      mov     eax, dword ptr [r14 + 0x10]
0x11e52:      test    eax, eax
0x11e54:      je      0x12188
0x11e5a:      mov     r9d, dword ptr [rsp + 0xc0]
0x11e62:      mov     rdx, r12
0x11e65:      mov     rsi, rbp
0x11e68:      xor     eax, eax
0x11e6a:      mov     r8d, dword ptr [rsp + 0xbc]
0x11e72:      mov     ecx, dword ptr [rsp + 0xb8]
0x11e79:      mov     rdi, r14
0x11e7c:      call    0x11330
0x11e81:      mov     r13d, eax
0x11e84:      cmp     eax, 3
0x11e87:      je      0x11ea1
0x11e89:      jg      0x120d0
0x11e8f:      cmp     eax, 1
0x11e92:      je      0x121a8
0x11e98:      cmp     eax, 2

```

```

0x11e9b:    jne     0x12210
0x11ea1:    mov     rax, qword ptr [rsp + 0x138]
0x11ea9:    sub     rax, qword ptr fs:[0x28]
0x11eb2:    jne     0x122c1
0x11eb8:    add     rsp, 0x148
0x11ebf:    mov     eax, r13d
0x11ec2:    pop     rbx
0x11ec3:    pop     rbp
0x11ec4:    pop     r12
0x11ec6:    pop     r13
0x11ec8:    pop     r14
0x11eca:    pop     r15
0x11ecc:    ret
0x11ecd:    nop     dword ptr [rax]
0x11ed0:    mov     eax, dword ptr [rsp + 0x28]
0x11ed4:    and     eax, 0xf000
0x11ed9:    cmp     eax, 0x4000
0x11ede:    jne     0x11d81
0x11ee4:    mov     rdi, rbp
0x11ee7:    call    0x49b0
0x11eec:    mov     rdi, r12
0x11eef:    lea     rsi, [r13 + rax + 2]
0x11ef4:    call    0x1d690
0x11ef9:    mov     rdi, rax
0x11efc:    mov     r12, rax
0x11eff:    call    0x49b0
0x11f04:    mov     rsi, rbp
0x11f07:    mov     byte ptr [r12 + rax], 0x2f
0x11f0c:    lea     rdi, [r12 + rax + 1]
0x11f11:    call    0x48b0
0x11f16:    jmp     0x11d81
0x11d81:
0x11f1b:    nop     dword ptr [rax + rax]
0x11f20:    lea     rsi, [rsp + 0x10]
0x11f25:    mov     rdi, rdx
0x11f28:    call    0x18f40
0x11f2d:    test    eax, eax
0x11f2f:    jne     0x11fb7
0x11f35:    mov     eax, dword ptr [rsp + 0x28]
0x11f39:    mov     r13d, 9
0x11f3f:    and     eax, 0xf000
0x11f44:    cmp     eax, 0x4000
0x11f49:    jne     0x11ea1
0x11f4f:    mov     rdi, r12
0x11f52:    call    0x49b0
0x11f57:    mov     rdi, rbp
0x11f5a:    mov     rbx, rax
0x11f5d:    call    0x49b0

```

```

0x11f62:    lea     rdi, [rbx + rax + 2]
0x11f67:    call   0x1d670
0x11f6c:    mov     esi, 1
0x11f71:    mov     r9, rbp
0x11f74:    mov     r8, r12
0x11f77:    mov     rdi, rax
0x11f7a:    mov     r15, rax
0x11f7d:    xor     eax, eax
0x11f7f:    mov     rdx, -1
0x11f86:    lea     rcx, [rip + 0xe774]
0x11f8d:    call   0x4f10
0x11f92:    mov     esi, dword ptr [r14 + 0x90]
0x11f99:    test    esi, esi
0x11f9b:    je      0x11faf
0x11f9d:    xor     esi, esi
0x11f9f:    mov     rdi, r15
0x11fa2:    call   0x4d90
0x11fa7:    test    eax, eax
0x11fa9:    je      0x12278
0x11faf:    mov     rdi, r15
0x11fb2:    call   0x18d10
0x11fb7:    mov     rdi, rbp
0x11fba:    call   0x49b0
0x11fbf:    mov     rdi, r12
0x11fc2:    mov     rbx, rax
0x11fc5:    call   0x49b0
0x11fca:    lea     rdi, [rbx + rax + 0x21]
0x11fcf:    call   0x1d670
0x11fd4:    mov     rdi, r12
0x11fd7:    mov     r13, rax
0x11fda:    call   0x190c0
0x11fdf:    cmp     byte ptr [rax], 0
0x11fe2:    je      0x12178
0x11fe8:    mov     rdi, rbp
0x11feb:    call   0x49b0
0x11ff0:    mov     rdi, r12
0x11ff3:    mov     rbx, rax
0x11ff6:    call   0x49b0
0x11ffb:    lea     rdi, [rbx + rax + 0x20]
0x12000:    call   0x1d670
0x12005:    mov     r9, r12
0x12008:    mov     r8, rbp
0x1200b:    mov     esi, 1
0x12010:    lea     rcx, [rip + 0xf26c]
0x12017:    mov     r15, rax
0x1201a:    mov     rdi, rax
0x1201d:    xor     eax, eax
0x1201f:    mov     rdx, -1

```

```

0x12026:      call    0x4f10
0x1202b:      mov     edx, 1
0x12030:      mov     esi, 1
0x12035:      mov     rdi, r15
0x12038:      call    0xe480
0x1203d:      xor     eax, eax
0x1203f:      call    0xa820
0x12044:      mov     rdi, r15
0x12047:      call    0x18d10
0x1204c:      mov     r9, r12
0x1204f:      mov     r8, rbp
0x12052:      mov     rdi, r13
0x12055:      lea     rcx, [rip + 0xf22e]
0x1205c:      mov     rdx, -1
0x12063:      mov     esi, 1
0x12068:      xor     eax, eax
0x1206a:      call    0x4f10
0x1206f:      mov     esi, 1
0x12074:      mov     rdi, r13
0x12077:      call    0x193f0
0x1207c:      mov     rdi, r13
0x1207f:      mov     r13d, 0xc
0x12085:      mov     ebx, eax
0x12087:      call    0x18d10
0x1208c:      mov     rdi, qword ptr [r14 + 0x18]
0x12090:      call    0xc4b0
0x12095:      mov     eax, ebx
0x12097:      and     eax, 0x7f
0x1209a:      add     eax, 1
0x1209d:      cmp     al, 1
0x1209f:      jg      0x11ea1
0x120a5:      mov     r13d, 8
0x120ab:      test    bl, 0x7f
0x120ae:      jne     0x11ea1
0x120b4:      and     bh, 0xff
0x120b7:      jne     0x122b0
0x120bd:      mov     r13d, 1
0x120c3:      jmp     0x11ea1
0x11ea1
0x120c8:      nop     dword ptr [rax + rax]
0x120d0:      cmp     eax, 8
0x120d3:      jne     0x12210
0x120d9:      mov     r13d, 0xa
0x120df:      jmp     0x11ea1
0x11ea1
0x120e4:      nop     dword ptr [rax]
0x120e8:      mov     rdi, r12
0x120eb:      call    0x49b0

```

```

0x120f0:    lea     rdi, [rax + 3]
0x120f4:    mov     r13, rax
0x120f7:    call    0x1d670
0x120fc:    mov     ecx, 0x2f2e
0x12101:    lea     rdx, [r13 + 1]
0x12105:    mov     rsi, r15
0x12108:    mov     word ptr [rax], cx
0x1210b:    lea     rcx, [rax + 2]
0x1210f:    mov     r12, rax
0x12112:    mov     rdi, rcx
0x12115:    mov     qword ptr [rsp + 8], rcx
0x1211a:    call    0x4bb0
0x1211f:    mov     rdi, r15
0x12122:    call    0x18d10
0x12127:    mov     edx, dword ptr [r14 + 0x90]
0x1212e:    mov     r15, qword ptr [rsp + 8]
0x12133:    test    edx, edx
0x12135:    jne     0x11da9
0x1213b:    jmp     0x11dbb
0x11dbb
0x12140:    call    0xe4e0
0x12145:    xor     eax, eax
0x12147:    mov     rsi, r15
0x1214a:    mov     rdi, r14
0x1214d:    call    0x11220
0x12152:    mov     r13d, eax
0x12155:    xor     eax, eax
0x12157:    call    0xa820
0x1215c:    test    r13d, r13d
0x1215f:    je      0x11dbb
0x12165:    jmp     0x11ea1
0x11ea1
0x1216a:    nop     word ptr [rax + rax]
0x12170:    mov     rdi, r12
0x12173:    call    0x18d10
0x12178:    mov     r13d, 4
0x1217e:    jmp     0x11ea1
0x11ea1
0x12183:    nop     dword ptr [rax + rax]
0x12188:    xor     eax, eax
0x1218a:    call    0x18ab0
0x1218f:    mov     eax, dword ptr [rsp + 0x28]
0x12193:    and     eax, 0xf000
0x12198:    sub     eax, 0x4000
0x1219d:    test    eax, 0xfffffb000
0x121a2:    je      0x12250
0x121a8:    mov     rdi, rbp
0x121ab:    mov     r13d, 5

```

```

0x121b1:      call    0x4870
0x121b6:      cmp     eax, -1
0x121b9:      jne     0x120bd
0x121bf:      jmp     0x11ea1
0x11ea1
0x121c4:      nop     dword ptr [rax]
0x121c8:      movzx   r13d, byte ptr [r15 - 1]
0x121cd:      mov     byte ptr [r15 - 1], 0
0x121d2:      mov     rsi, rbx
0x121d5:      cmp     byte ptr [r12], 0
0x121da:      je      0x12200
0x121dc:      mov     rdi, r12
0x121df:      call    0x18f40
0x121e4:      mov     byte ptr [r15 - 1], r13b
0x121e8:      cmp     eax, -1
0x121eb:      jne     0x11e3f
0x121f1:      mov     r13d, 7
0x121f7:      jmp     0x11ea1
0x11ea1
0x121fc:      nop     dword ptr [rax]
0x12200:      lea     rdi, [rip + 0xf0a0]
0x12207:      call    0x18f40
0x1220c:      jmp     0x121e4
0x121e4
0x1220e:      nop
0x12210:      sub     r13d, 1
0x12214:      lea     rax, [rip + 0x18205]
0x1221b:      xor     ecx, ecx
0x1221d:      mov     rsi, rbp
0x12220:      movsxd  r13, r13d
0x12223:      mov     r8d, 0xe
0x12229:      lea     rdi, [rip + 0xf079]
0x12230:      mov     rdx, qword ptr [rax + r13*8]
0x12234:      xor     eax, eax
0x12236:      mov     r13d, 0xb
0x1223c:      call    0xf7e0
0x12241:      jmp     0x11ea1
0x11ea1
0x12246:      nop     word ptr cs:[rax + rax]
0x12250:      mov     rdi, r12
0x12253:      call    0x4870
0x12258:      mov     rsi, r12
0x1225b:      mov     rdi, rbp
0x1225e:      call    0x4e10
0x12263:      cmp     eax, -1
0x12266:      jne     0x121a8
0x1226c:      jmp     0x12178
0x12178

```

```

0x12271:      nop      dword ptr [rax]
0x12278:      call     0xe4e0
0x1227d:      mov      rdi, r14
0x12280:      mov      rsi, r15
0x12283:      xor      eax, eax
0x12285:      call     0x11220
0x1228a:      mov      r13d, eax
0x1228d:      xor      eax, eax
0x1228f:      call     0xa820
0x12294:      mov      rdi, r15
0x12297:      call     0x18d10
0x1229c:      test     r13d, r13d
0x1229f:      je        0x11fb7
0x122a5:      jmp      0x11ea1
0x11ea1
0x122aa:      nop      word ptr [rax + rax]
0x122b0:      lea      rdi, [rip + 0xfef3]
0x122b7:      call     0x195e0
0x122bc:      jmp      0x11ea1
0x11ea1
0x122c1:      call     0x49d0
0x122c6:      nop      word ptr cs:[rax + rax]
0x122d0:      endbr64
0x122d4:      push     r15
0x122d6:      push     r14
0x122d8:      push     r13
0x122da:      mov      r13, rsi
0x122dd:      push     r12
0x122df:      mov      r12, rdi
0x122e2:      mov      rdi, rsi
0x122e5:      push     rbp
0x122e6:      push     rbx
0x122e7:      sub      rsp, 0x18
0x122eb:      call     0x49b0
0x122f0:      lea      rdi, [rax + 1]
0x122f4:      mov      rbx, rax
0x122f7:      call     0x1d670
0x122fc:      movsxd   r14, dword ptr [r12 + 0x30]
0x12301:      mov      rsi, r13
0x12304:      mov      rbp, rax
0x12307:      mov      rdi, rax
0x1230a:      cmp      r14, rbx
0x1230d:      cmova    r14, rbx
0x12311:      mov      rdx, r14
0x12314:      call     0x4880
0x12319:      mov      byte ptr [rbp + r14], 0
0x1231f:      mov      ebx, dword ptr [r12 + 0x68]
0x12324:      test     ebx, ebx

```



```

0x12326:      jle      0x12450
0x1232c:      mov      r15, qword ptr [r12 + 0x38]
0x12331:      xor      r12d, r12d
0x12334:      mov      qword ptr [rsp + 8], r15
0x12339:      mov      r13, r15
0x1233c:      jmp      0x1234d
0x1234d:
0x1233e:      nop
0x12340:      add      r12d, 1
0x12344:      add      r15, 0x50
0x12348:      cmp      ebx, r12d
0x1234b:      je       0x12380
0x1234d:      mov      rsi, qword ptr [r15]
0x12350:      mov      rdi, rbp
0x12353:      call     0x4b30
0x12358:      test     eax, eax
0x1235a:      jne      0x12340
0x1235c:      mov      rdi, rbp
0x1235f:      call     0x18d10
0x12364:      add      rsp, 0x18
0x12368:      mov      eax, r12d
0x1236b:      pop      rbx
0x1236c:      pop      rbp
0x1236d:      pop      r12
0x1236f:      pop      r13
0x12371:      pop      r14
0x12373:      pop      r15
0x12375:      ret
0x12376:      nop      word ptr cs:[rax + rax]
0x12380:      xor      r12d, r12d
0x12383:      jmp      0x12395
0x12395:
0x12385:      nop      dword ptr [rax]
0x12388:      add      r12d, 1
0x1238c:      add      r13, 0x50
0x12390:      cmp      ebx, r12d
0x12393:      je       0x123b0
0x12395:      mov      rsi, qword ptr [r13]
0x12399:      mov      rdi, rbp
0x1239c:      call     0x4830
0x123a1:      test     eax, eax
0x123a3:      jne      0x12388
0x123a5:      cmp      ebx, r12d
0x123a8:      jne      0x1235c
0x123aa:      nop      word ptr [rax + rax]
0x123b0:      mov      r15, qword ptr [rsp + 8]
0x123b5:      xor      r12d, r12d
0x123b8:      jmp      0x123cd

```

```

0x123cd
0x123ba:    nop        word ptr [rax + rax]
0x123c0:    add        r12d, 1
0x123c4:    add        r15, 0x50
0x123c8:    cmp        r12d, ebx
0x123cb:    je         0x12400
0x123cd:    mov        r13, qword ptr [r15]
0x123d0:    mov        rdi, r13
0x123d3:    call       0x49b0
0x123d8:    mov        rsi, r13
0x123db:    mov        rdi, rbp
0x123de:    cmp        rax, r14
0x123e1:    cmova     rax, r14
0x123e5:    mov        rdx, rax
0x123e8:    call       0x4890
0x123ed:    test       eax, eax
0x123ef:    jne        0x123c0
0x123f1:    cmp        r12d, ebx
0x123f4:    jne        0x1235c
0x123fa:    nop        word ptr [rax + rax]
0x12400:    xor        r12d, r12d
0x12403:    jmp        0x1241a
0x1241a
0x12405:    nop        dword ptr [rax]
0x12408:    add        r15, 0x50
0x1240c:    add        r12d, 1
0x12410:    mov        qword ptr [rsp + 8], r15
0x12415:    cmp        r12d, ebx
0x12418:    je         0x12450
0x1241a:    mov        r15, qword ptr [rsp + 8]
0x1241f:    mov        r13, qword ptr [r15]
0x12422:    mov        rdi, r13
0x12425:    call       0x49b0
0x1242a:    mov        rsi, r13
0x1242d:    mov        rdi, rbp
0x12430:    cmp        rax, r14
0x12433:    cmova     rax, r14
0x12437:    mov        rdx, rax
0x1243a:    call       0x4c30
0x1243f:    test       eax, eax
0x12441:    jne        0x12408
0x12443:    cmp        r12d, ebx
0x12446:    jne        0x1235c
0x1244c:    nop        dword ptr [rax]
0x12450:    xor        r12d, r12d
0x12453:    jmp        0x1235c
0x1235c
0x12458:    nop        dword ptr [rax + rax]

```

```

0x12460:      endbr64
0x12464:      sub     rsp, 0xa8
0x1246b:      mov     rax, qword ptr fs:[0x28]
0x12474:      mov     qword ptr [rsp + 0x98], rax
0x1247c:      xor     eax, eax
0x1247e:      mov     rsi, rsp
0x12481:      call    0x4b90
0x12486:      mov     r8d, eax
0x12489:      xor     eax, eax
0x1248b:      test    r8d, r8d
0x1248e:      js      0x124a4
0x12490:      mov     eax, dword ptr [rsp + 0x18]
0x12494:      and     eax, 0xf000
0x12499:      cmp     eax, 0x4000
0x1249e:      sete    al
0x124a1:      movzx   eax, al
0x124a4:      mov     rdx, qword ptr [rsp + 0x98]
0x124ac:      sub     rdx, qword ptr fs:[0x28]
0x124b5:      jne     0x124bf
0x124b7:      add     rsp, 0xa8
0x124be:      ret
0x124bf:      call    0x49d0
0x124c4:      nop     word ptr cs:[rax + rax]
0x124cf:      nop
0x124d0:      endbr64
0x124d4:      push    r13
0x124d6:      push    r12
0x124d8:      mov     r12d, esi
0x124db:      push    rbp
0x124dc:      push    rbx
0x124dd:      sub     rsp, 0xa8
0x124e4:      mov     rax, qword ptr fs:[0x28]
0x124ed:      mov     qword ptr [rsp + 0x98], rax
0x124f5:      xor     eax, eax
0x124f7:      mov     r13, rsp
0x124fa:      call    0x1d7a0
0x124ff:      cmp     byte ptr [rax], 0
0x12502:      mov     rbp, rax
0x12505:      mov     rbx, rax
0x12508:      je      0x12560
0x1250a:      nop     word ptr [rax + rax]
0x12510:      movzx   eax, byte ptr [rbx + 1]
0x12514:      add     rbx, 1
0x12518:      test    al, al
0x1251a:      je      0x12560
0x1251c:      cmp     rbp, rbx
0x1251f:      je      0x12510
0x12521:      cmp     al, 0x2f

```

```

0x12523:    jne     0x12510
0x12525:    mov     byte ptr [rbx], 0
0x12528:    mov     rsi, r13
0x1252b:    mov     rdi, rbp
0x1252e:    call    0x4b90
0x12533:    test    eax, eax
0x12535:    js      0x12590
0x12537:    mov     eax, dword ptr [rsp + 0x18]
0x1253b:    and     eax, 0xf000
0x12540:    cmp     eax, 0x4000
0x12545:    jne     0x12590
0x12547:    mov     byte ptr [rbx], 0x2f
0x1254a:    movzx   eax, byte ptr [rbx + 1]
0x1254e:    add     rbx, 1
0x12552:    test    al, al
0x12554:    jne     0x1251c
0x12556:    nop     word ptr cs:[rax + rax]
0x12560:    mov     rdi, rbp
0x12563:    call    0x18d10
0x12568:    xor     eax, eax
0x1256a:    mov     rdx, qword ptr [rsp + 0x98]
0x12572:    sub     rdx, qword ptr fs:[0x28]
0x1257b:    jne     0x125ae
0x1257d:    add     rsp, 0xa8
0x12584:    pop     rbx
0x12585:    pop     rbp
0x12586:    pop     r12
0x12588:    pop     r13
0x1258a:    ret
0x1258b:    nop     dword ptr [rax + rax]
0x12590:    mov     esi, r12d
0x12593:    mov     rdi, rbp
0x12596:    call    0x48c0
0x1259b:    test    eax, eax
0x1259d:    je      0x12547
0x1259f:    mov     rdi, rbp
0x125a2:    call    0x18d10
0x125a7:    mov     eax, 0xffffffff
0x125ac:    jmp     0x1256a
0x1256a:
0x125ae:    call    0x49d0
0x125b3:    nop     word ptr cs:[rax + rax]
0x125be:    nop
0x125c0:    endbr64
0x125c4:    push    r15
0x125c6:    movsxd  rsi, esi
0x125c9:    mov     r10, rdx
0x125cc:    movsxd  r8, r8d

```

```

0x125cf:    push    r14
0x125d1:    lea     rax, [rsi + rsi*4]
0x125d5:    mov     rsi, rdi
0x125d8:    push    r13
0x125da:    shl     rax, 4
0x125de:    push    r12
0x125e0:    push    rbp
0x125e1:    push    rbx
0x125e2:    sub     rsp, 0x68
0x125e6:    mov     qword ptr [rsp + 0x48], rdi
0x125eb:    mov     rdi, rax
0x125ee:    mov     qword ptr [rsp + 0x50], rax
0x125f3:    add     rax, qword ptr [rsi + 0x38]
0x125f7:    mov     qword ptr [rsp + 0x20], rdx
0x125fc:    mov     rdx, rax
0x125ff:    mov     rax, qword ptr [rax]
0x12602:    mov     qword ptr [rsp + 0x38], rcx
0x12607:    mov     rcx, qword ptr [rcx + 0x38]
0x1260b:    mov     qword ptr [rsp + 0x10], rax
0x12610:    lea     rax, [r8 + r8*4]
0x12614:    shl     rax, 4
0x12618:    mov     qword ptr [rsp + 0x28], r9
0x1261d:    lea     rsi, [rcx + rax]
0x12621:    mov     qword ptr [rsp + 0x58], rax
0x12626:    mov     eax, dword ptr [rdx + 0x10]
0x12629:    mov     ecx, dword ptr [rcx + rdi + 0x10]
0x1262d:    mov     rbx, qword ptr [rsi]
0x12630:    mov     qword ptr [r9], 0
0x12637:    mov     r8d, eax
0x1263a:    mov     qword ptr [r10], 0
0x12641:    mov     edi, ecx
0x12643:    and     r8d, 0xf000
0x1264a:    and     edi, 0xf000
0x12650:    cmp     r8d, 0x4000
0x12657:    je      0x12669
0x12659:    and     eax, 0xd000
0x1265e:    cmp     eax, 0x8000
0x12663:    jne     0x12928
0x12669:    mov     rax, qword ptr [rdx + 8]
0x1266d:    mov     qword ptr [rsp + 8], rax
0x12672:    cmp     edi, 0x4000
0x12678:    je      0x12ac0
0x1267e:    and     ecx, 0xd000
0x12684:    mov     dword ptr [rsp + 0x40], 0
0x1268c:    mov     dword ptr [rsp + 0x44], 1
0x12694:    cmp     ecx, 0x8000
0x1269a:    je      0x12ac0
0x126a0:    cmp     qword ptr [rsp + 8], 0

```

```

0x126a6:      je      0x12c90
0x126ac:      mov     r14, qword ptr [rsp + 0x38]
0x126b1:      mov     rdi, qword ptr [r14 + 0x18]
0x126b5:      call    0x49b0
0x126ba:      mov     rdi, rbx
0x126bd:      mov     rbp, rax
0x126c0:      call    0x49b0
0x126c5:      lea     rdi, [rbp + rax + 2]
0x126ca:      call    0x1d670
0x126cf:      mov     r8, qword ptr [r14 + 0x18]
0x126d3:      mov     r9, rbx
0x126d6:      lea     rcx, [rip + 0xe024]
0x126dd:      mov     rdx, -1
0x126e4:      mov     esi, 1
0x126e9:      mov     rdi, rax
0x126ec:      mov     r12, rax
0x126ef:      xor     eax, eax
0x126f1:      call    0x4f10
0x126f6:      mov     rbx, qword ptr [rsp + 0x10]
0x126fb:      mov     rdi, rbx
0x126fe:      call    0x49b0
0x12703:      lea     rdi, [rax + 0x21]
0x12707:      call    0x1d670
0x1270c:      mov     r8, rbx
0x1270f:      mov     esi, 1
0x12714:      lea     rcx, [rip + 0xebbd]
0x1271b:      mov     rbp, rax
0x1271e:      mov     rdi, rax
0x12721:      mov     rdx, -1
0x12728:      xor     eax, eax
0x1272a:      call    0x4f10
0x1272f:      mov     edx, 1
0x12734:      mov     esi, 1
0x12739:      mov     rdi, rbp
0x1273c:      call    0xe480
0x12741:      xor     eax, eax
0x12743:      call    0xa820
0x12748:      mov     rdi, rbp
0x1274b:      call    0x18d10
0x12750:      mov     rdi, qword ptr [rsp + 0x10]
0x12755:      xor     esi, esi
0x12757:      xor     eax, eax
0x12759:      call    0x4d30
0x1275e:      mov     r15d, eax
0x12761:      cmp     eax, -1
0x12764:      je      0x12c6b
0x1276a:      mov     rdi, r12
0x1276d:      xor     esi, esi

```

```

0x1276f:    xor     eax, eax
0x12771:    call   0x4d30
0x12776:    mov     rdi, r12
0x12779:    mov     dword ptr [rsp + 0x1c], eax
0x1277d:    cmp     eax, -1
0x12780:    je      0x12c77
0x12786:    call   0x18d10
0x1278b:    mov     edi, 0x8000
0x12790:    xor     r12d, r12d
0x12793:    call   0x1d670
0x12798:    mov     edi, 0x8000
0x1279d:    mov     rbp, rax
0x127a0:    call   0x1d670
0x127a5:    mov     rdi, qword ptr [rsp + 0x10]
0x127aa:    mov     r13, rax
0x127ad:    call   0x49b0
0x127b2:    lea     rdi, [rax + 0x21]
0x127b6:    call   0x1d670
0x127bb:    mov     dword ptr [rsp + 0x18], r15d
0x127c0:    movabs  rdx, 0xa3d70a3d70a3d70b
0x127ca:    mov     r14, rax
0x127cd:    mov     rax, qword ptr [rsp + 8]
0x127d2:    imul    rdx
0x127d5:    mov     rax, qword ptr [rsp + 8]
0x127da:    add     rdx, rax
0x127dd:    sar     rax, 0x3f
0x127e1:    sar     rdx, 6
0x127e5:    sub     rdx, rax
0x127e8:    mov     qword ptr [rsp + 0x30], rdx
0x127ed:    jmp     0x1282b
0x1282b:
0x127ef:    nop
0x127f0:    mov     r8, qword ptr [rsp + 0x10]
0x127f5:    mov     esi, 1
0x127fa:    mov     rdi, r14
0x127fd:    xor     eax, eax
0x127ff:    lea     rcx, [rip + 0xeab8]
0x12806:    mov     rdx, -1
0x1280d:    call   0x4f10
0x12812:    mov     edx, 1
0x12817:    mov     esi, 1
0x1281c:    mov     rdi, r14
0x1281f:    call   0xe480
0x12824:    xor     eax, eax
0x12826:    call   0xa820
0x1282b:    mov     ecx, dword ptr [rip + 0x1927f]
0x12831:    test    ecx, ecx
0x12833:    jne     0x12a30

```

```

0x12839:      mov     edi, 1
0x1283e:      call    0x18b70
0x12843:      mov     edi, dword ptr [rsp + 0x18]
0x12847:      mov     edx, 0x8000
0x1284c:      mov     rsi, rbp
0x1284f:      call    0x47a0
0x12854:      xor     edi, edi
0x12856:      mov     rbx, rax
0x12859:      call    0x18b70
0x1285e:      test    ebx, ebx
0x12860:      js       0x12af0
0x12866:      mov     rcx, qword ptr [rsp + 0x20]
0x1286b:      movsxd   rax, ebx
0x1286e:      mov     edi, 1
0x12873:      add     qword ptr [rcx], rax
0x12876:      call    0x18b70
0x1287b:      mov     edi, dword ptr [rsp + 0x1c]
0x1287f:      mov     edx, 0x8000
0x12884:      mov     rsi, r13
0x12887:      call    0x47a0
0x1288c:      xor     edi, edi
0x1288e:      mov     r15, rax
0x12891:      call    0x18b70
0x12896:      test    r15d, r15d
0x12899:      js       0x12b20
0x1289f:      mov     rax, qword ptr [rsp + 0x28]
0x128a4:      movsxd   rdx, r15d
0x128a7:      add     qword ptr [rax], rdx
0x128aa:      cmp     r15d, ebx
0x128ad:      cmovle   ebx, r15d
0x128b1:      test    ebx, ebx
0x128b3:      je       0x12b40
0x128b9:      movsxd   r15, ebx
0x128bc:      mov     rsi, r13
0x128bf:      mov     rdi, rbp
0x128c2:      mov     rdx, r15
0x128c5:      call    0x4af0
0x128ca:      test    eax, eax
0x128cc:      jne      0x12b70
0x128d2:      add     r12, r15
0x128d5:      cmp     qword ptr [rsp + 8], 0
0x128db:      je       0x127f0
0x128e1:      cmp     qword ptr [rsp + 8], 0x1869f
0x128ea:      jle      0x12a10
0x128f0:      mov     rax, r12
0x128f3:      cqo
0x128f5:      idiv     qword ptr [rsp + 0x30]
0x128fa:      mov     r8, rax

```



```

0x128fd:    mov     r9, qword ptr [rsp + 0x10]
0x12902:    mov     esi, 1
0x12907:    mov     rdi, r14
0x1290a:    xor     eax, eax
0x1290c:    lea     rcx, [rip + 0xe9b4]
0x12913:    mov     rdx, -1
0x1291a:    call    0x4f10
0x1291f:    jmp     0x12812
0x12812:
0x12924:    nop     dword ptr [rax]
0x12928:    cmp     edi, 0x4000
0x1292e:    je      0x129f0
0x12934:    and     ecx, 0xd000
0x1293a:    cmp     ecx, 0x8000
0x12940:    je      0x129f0
0x12946:    mov     r14, qword ptr [rsp + 0x38]
0x1294b:    mov     rdi, qword ptr [r14 + 0x18]
0x1294f:    call    0x49b0
0x12954:    mov     rdi, rbx
0x12957:    mov     rbp, rax
0x1295a:    call    0x49b0
0x1295f:    lea     rdi, [rbp + rax + 2]
0x12964:    call    0x1d670
0x12969:    mov     r8, qword ptr [r14 + 0x18]
0x1296d:    mov     r9, rbx
0x12970:    lea     rcx, [rip + 0xdd8a]
0x12977:    mov     rdx, -1
0x1297e:    mov     esi, 1
0x12983:    mov     rdi, rax
0x12986:    mov     r12, rax
0x12989:    xor     eax, eax
0x1298b:    call    0x4f10
0x12990:    mov     rbx, qword ptr [rsp + 0x10]
0x12995:    mov     rdi, rbx
0x12998:    call    0x49b0
0x1299d:    lea     rdi, [rax + 0x21]
0x129a1:    call    0x1d670
0x129a6:    mov     r8, rbx
0x129a9:    mov     esi, 1
0x129ae:    lea     rcx, [rip + 0xe909]
0x129b5:    mov     rdi, rax
0x129b8:    mov     rbp, rax
0x129bb:    mov     rdx, -1
0x129c2:    xor     eax, eax
0x129c4:    call    0x4f10
0x129c9:    mov     dword ptr [rsp + 0x44], 1
0x129d1:    mov     qword ptr [rsp + 8], 0
0x129da:    mov     dword ptr [rsp + 0x40], 1

```

```

0x129e2:      jmp      0x1272f
0x1272f
0x129e7:      nop      word ptr [rax + rax]
0x129f0:      mov      rax, qword ptr [rsi + 8]
0x129f4:      mov      dword ptr [rsp + 0x40], 1
0x129fc:      mov      dword ptr [rsp + 0x44], 0
0x12a04:      mov      qword ptr [rsp + 8], rax
0x12a09:      jmp      0x126a0
0x126a0
0x12a0e:      nop
0x12a10:      lea      rax, [r12 + r12*4]
0x12a14:      lea      rax, [rax + rax*4]
0x12a18:      shl      rax, 2
0x12a1c:      cqo
0x12a1e:      idiv     qword ptr [rsp + 8]
0x12a23:      mov      r8, rax
0x12a26:      jmp      0x128fd
0x128fd
0x12a2b:      nop      dword ptr [rax + rax]
0x12a30:      mov      dword ptr [rip + 0x19076], 0
0x12a3a:      xor      eax, eax
0x12a3c:      call     0xc8c0
0x12a41:      mov      edx, 4
0x12a46:      lea      rsi, [rip + 0xdae5]
0x12a4d:      lea      rdi, [rip + 0xe7a5]
0x12a54:      mov      r15, rax
0x12a57:      call     0x8070
0x12a5c:      mov      rdi, r15
0x12a5f:      mov      ebx, eax
0x12a61:      call     0xc970
0x12a66:      xor      eax, eax
0x12a68:      and      ebx, 0xfffffffff
0x12a6b:      call     0xd870
0x12a70:      xor      eax, eax
0x12a72:      call     0xd740
0x12a77:      xor      eax, eax
0x12a79:      call     0xa820
0x12a7e:      cmp      bl, 0x4e
0x12a81:      je       0x12839
0x12a87:      mov      rdi, r14
0x12a8a:      mov      r15d, dword ptr [rsp + 0x18]
0x12a8f:      mov      r12, -1
0x12a96:      call     0x18d10
0x12a9b:      mov      rdi, rbp
0x12a9e:      call     0x18d10
0x12aa3:      mov      rdi, r13
0x12aa6:      call     0x18d10
0x12aab:      mov      edi, r15d

```

```

0x12aae:      call    0x4ab0
0x12ab3:      mov     edi, dword ptr [rsp + 0x1c]
0x12ab7:      call    0x4ab0
0x12abc:      jmp     0x12b09
0x12b09:
0x12abe:      nop
0x12ac0:      mov     rcx, qword ptr [rsp + 8]
0x12ac5:      mov     rax, qword ptr [rsi + 8]
0x12ac9:      mov     dword ptr [rsp + 0x40], 0
0x12ad1:      mov     dword ptr [rsp + 0x44], 0
0x12ad9:      cmp     rcx, rax
0x12adc:      cmovge  rax, rcx
0x12ae0:      mov     qword ptr [rsp + 8], rax
0x12ae5:      jmp     0x126a0
0x126a0:
0x12aea:      nop     word ptr [rax + rax]
0x12af0:      mov     edi, dword ptr [rsp + 0x18]
0x12af4:      mov     r12, -2
0x12afb:      call    0x4ab0
0x12b00:      mov     edi, dword ptr [rsp + 0x1c]
0x12b04:      call    0x4ab0
0x12b09:      add     rsp, 0x68
0x12b0d:      mov     rax, r12
0x12b10:      pop     rbx
0x12b11:      pop     rbp
0x12b12:      pop     r12
0x12b14:      pop     r13
0x12b16:      pop     r14
0x12b18:      pop     r15
0x12b1a:      ret
0x12b1b:      nop     dword ptr [rax + rax]
0x12b20:      mov     edi, dword ptr [rsp + 0x18]
0x12b24:      mov     r12, -3
0x12b2b:      call    0x4ab0
0x12b30:      mov     edi, dword ptr [rsp + 0x1c]
0x12b34:      call    0x4ab0
0x12b39:      jmp     0x12b09
0x12b09:
0x12b3b:      nop     dword ptr [rax + rax]
0x12b40:      mov     r15d, dword ptr [rsp + 0x18]
0x12b45:      mov     rdi, r14
0x12b48:      call    0x18d10
0x12b4d:      mov     rdi, rbp
0x12b50:      call    0x18d10
0x12b55:      mov     rdi, r13
0x12b58:      call    0x18d10
0x12b5d:      mov     edi, r15d
0x12b60:      call    0x4ab0

```

```

0x12b65:      mov     edi, dword ptr [rsp + 0x1c]
0x12b69:      call   0x4ab0
0x12b6e:      jmp     0x12b09
0x12b09
0x12b70:      mov     r15d, dword ptr [rsp + 0x18]
0x12b75:      xor     eax, eax
0x12b77:      nop     word ptr [rax + rax]
0x12b80:      movzx   ecx, byte ptr [r13 + rax]
0x12b86:      mov     edx, eax
0x12b88:      cmp     byte ptr [rbp + rax], cl
0x12b8c:      jne     0x12b99
0x12b8e:      lea     edx, [rax + 1]
0x12b91:      add     rax, 1
0x12b95:      cmp     ebx, eax
0x12b97:      jg      0x12b80
0x12b99:      movsxd  rdx, edx
0x12b9c:      add     r12, rdx
0x12b9f:      mov     edx, dword ptr [rsp + 0x40]
0x12ba3:      test    edx, edx
0x12ba5:      jne     0x12bea
0x12ba7:      mov     rax, qword ptr [rsp + 0x48]
0x12bac:      mov     rcx, qword ptr [rsp + 0x50]
0x12bb1:      mov     rax, qword ptr [rax + 0x38]
0x12bb5:      mov     rax, qword ptr [rax + rcx + 8]
0x12bba:      mov     rcx, qword ptr [rsp + 0x20]
0x12bbf:      mov     qword ptr [rcx], rax
0x12bc2:      mov     eax, dword ptr [rsp + 0x44]
0x12bc6:      test    eax, eax
0x12bc8:      jne     0x12c24
0x12bca:      mov     rax, qword ptr [rsp + 0x38]
0x12bcf:      mov     rcx, qword ptr [rsp + 0x58]
0x12bd4:      mov     rax, qword ptr [rax + 0x38]
0x12bd8:      mov     rax, qword ptr [rax + rcx + 8]
0x12bdd:      mov     rcx, qword ptr [rsp + 0x28]
0x12be2:      mov     qword ptr [rcx], rax
0x12be5:      jmp     0x12b45
0x12b45
0x12bea:      mov     edi, 1
0x12bef:      call   0x18b70
0x12bf4:      mov     rbx, qword ptr [rsp + 0x20]
0x12bf9:      jmp     0x12c07
0x12c07
0x12bfb:      nop     dword ptr [rax + rax]
0x12c00:      cdqe
0x12c02:      add     qword ptr [rbx], rax
0x12c05:      je      0x12c1b
0x12c07:      mov     edx, 0x8000
0x12c0c:      mov     rsi, rbp

```

```

0x12c0f:      mov     edi, r15d
0x12c12:      call    0x47a0
0x12c17:      test    eax, eax
0x12c19:      jg       0x12c00
0x12c1b:      xor     edi, edi
0x12c1d:      call    0x18b70
0x12c22:      jmp      0x12bc2
0x12bc2:
0x12c24:      mov     edi, 1
0x12c29:      call    0x18b70
0x12c2e:      mov     qword ptr [rsp + 8], rbp
0x12c33:      mov     rbx, qword ptr [rsp + 0x28]
0x12c38:      mov     ebp, dword ptr [rsp + 0x1c]
0x12c3c:      jmp      0x12c47
0x12c47:
0x12c3e:      nop
0x12c40:      cdqe
0x12c42:      add     qword ptr [rbx], rax
0x12c45:      je       0x12c5a
0x12c47:      mov     edx, 0x8000
0x12c4c:      mov     rsi, r13
0x12c4f:      mov     edi, ebp
0x12c51:      call    0x47a0
0x12c56:      test    eax, eax
0x12c58:      jg       0x12c40
0x12c5a:      xor     edi, edi
0x12c5c:      mov     rbp, qword ptr [rsp + 8]
0x12c61:      call    0x18b70
0x12c66:      jmp      0x12b45
0x12b45:
0x12c6b:      mov     r12, -4
0x12c72:      jmp      0x12b09
0x12b09:
0x12c77:      call    0x18d10
0x12c7c:      mov     edi, r15d
0x12c7f:      mov     r12, -5
0x12c86:      call    0x4ab0
0x12c8b:      jmp      0x12b09
0x12b09:
0x12c90:      xor     r12d, r12d
0x12c93:      jmp      0x12b09
0x12b09:
0x12c98:      nop     dword ptr [rax + rax]
0x12ca0:      endbr64
0x12ca4:      push    r15
0x12ca6:      push    r14
0x12ca8:      push    r13
0x12caa:      push    r12

```

```

0x12cac:    push    rbp
0x12cad:    mov     rbp, rsi
0x12cb0:    push    rbx
0x12cb1:    mov     rbx, rdi
0x12cb4:    sub     rsp, 0x38
0x12cb8:    mov     r13d, dword ptr [rdi + 0x28]
0x12cbc:    mov     rdi, qword ptr [rip + 0x1780d]
0x12cc3:    mov     rax, qword ptr fs:[0x28]
0x12ccc:    mov     qword ptr [rsp + 0x28], rax
0x12cd1:    xor     eax, eax
0x12cd3:    mov     r14d, dword ptr [rsi + 0x28]
0x12cd7:    call    0xdb60
0x12cdc:    xor     eax, eax
0x12cde:    call    0xa820
0x12ce3:    mov     r15, qword ptr [rbx + 0x18]
0x12ce7:    mov     rsi, qword ptr [rbp + 0x18]
0x12ceb:    mov     rdi, r15
0x12cee:    call    0x4b30
0x12cf3:    mov     rdx, qword ptr [rbx + 0x38]
0x12cf7:    mov     r8d, eax
0x12cfa:    movsxd  rax, r13d
0x12cfd:    lea     r12, [rax + rax*4]
0x12d01:    shl     r12, 4
0x12d05:    add     rdx, r12
0x12d08:    test    r8d, r8d
0x12d0b:    jne     0x12d39
0x12d0d:    movsxd  rax, r14d
0x12d10:    mov     rdi, qword ptr [rdx]
0x12d13:    mov     qword ptr [rsp + 8], rdx
0x12d18:    lea     rax, [rax + rax*4]
0x12d1c:    shl     rax, 4
0x12d20:    add     rax, qword ptr [rbp + 0x38]
0x12d24:    mov     rsi, qword ptr [rax]
0x12d27:    call    0x4b30
0x12d2c:    mov     rdx, qword ptr [rsp + 8]
0x12d31:    test    eax, eax
0x12d33:    je      0x12e40
0x12d39:    cmp     byte ptr [rdx + 0x48], 1
0x12d3d:    jne     0x12d57
0x12d3f:    movsxd  rax, r14d
0x12d42:    lea     r10, [rax + rax*4]
0x12d46:    mov     rax, qword ptr [rbp + 0x38]
0x12d4a:    shl     r10, 4
0x12d4e:    add     rax, r10
0x12d51:    cmp     byte ptr [rax + 0x48], 1
0x12d55:    je      0x12da0
0x12d57:    mov     ecx, 0x1e
0x12d5c:    xor     edx, edx

```

```

0x12d5e:    mov     rsi, r15
0x12d61:    xor     eax, eax
0x12d63:    lea     rdi, [rip + 0xed6]
0x12d6a:    call    0xf760
0x12d6f:    xor     eax, eax
0x12d71:    call    0xe4e0
0x12d76:    mov     rax, qword ptr [rsp + 0x28]
0x12d7b:    sub     rax, qword ptr fs:[0x28]
0x12d84:    jne     0x12fbc
0x12d8a:    add     rsp, 0x38
0x12d8e:    xor     eax, eax
0x12d90:    pop     rbx
0x12d91:    pop     rbp
0x12d92:    pop     r12
0x12d94:    pop     r13
0x12d96:    pop     r14
0x12d98:    pop     r15
0x12d9a:    jmp     0xa820
0xa820:
0x12d9f:    nop
0x12da0:    mov     rcx, qword ptr [rax + 8]
0x12da4:    cmp     qword ptr [rdx + 8], rcx
0x12da8:    je      0x12df2
0x12daa:    mov     edx, dword ptr [rdx + 0x10]
0x12dad:    mov     ecx, edx
0x12daf:    and     ecx, 0xf000
0x12db5:    cmp     ecx, 0x4000
0x12dbb:    je      0x12dcb
0x12dbd:    and     edx, 0xd000
0x12dc3:    cmp     edx, 0x8000
0x12dc9:    jne     0x12df2
0x12dcb:    mov     eax, dword ptr [rax + 0x10]
0x12dce:    mov     edx, eax
0x12dd0:    and     edx, 0xf000
0x12dd6:    cmp     edx, 0x4000
0x12ddc:    je      0x12ee8
0x12de2:    and     eax, 0xd000
0x12de7:    cmp     eax, 0x8000
0x12dec:    je      0x12ee8
0x12df2:    mov     esi, r13d
0x12df5:    lea     rdx, [rsp + 0x18]
0x12dfa:    lea     r9, [rsp + 0x20]
0x12dff:    mov     r8d, r14d
0x12e02:    mov     rcx, rbp
0x12e05:    mov     rdi, rbx
0x12e08:    xor     eax, eax
0x12e0a:    mov     qword ptr [rsp + 8], r10
0x12e0f:    call    0x125c0

```

```

0x12e14:      mov     r13, rax
0x12e17:      lea     eax, [rax + 5]
0x12e1a:      cmp     r13d, -5
0x12e1e:      jb      0x12f17
0x12e24:      lea     rdx, [rip + 0xeac5]
0x12e2b:      mov     r10, qword ptr [rsp + 8]
0x12e30:      movsxd  rax, dword ptr [rdx + rax*4]
0x12e34:      add     rax, rdx
0x12e37:      jmp     rax
rax
->->->                                indirect JMP
0x8070: endbr64
0x8074: push    r15
0x8076: push    r14
0x8078: push    r13
0x807a: push    r12
0x807c: mov     r12, rdi
0x807f: push    rbp
0x8080: mov     rbp, rsi
0x8083: push    rbx
0x8084: mov     ebx, edx
0x8086: sub     rsp, 0x28
0x808a: mov     rax, qword ptr fs:[0x28]
0x8093: mov     qword ptr [rsp + 0x18], rax
0x8098: mov     eax, edx
0x809a: and     eax, 0x10
0x809d: mov     dword ptr [rsp + 0xc], eax
0x80a1: jne     0x82f0
0x80a7: xor     r13d, r13d
0x80aa: xor     eax, eax
0x80ac: call   0xcff0
0x80b1: test    r12, r12
0x80b4: je      0x8103
0x80b6: mov     rdi, r12
0x80b9: call   0x7ef0
0x80be: mov     r12, rax
0x80c1: test    bl, 8
0x80c4: jne     0x8330
0x80ca: mov     rdi, r12
0x80cd: call   0xd610
0x80d2: test    bl, 0x20
0x80d5: jne     0x8310
0x80db: mov     rdi, r12
0x80de: call   0x18d10
0x80e3: test    rbp, rbp
0x80e6: je      0x8103
0x80e8: mov     rdi, rbp
0x80eb: call   0x7fe0

```



```

0x80f0: mov     r12, rax
0x80f3: mov     rdi, rax
0x80f6: call    0xd610
0x80fb: mov     rdi, r12
0x80fe: call    0x18d10
0x8103: xor     eax, eax
0x8105: call    0xd870
0x810a: xor     eax, eax
0x810c: call    0xd740
0x8111: xor     eax, eax
0x8113: call    0xa820
0x8118: test    bl, 4
0x811b: jne     0x8300
0x8121: mov     r14d, ebx
0x8124: lea     r12, [rsp + 0x14]
0x8129: and     r14d, 2
0x812d: mov     rdi, r12
0x8130: call    0xb180
0x8135: test    rax, rax
0x8138: je      0x81b2
0x813a: mov     rdx, qword ptr [rax]
0x813d: mov     rax, qword ptr [rax + 0x10]
0x8141: movzx   r15d, byte ptr [rdx]
0x8145: test    rax, rax
0x8148: je      0x8250
0x814e: cmp     byte ptr [rax + 0x1b], 0
0x8152: je      0x8250
0x8158: mov     rcx, qword ptr [rax]
0x815b: lea     rax, [rip + 0x20ebe]
0x8162: sub     rcx, rax
0x8165: movabs  rax, 0xea0ea0ea0ea0ea1
0x816f: imul    rcx
0x8172: sar     rcx, 0x3f
0x8176: sar     rdx, 1
0x8179: sub     rdx, rcx
0x817c: mov     r15d, edx
0x817f: not     r15d
0x8182: cmp     edx, 7
0x8185: jne     0x81d0
0x8187: xor     edi, edi
0x8189: xor     eax, eax
0x818b: call    0x7bd0
0x8190: xor     eax, eax
0x8192: call    0xdea0
0x8197: xor     eax, eax
0x8199: call    0xd740
0x819e: xor     eax, eax
0x81a0: call    0xa820

```

```

0x81a5: mov     rdi, r12
0x81a8: call    0xb180
0x81ad: test    rax, rax
0x81b0: jne     0x813a
0x81b2: xor     eax, eax
0x81b4: call    0xa5a0
0x81b9: xor     eax, eax
0x81bb: call    0xdea0
0x81c0: xor     eax, eax
0x81c2: call    0xd740
0x81c7: jmp     0x812d
0x812d:
0x81cc: nop     dword ptr [rax]
0x81d0: cmp     r15d, 7
0x81d4: je      0x82b0
0x81da: cmp     r15d, -0x3d
0x81de: jne     0x8256
0x81e0: test    rbp, rbp
0x81e3: je      0x8360
0x81e9: xor     edi, edi
0x81eb: movsx   r15d, byte ptr [rbp]
0x81f0: call    0xdba0
0x81f5: mov     esi, dword ptr [rsp + 0xc]
0x81f9: test    esi, esi
0x81fb: je      0x821f
0x81fd: test    r13, r13
0x8200: je      0x821f
0x8202: mov     rdi, r13
0x8205: call    0xc970
0x820a: xor     eax, eax
0x820c: call    0xd870
0x8211: xor     eax, eax
0x8213: call    0xd740
0x8218: xor     eax, eax
0x821a: call    0xa820
0x821f: cmp     r15d, 7
0x8223: je      0x82e1
0x8229: mov     eax, r15d
0x822c: mov     rdx, qword ptr [rsp + 0x18]
0x8231: sub     rdx, qword ptr fs:[0x28]
0x823a: jne     0x83d9
0x8240: add     rsp, 0x28
0x8244: pop     rbx
0x8245: pop     rbp
0x8246: pop     r12
0x8248: pop     r13
0x824a: pop     r14
0x824c: pop     r15

```

```

0x824e: ret
0x824f: nop
0x8250: cmp     r15d, 7
0x8254: je      0x82b0
0x8256: mov     eax, dword ptr [rsp + 0x14]
0x825a: lea     edx, [rax - 1]
0x825d: mov     dword ptr [rsp + 0x14], edx
0x8261: test    eax, eax
0x8263: je      0x83a8
0x8269: test    r14d, r14d
0x826c: je      0x8280
0x826e: mov     esi, ebx
0x8270: mov     edi, r15d
0x8273: call    0x7570
0x8278: test    eax, eax
0x827a: jne     0x8256
0x827c: nop     dword ptr [rax]
0x8280: test    rbp, rbp
0x8283: je      0x83b8
0x8289: mov     esi, r15d
0x828c: mov     rdi, rbp
0x828f: call    0x4a00
0x8294: test    rax, rax
0x8297: jne     0x83b8
0x829d: call    0xa5a0
0x82a2: jmp     0x8190
0x8190:
0x82a7: nop     word ptr [rax + rax]
0x82b0: xor     edi, edi
0x82b2: call    0xdba0
0x82b7: mov     ecx, dword ptr [rsp + 0xc]
0x82bb: test    ecx, ecx
0x82bd: je      0x82e1
0x82bf: test    r13, r13
0x82c2: je      0x82e1
0x82c4: mov     rdi, r13
0x82c7: call    0xc970
0x82cc: xor     eax, eax
0x82ce: call    0xd870
0x82d3: xor     eax, eax
0x82d5: call    0xd740
0x82da: xor     eax, eax
0x82dc: call    0xa820
0x82e1: xor     eax, eax
0x82e3: jmp     0x822c
0x822c:
0x82e8: nop     dword ptr [rax + rax]
0x82f0: xor     eax, eax

```

```

0x82f2: call    0xc8c0
0x82f7: mov     r13, rax
0x82fa: jmp     0x80aa
0x80aa
0x82ff: nop
0x8300: xor     eax, eax
0x8302: call    0xa5a0
0x8307: jmp     0x8121
0x8121
0x830c: nop     dword ptr [rax]
0x8310: xor     eax, eax
0x8312: call    0xccf0
0x8317: mov     rdi, r12
0x831a: call    0x18d10
0x831f: test    rbp, rbp
0x8322: jne     0x80e8
0x8328: jmp     0x8103
0x8103
0x832d: nop     dword ptr [rax]
0x8330: lea     rdi, [rip + 0x181da]
0x8337: call    0xd610
0x833c: mov     edi, 1
0x8341: call    0xdba0
0x8346: mov     rdi, r12
0x8349: call    0xd610
0x834e: test    bl, 0x20
0x8351: je      0x80db
0x8357: jmp     0x8310
0x8310
0x8359: nop     dword ptr [rax]
0x8360: xor     edi, edi
0x8362: call    0xdba0
0x8367: mov     edx, dword ptr [rsp + 0xc]
0x836b: mov     eax, 0xffffffffc3
0x8370: test    edx, edx
0x8372: je      0x822c
0x8378: test    r13, r13
0x837b: je      0x822c
0x8381: mov     rdi, r13
0x8384: call    0xc970
0x8389: xor     eax, eax
0x838b: call    0xd870
0x8390: xor     eax, eax
0x8392: call    0xd740
0x8397: xor     eax, eax
0x8399: call    0xa820
0x839e: jmp     0x8229
0x8229

```

```

0x83a3: nop      dword ptr [rax + rax]
0x83a8: call     0xd870
0x83ad: jmp      0x8190
0x8190:
0x83b2: nop      word ptr [rax + rax]
0x83b8: xor      edi, edi
0x83ba: call     0xdba0
0x83bf: mov      eax, dword ptr [rsp + 0xc]
0x83c3: test     eax, eax
0x83c5: je       0x8229
0x83cb: test     r13, r13
0x83ce: jne      0x8202
0x83d4: jmp      0x8229
0x8229:
0x83d9: call     0x49d0
0x83de: nop
0x83e0: endbr64
0x83e4: push     r15
0x83e6: push     r14
0x83e8: mov      r14, rdi
0x83eb: push     r13
0x83ed: mov      r13, rsi
0x83f0: push     r12
0x83f2: mov      r12, rdx
0x83f5: push     rbp
0x83f6: mov      rbp, rcx
0x83f9: push     rbx
0x83fa: sub      rsp, 0x38
0x83fe: mov      rax, qword ptr fs:[0x28]
0x8407: mov      qword ptr [rsp + 0x28], rax
0x840c: xor      eax, eax
0x840e: call     0xcff0
0x8413: test     r14, r14
0x8416: je       0x8420
0x8418: mov      rdi, r14
0x841b: call     0xd540
0x8420: test     r12, r12
0x8423: je       0x8450
0x8425: mov      rdi, r12
0x8428: call     0xd610
0x842d: test     rbp, rbp
0x8430: je       0x8450
0x8432: mov      rsi, r12
0x8435: mov      rdi, rbp
0x8438: xor      eax, eax
0x843a: mov      r14d, 1
0x8440: call     0x7e00
0x8445: jmp      0x8453

```

```

0x8453
0x8447: nop      word ptr [rax + rax]
0x8450: xor      r14d, r14d
0x8453: xor      eax, eax
0x8455: lea      r12, [rsp + 0x1c]
0x845a: lea      rbx, [rip + 0x17c03]
0x8461: call     0xd870
0x8466: xor      eax, eax
0x8468: call     0xd740
0x846d: xor      eax, eax
0x846f: call     0xa820
0x8474: lea      rax, [rsp + 0x20]
0x8479: mov      qword ptr [rsp], rax
0x847d: mov      rdi, r12
0x8480: call     0xb180
0x8485: test     rax, rax
0x8488: je       0x853f
0x848e: mov      rdx, qword ptr [rax]
0x8491: mov      rax, qword ptr [rax + 0x10]
0x8495: movzx    r15d, byte ptr [rdx]
0x8499: test     rax, rax
0x849c: je       0x84d7
0x849e: cmp      byte ptr [rax + 0x1b], 0
0x84a2: je       0x84d7
0x84a4: mov      rcx, qword ptr [rax]
0x84a7: lea      rax, [rip + 0x20b72]
0x84ae: sub      rcx, rax
0x84b1: movabs   rax, 0xea0ea0ea0ea0ea1
0x84bb: imul     rcx
0x84be: sar      rcx, 0x3f
0x84c2: sar      rdx, 1
0x84c5: sub      rdx, rcx
0x84c8: mov      r15d, edx
0x84cb: not      r15d
0x84ce: cmp      edx, 0x3c
0x84d1: je       0x8560
0x84d7: lea      eax, [r15 + 0x22]
0x84db: cmp      eax, 0x29
0x84de: ja       0x8505
0x84e0: movsxd   rax, dword ptr [rbx + rax*4]
0x84e4: add      rax, rbx
0x84e7: jmp      rax
rax
->->->                                indirect JMP

0x14857: lea      rdi, [rip + 0xcb3a]
0x1485e: call     0x8b00
0x14863: jmp      0x13f50
0x13f50

```

```

0x14868:      nop      dword ptr [rax + rax]
0x14870:      test     r13d, r13d
0x14873:      je       0x14931
0x14879:      test     dl, dl
0x1487b:      je       0x13f50
0x14881:      mov      rdi, rbp
0x14884:      xor      eax, eax
0x14886:      call     0x10d70
0x1488b:      movzx     edx, byte ptr [rbp + 0x6d]
0x1488f:      test     dl, dl
0x14891:      jne      0x14150
0x14897:      jmp      0x13f50
0x13f50
0x1489c:      nop      dword ptr [rax]
0x148a0:      test     r13d, r13d
0x148a3:      je       0x14956
0x148a9:      test     dil, dil
0x148ac:      je       0x13f50
0x148b2:      mov      rdi, rbp
0x148b5:      xor      eax, eax
0x148b7:      call     0x10d70
0x148bc:      movzx     edi, byte ptr [rbp + 0x6d]
0x148c0:      test     dil, dil
0x148c3:      jne      0x14150
0x148c9:      jmp      0x13f50
0x13f50
0x148ce:      test     al, al
0x148d0:      je       0x13f50
0x148d6:      jmp      0x1413c
0x1413c
0x148db:      mov      rdi, qword ptr [rbp + 0x48]
0x148df:      lea      r14, [rsp + 0x20]
0x148e4:      mov      rsi, r14
0x148e7:      call     0x191a0
0x148ec:      mov      edx, dword ptr [rsp + 0x20]
0x148f0:      mov      rax, qword ptr [rbp + 0x40]
0x148f4:      cmp      rax, qword ptr [rsp + 0x10]
0x148f9:      jne      0x1451c
0x148ff:      mov      r8d, dword ptr [rbp + 0x28]
0x14903:      cmp      r8d, edx
0x14906:      jl       0x14cb2
0x1490c:      jg       0x14ccf
0x14912:      mov      rdi, qword ptr [rbp + 0x48]
0x14916:      mov      edx, 1
0x1491b:      mov      rsi, r14
0x1491e:      call     0x19200
0x14923:      mov      rax, qword ptr [rsp + 0x28]
0x14928:      mov      qword ptr [rbp + 0x40], rax

```

```

0x1492c:      jmp      0x1451c
0x1451c
0x14931:      test     r12d, r12d
0x14934:      je       0x1488f
0x1493a:      mov      esi, dword ptr [rbp + 0x28]
0x1493d:      test     dl, dl
0x1493f:      je       0x13f50
0x14945:      mov      rdi, rbp
0x14948:      call     0x10760
0x1494d:      movzx    edx, byte ptr [rbp + 0x6d]
0x14951:      jmp      0x1488f
0x1488f
0x14956:      test     r12d, r12d
0x14959:      je       0x148c0
0x1495f:      mov      esi, dword ptr [rbp + 0x28]
0x14962:      test     dil, dil
0x14965:      je       0x13f50
0x1496b:      mov      rdi, rbp
0x1496e:      call     0x10760
0x14973:      movzx    edi, byte ptr [rbp + 0x6d]
0x14977:      jmp      0x148c0
0x148c0
0x1497c:      cmp      byte ptr [rax + 1], 0
0x14980:      je       0x14791
0x14986:      jmp      0x1478d
0x1478d
0x1498b:      movsxd   rsi, eax
0x1498e:      lea      rcx, [rip - 0x6315]
0x14995:      mov      edx, 0x50
0x1499a:      call     0x48e0
0x1499f:      jmp      0x13ed7
0x13ed7
0x149a4:      cmp      byte ptr [rax + 1], 0
0x149a8:      je       0x13f50
0x149ae:      jmp      0x1416c
0x1416c
0x149b3:      test     al, al
0x149b5:      jne      0x14048
0x149bb:      mov      byte ptr [r12 + 0x49], 1
0x149c1:      add      dword ptr [rbp + 0x70], 1
0x149c5:      jmp      0x14048
0x14048
0x149ca:      mov      edx, r8d
0x149cd:      sub      edx, eax
0x149cf:      xor      eax, eax
0x149d1:      mov      dword ptr [rdi + 0x28], edx
0x149d4:      call     0xf110
0x149d9:      jmp      0x146d0

```



```

0x146d0
0x149de:    mov     eax, dword ptr [rbp + 0x28]
0x149e1:    mov     rdi, qword ptr [rbp + 0x48]
0x149e5:    lea     rsi, [rsp + 0x20]
0x149ea:    mov     qword ptr [rsp + 0x28], r14
0x149ef:    mov     dword ptr [rsp + 0x20], eax
0x149f3:    call    0x19150
0x149f8:    mov     rax, qword ptr [r13 + 8]
0x149fc:    mov     ebx, dword ptr [rbp + 0x28]
0x149ff:    mov     qword ptr [rsp + 8], rax
0x14a04:    test    ebx, ebx
0x14a06:    js      0x14a80
0x14a08:    movsxd  rax, ebx
0x14a0b:    lea     r8, [rax + rax*4]
0x14a0f:    shl     r8, 4
0x14a13:    add     r8, qword ptr [rbp + 0x38]
0x14a17:    mov     r15, r8
0x14a1a:    jmp     0x14a2c
0x14a2c
0x14a1c:    nop     dword ptr [rax]
0x14a20:    sub     ebx, 1
0x14a23:    sub     r15, 0x50
0x14a27:    cmp     ebx, -1
0x14a2a:    je      0x14a80
0x14a2c:    mov     rsi, qword ptr [r15]
0x14a2f:    mov     rdi, qword ptr [rsp + 8]
0x14a34:    mov     rdx, r14
0x14a37:    call    0x4c30
0x14a3c:    test    eax, eax
0x14a3e:    jne     0x14a20
0x14a40:    cmp     byte ptr [rbp + 0x6e], 0
0x14a44:    mov     qword ptr [rbp + 0x40], r14
0x14a48:    je      0x14c8d
0x14a4e:    mov     dword ptr [rbp + 0x28], ebx
0x14a51:    xor     eax, eax
0x14a53:    mov     rdi, rbp
0x14a56:    call    0xf110
0x14a5b:    cmp     byte ptr [rbp + 0x6d], 0
0x14a5f:    mov     dword ptr [rbp + 0x2c], eax
0x14a62:    je      0x14a77
0x14a64:    xor     eax, eax
0x14a66:    mov     rdi, rbp
0x14a69:    call    0x10d70
0x14a6e:    mov     rax, qword ptr [rbp + 0x40]
0x14a72:    mov     qword ptr [rsp + 0x10], rax
0x14a77:    mov     byte ptr [rbp + 0x6e], 0
0x14a7b:    jmp     0x14568
0x14568

```

```

0x14a80:    mov     rax, qword ptr [rbp + 0x40]
0x14a84:    mov     dword ptr [r13], 0xffffffff
0x14a8c:    mov     qword ptr [rsp + 0x10], rax
0x14a91:    jmp     0x14568
0x14568:
0x14a96:    cmp     byte ptr [rbp + 0x6e], 0
0x14a9a:    je      0x14c0e
0x14aa0:    mov     eax, dword ptr [rbp + 0x68]
0x14aa3:    lea     ebx, [rax - 1]
0x14aa6:    test    ebx, ebx
0x14aa8:    js      0x14a80
0x14aaa:    movsxd  rax, ebx
0x14aad:    lea     r15, [rax + rax*4]
0x14ab1:    shl     r15, 4
0x14ab5:    add     r15, qword ptr [rbp + 0x38]
0x14ab9:    jmp     0x14ad1
0x14ad1:
0x14abb:    nop     dword ptr [rax + rax]
0x14ac0:    sub     ebx, 1
0x14ac3:    sub     r15, 0x50
0x14ac7:    mov     rdi, qword ptr [rsp + 8]
0x14acc:    cmp     ebx, -1
0x14acf:    je      0x14a80
0x14ad1:    mov     rsi, qword ptr [r15]
0x14ad4:    mov     rdx, r14
0x14ad7:    mov     qword ptr [rsp + 8], rdi
0x14adc:    call    0x4c30
0x14ae1:    test    eax, eax
0x14ae3:    jne     0x14ac0
0x14ae5:    jmp     0x14a40
0x14a40:
0x14aea:    mov     eax, dword ptr [rbp + 0x28]
0x14aed:    mov     rdi, qword ptr [rbp + 0x48]
0x14af1:    lea     rsi, [rsp + 0x20]
0x14af6:    mov     qword ptr [rsp + 0x28], r14
0x14afb:    mov     dword ptr [rsp + 0x20], eax
0x14aff:    call    0x19150
0x14b04:    mov     rax, qword ptr [r13 + 8]
0x14b08:    mov     ebx, dword ptr [rbp + 0x28]
0x14b0b:    mov     qword ptr [rsp + 8], rax
0x14b10:    mov     eax, dword ptr [rbp + 0x68]
0x14b13:    mov     dword ptr [rsp + 0x1c], eax
0x14b17:    cmp     ebx, eax
0x14b19:    jge     0x14ba3
0x14b1f:    movsxd  rax, ebx
0x14b22:    lea     rcx, [rax + rax*4]
0x14b26:    shl     rcx, 4
0x14b2a:    add     rcx, qword ptr [rbp + 0x38]

```

```

0x14b2e:      mov     r15, rcx
0x14b31:      jmp     0x14b45
0x14b45:
0x14b33:      nop     dword ptr [rax + rax]
0x14b38:      add     ebx, 1
0x14b3b:      add     r15, 0x50
0x14b3f:      cmp     ebx, dword ptr [rsp + 0x1c]
0x14b43:      je      0x14ba3
0x14b45:      mov     rsi, qword ptr [r15]
0x14b48:      mov     rdi, qword ptr [rsp + 8]
0x14b4d:      mov     rdx, r14
0x14b50:      call    0x4c30
0x14b55:      test    eax, eax
0x14b57:      jne     0x14b38
0x14b59:      cmp     ebx, -1
0x14b5c:      je      0x14ba3
0x14b5e:      cmp     byte ptr [rbp + 0x6e], 0
0x14b62:      mov     qword ptr [rbp + 0x40], r14
0x14b66:      je      0x14c70
0x14b6c:      mov     dword ptr [rbp + 0x28], ebx
0x14b6f:      xor     eax, eax
0x14b71:      mov     rdi, rbp
0x14b74:      call    0xf110
0x14b79:      cmp     byte ptr [rbp + 0x6d], 0
0x14b7d:      mov     dword ptr [rbp + 0x2c], eax
0x14b80:      je      0x14b95
0x14b82:      xor     eax, eax
0x14b84:      mov     rdi, rbp
0x14b87:      call    0x10d70
0x14b8c:      mov     rax, qword ptr [rbp + 0x40]
0x14b90:      mov     qword ptr [rsp + 0x10], rax
0x14b95:      mov     byte ptr [rbp + 0x6e], 0
0x14b99:      mov     rax, qword ptr [rsp + 0x10]
0x14b9e:      jmp     0x1451c
0x1451c:
0x14ba3:      mov     dword ptr [r13], 0xffffffff
0x14bab:      mov     rax, qword ptr [rbp + 0x40]
0x14baf:      jmp     0x1451c
0x1451c:
0x14bb4:      xor     ebx, ebx
0x14bb6:      cmp     byte ptr [rbp + 0x6e], 0
0x14bba:      je      0x14c41
0x14bc0:      mov     eax, dword ptr [rbp + 0x68]
0x14bc3:      mov     dword ptr [rsp + 0x1c], eax
0x14bc7:      cmp     ebx, eax
0x14bc9:      jge     0x14ba3
0x14bcb:      movsxd  rax, ebx
0x14bce:      lea     r15, [rax + rax*4]

```

```

0x14bd2:    shl     r15, 4
0x14bd6:    add     r15, qword ptr [rbp + 0x38]
0x14bda:    jmp     0x14bf2
0x14bf2:
0x14bdc:    nop     dword ptr [rax]
0x14be0:    add     ebx, 1
0x14be3:    add     r15, 0x50
0x14be7:    cmp     ebx, dword ptr [rsp + 0x1c]
0x14beb:    mov     rcx, qword ptr [rsp + 8]
0x14bf0:    je      0x14ba3
0x14bf2:    mov     rsi, qword ptr [r15]
0x14bf5:    mov     rdx, r14
0x14bf8:    mov     rdi, rcx
0x14bfb:    mov     qword ptr [rsp + 8], rcx
0x14c00:    call    0x4c30
0x14c05:    test    eax, eax
0x14c07:    jne     0x14be0
0x14c09:    jmp     0x14b59
0x14b59:
0x14c0e:    mov     eax, dword ptr [rbp + 0x28]
0x14c11:    mov     rdi, qword ptr [rbp + 0x48]
0x14c15:    lea     rsi, [rsp + 0x20]
0x14c1a:    mov     qword ptr [rsp + 0x28], r14
0x14c1f:    mov     dword ptr [rsp + 0x20], eax
0x14c23:    call    0x19150
0x14c28:    cmp     byte ptr [rbp + 0x6e], 0
0x14c2c:    jne     0x14cee
0x14c32:    mov     eax, dword ptr [rbp + 0x28]
0x14c35:    mov     rdi, qword ptr [r13 + 8]
0x14c39:    lea     ebx, [rax - 1]
0x14c3c:    jmp     0x14aa6
0x14aa6:
0x14c41:    mov     eax, dword ptr [rbp + 0x28]
0x14c44:    mov     rdi, qword ptr [rbp + 0x48]
0x14c48:    lea     rsi, [rsp + 0x20]
0x14c4d:    mov     qword ptr [rsp + 0x28], r14
0x14c52:    mov     dword ptr [rsp + 0x20], eax
0x14c56:    call    0x19150
0x14c5b:    cmp     byte ptr [rbp + 0x6e], 0
0x14c5f:    jne     0x14c67
0x14c61:    mov     eax, dword ptr [rbp + 0x28]
0x14c64:    lea     ebx, [rax + 1]
0x14c67:    mov     rcx, qword ptr [r13 + 8]
0x14c6b:    jmp     0x14bc0
0x14bc0:
0x14c70:    sub     ebx, dword ptr [rbp + 0x28]
0x14c73:    xor     ecx, ecx
0x14c75:    mov     rdx, r12

```

```

0x14c78:      mov     esi, 0x1c
0x14c7d:      mov     r8d, ebx
0x14c80:      mov     rdi, rbp
0x14c83:      call    0x13e30
0x14c88:      jmp     0x14518
0x14518
0x14c8d:      mov     r8d, dword ptr [rbp + 0x28]
0x14c91:      xor     ecx, ecx
0x14c93:      mov     rdx, r12
0x14c96:      mov     esi, 0x1b
0x14c9b:      mov     rdi, rbp
0x14c9e:      sub     r8d, ebx
0x14ca1:      call    0x13e30
0x14ca6:      jmp     0x1455f
0x1455f
0x14cab:      xor     eax, eax
0x14cad:      jmp     0x13f00
0x13f00
0x14cb2:      sub     edx, r8d
0x14cb5:      xor     ecx, ecx
0x14cb7:      mov     esi, 0x1c
0x14cbc:      mov     rdi, rbp
0x14cbf:      mov     r8d, edx
0x14cc2:      mov     rdx, r12
0x14cc5:      call    0x13e30
0x14cca:      jmp     0x14912
0x14912
0x14ccf:      sub     r8d, edx
0x14cd2:      xor     ecx, ecx
0x14cd4:      mov     rdx, r12
0x14cd7:      mov     esi, 0x1b
0x14cdc:      mov     rdi, rbp
0x14cdf:      call    0x13e30
0x14ce4:      jmp     0x14912
0x14912
0x14ce9:      call    0x49d0
0x14cee:      mov     rdi, qword ptr [r13 + 8]
0x14cf2:      jmp     0x14aa0
0x14aa0
0x14cf7:      nop     word ptr [rax + rax]
0x14d00:      endbr64
0x14d04:      push    rbp
0x14d05:      xor     eax, eax
0x14d07:      mov     rbp, rdi
0x14d0a:      mov     dword ptr [rdi + 0x28], 0
0x14d11:      call    0xf110
0x14d16:      mov     rsi, qword ptr [rbp + 0x18]
0x14d1a:      xor     edx, edx

```

```

0x14d1c:    mov     ecx, 0x1e
0x14d21:    mov     dword ptr [rbp + 0x2c], eax
0x14d24:    lea     rdi, [rip + 0xc677]
0x14d2b:    xor     eax, eax
0x14d2d:    call    0xf760
0x14d32:    mov     rax, qword ptr [rbp + 0x18]
0x14d36:    cmp     byte ptr [rax], 0x2f
0x14d39:    jne     0x14d51
0x14d3b:    cmp     byte ptr [rax + 1], 0
0x14d3f:    jne     0x14d51
0x14d41:    lea     rdi, [rip + 0xc672]
0x14d48:    call    0x8b00
0x14d4d:    mov     rax, qword ptr [rbp + 0x18]
0x14d51:    mov     edx, 0x2f
0x14d56:    mov     word ptr [rax], dx
0x14d59:    mov     rdi, qword ptr [rbp + 0x18]
0x14d5d:    mov     qword ptr [rbp + 0x78], 1
0x14d65:    call    0x49c0
0x14d6a:    mov     rdi, rbp
0x14d6d:    mov     r8d, 1
0x14d73:    xor     ecx, ecx
0x14d75:    xor     edx, edx
0x14d77:    mov     esi, 0x22
0x14d7c:    pop     rbp
0x14d7d:    jmp     0x13e30
0x13e30:
0x14d82:    nop     word ptr cs:[rax + rax]
0x14d8d:    nop     dword ptr [rax]
0x14d90:    endbr64
0x14d94:    push    rbp
0x14d95:    cmp     byte ptr [rdi + 0x6d], 0
0x14d99:    mov     rbp, rdi
0x14d9c:    mov     eax, esi
0x14d9e:    mov     byte ptr [rdi + 0x6c], sil
0x14da2:    je      0x14db0
0x14da4:    mov     esi, dword ptr [rdi + 0x28]
0x14da7:    call    0x10760
0x14dac:    movzx   eax, byte ptr [rbp + 0x6c]
0x14db0:    test    al, al
0x14db2:    jne     0x14dc0
0x14db4:    pop     rbp
0x14db5:    ret
0x14db6:    nop     word ptr cs:[rax + rax]
0x14dc0:    mov     rdi, qword ptr [rbp + 0x18]
0x14dc4:    call    0x49c0
0x14dc9:    cmp     eax, -1
0x14dcc:    jne     0x14db4
0x14dce:    xor     eax, eax

```

```

0x14dd0:    mov     rdi, rbp
0x14dd3:    call   0x14d00
0x14dd8:    xor     eax, eax
0x14dda:    mov     rdi, rbp
0x14ddd:    call   0x10ec0
0x14de2:    xor     eax, eax
0x14de4:    mov     rdi, rbp
0x14de7:    call   0x10160
0x14dec:    cmp     byte ptr [rbp + 0x6d], 0
0x14df0:    jne     0x14e10
0x14df2:    xor     eax, eax
0x14df4:    mov     rdi, rbp
0x14df7:    call   0x10240
0x14dfc:    cmp     byte ptr [rbp + 0x6d], 0
0x14e00:    je      0x14db4
0x14e02:    mov     rdi, rbp
0x14e05:    xor     eax, eax
0x14e07:    pop     rbp
0x14e08:    jmp     0x10d70
0x10d70:
0x14e0d:    nop     dword ptr [rax]
0x14e10:    mov     rdi, rbp
0x14e13:    xor     eax, eax
0x14e15:    call   0xec50
0x14e1a:    jmp     0x14df2
0x14df2:
0x14e1c:    nop     dword ptr [rax]
0x14e20:    endbr64
0x14e24:    push    r15
0x14e26:    mov     r15, rsi
0x14e29:    push    r14
0x14e2b:    push    r13
0x14e2d:    push    r12
0x14e2f:    push    rbp
0x14e30:    mov     rbp, rdi
0x14e33:    push    rbx
0x14e34:    sub     rsp, 0x58
0x14e38:    mov     edx, dword ptr [rdi + 0x70]
0x14e3b:    mov     rax, qword ptr fs:[0x28]
0x14e44:    mov     qword ptr [rsp + 0x48], rax
0x14e49:    xor     eax, eax
0x14e4b:    mov     qword ptr [rsp + 0x38], 0
0x14e54:    mov     qword ptr [rsp + 0x40], 0
0x14e5d:    mov     dword ptr [rdi + 0x90], 1
0x14e67:    test    edx, edx
0x14e69:    jne     0x15138
0x14e6f:    movsxd  rax, dword ptr [rdi + 0x28]
0x14e73:    mov     rdx, rax

```

```

0x14e76:    lea     rax, [rax + rax*4]
0x14e7a:    shl     rax, 4
0x14e7e:    add     rax, qword ptr [rdi + 0x38]
0x14e82:    mov     r12, qword ptr [rax]
0x14e85:    test    edx, edx
0x14e87:    jne     0x14ea0
0x14e89:    mov     rax, qword ptr [rdi + 0x18]
0x14e8d:    cmp     byte ptr [rax], 0x2f
0x14e90:    jne     0x15110
0x14e96:    cmp     byte ptr [rax + 1], 0
0x14e9a:    jne     0x15110
0x14ea0:    mov     rdi, r12
0x14ea3:    call    0x49b0
0x14ea8:    lea     rdi, [rax + 0x11]
0x14eac:    call    0x1d670
0x14eb1:    mov     edx, 0xe
0x14eb6:    mov     rsi, r12
0x14eb9:    lea     rdi, [rip + 0x16720]
0x14ec0:    mov     r13, rax
0x14ec3:    call    0x4880
0x14ec8:    lea     rcx, [rip + 0xc500]
0x14ecf:    mov     esi, 1
0x14ed4:    mov     rdi, r13
0x14ed7:    mov     r8, rax
0x14eda:    mov     rdx, -1
0x14ee1:    xor     eax, eax
0x14ee3:    call    0x4f10
0x14ee8:    mov     rdi, r12
0x14eeb:    call    0x49b0
0x14ef0:    mov     rdi, qword ptr [r15 + 0x18]
0x14ef4:    mov     rbx, rax
0x14ef7:    call    0x49b0
0x14efc:    lea     rdi, [rbx + rax + 2]
0x14f01:    call    0x1d670
0x14f06:    mov     r8, qword ptr [r15 + 0x18]
0x14f0a:    mov     r9, r12
0x14f0d:    lea     rcx, [rip + 0xb7ed]
0x14f14:    mov     rdi, rax
0x14f17:    mov     r14, rax
0x14f1a:    mov     esi, 1
0x14f1f:    xor     eax, eax
0x14f21:    mov     rdx, -1
0x14f28:    call    0x4f10
0x14f2d:    mov     rdi, r13
0x14f30:    lea     rsi, [rsp + 0x40]
0x14f35:    mov     rdx, r14
0x14f38:    mov     rcx, qword ptr [rip + 0x166d9]
0x14f3f:    call    0x83e0

```



```

0x14f44:    mov     rdi, r13
0x14f47:    test    rax, rax
0x14f4a:    je      0x153c3
0x14f50:    call    0x18d10
0x14f55:    movsxd  rax, dword ptr [rbp + 0x28]
0x14f59:    lea     rax, [rax + rax*4]
0x14f5d:    shl     rax, 4
0x14f61:    add     rax, qword ptr [rbp + 0x38]
0x14f65:    mov     eax, dword ptr [rax + 0x10]
0x14f68:    and     eax, 0xf000
0x14f6d:    cmp     eax, 0x4000
0x14f72:    je      0x152f0
0x14f78:    mov     rdi, qword ptr [rip + 0x15509]
0x14f7f:    call    0xdb60
0x14f84:    xor     eax, eax
0x14f86:    call    0xa820
0x14f8b:    mov     rdi, qword ptr [rsp + 0x40]
0x14f90:    xor     eax, eax
0x14f92:    call    0x19940
0x14f97:    mov     rdi, qword ptr [rsp + 0x40]
0x14f9c:    mov     r13, rax
0x14f9f:    call    0x18d10
0x14fa4:    mov     rdi, r12
0x14fa7:    xor     eax, eax
0x14fa9:    mov     rsi, r13
0x14fac:    mov     qword ptr [rsp + 0x40], r13
0x14fb1:    call    0x11190
0x14fb6:    mov     rdi, r14
0x14fb9:    mov     ebx, eax
0x14fbb:    call    0x18d10
0x14fc0:    test    ebx, ebx
0x14fc2:    jne     0x153a0
0x14fc8:    call    0x4bf0
0x14fcd:    mov     ebx, eax
0x14fcf:    call    0x49e0
0x14fd4:    movsxd  rdx, dword ptr [rbp + 0x28]
0x14fd8:    mov     r9d, ebx
0x14fdb:    mov     rsi, r12
0x14fde:    mov     r8d, eax
0x14fe1:    mov     rdi, rbp
0x14fe4:    xor     eax, eax
0x14fe6:    lea     rdx, [rdx + rdx*4]
0x14fea:    shl     rdx, 4
0x14fee:    add     rdx, qword ptr [rbp + 0x38]
0x14ff2:    mov     ecx, dword ptr [rdx + 0x10]
0x14ff5:    mov     rdx, qword ptr [rsp + 0x40]
0x14ffa:    call    0x11330
0x14fff:    mov     rdi, qword ptr [rsp + 0x40]

```

```

0x15004:    mov     ebx, eax
0x15006:    call   0x18d10
0x1500b:    lea     eax, [rbx - 1]
0x1500e:    cmp     eax, 1
0x15011:    jbe     0x15039
0x15013:    cdqe
0x15015:    lea     rdx, [rip + 0x15404]
0x1501c:    xor     ecx, ecx
0x1501e:    mov     rsi, r12
0x15021:    mov     rdx, qword ptr [rdx + rax*8]
0x15025:    mov     r8d, 0x1e
0x1502b:    lea     rdi, [rip + 0xc277]
0x15032:    xor     eax, eax
0x15034:    call   0xf7e0
0x15039:    xor     eax, eax
0x1503b:    call   0xe4e0
0x15040:    mov     rdi, rbp
0x15043:    xor     eax, eax
0x15045:    call   0x10240
0x1504a:    xor     eax, eax
0x1504c:    mov     rdi, r15
0x1504f:    call   0x10240
0x15054:    xor     eax, eax
0x15056:    call   0xa820
0x1505b:    mov     rsi, qword ptr [r15 + 0x18]
0x1505f:    xor     eax, eax
0x15061:    mov     edx, 1
0x15066:    mov     rdi, r15
0x15069:    call   0xf880
0x1506e:    test    eax, eax
0x15070:    je      0x153e0
0x15076:    xor     eax, eax
0x15078:    mov     rdi, r15
0x1507b:    call   0x10ec0
0x15080:    xor     eax, eax
0x15082:    mov     rdi, r15
0x15085:    call   0x10160
0x1508a:    cmp     byte ptr [r15 + 0x6d], 0
0x1508f:    je      0x1509b
0x15091:    mov     rdi, r15
0x15094:    xor     eax, eax
0x15096:    call   0xec50
0x1509b:    xor     eax, eax
0x1509d:    mov     rdi, r15
0x150a0:    call   0x10240
0x150a5:    cmp     byte ptr [r15 + 0x6d], 0
0x150aa:    je      0x150b6
0x150ac:    mov     rdi, r15

```

```

0x150af:    xor     eax, eax
0x150b1:    call   0x10d70
0x150b6:    mov     rsi, qword ptr [rbp + 0x18]
0x150ba:    xor     eax, eax
0x150bc:    mov     edx, 1
0x150c1:    mov     rdi, rbp
0x150c4:    call   0xf880
0x150c9:    test    eax, eax
0x150cb:    je      0x153d0
0x150d1:    xor     eax, eax
0x150d3:    mov     rdi, rbp
0x150d6:    call   0x10ec0
0x150db:    xor     eax, eax
0x150dd:    mov     rdi, rbp
0x150e0:    call   0x10160
0x150e5:    cmp     byte ptr [rbp + 0x6d], 0
0x150e9:    je      0x150f5
0x150eb:    mov     rdi, rbp
0x150ee:    xor     eax, eax
0x150f0:    call   0xec50
0x150f5:    xor     eax, eax
0x150f7:    mov     rdi, rbp
0x150fa:    call   0x10240
0x150ff:    cmp     byte ptr [rbp + 0x6d], 0
0x15103:    je      0x15110
0x15105:    mov     rdi, rbp
0x15108:    xor     eax, eax
0x1510a:    call   0x10d70
0x1510f:    nop
0x15110:    mov     rax, qword ptr [rsp + 0x48]
0x15115:    sub     rax, qword ptr fs:[0x28]
0x1511e:    jne     0x153ed
0x15124:    add     rsp, 0x58
0x15128:    pop     rbx
0x15129:    pop     rbp
0x1512a:    pop     r12
0x1512c:    pop     r13
0x1512e:    pop     r14
0x15130:    pop     r15
0x15132:    ret
0x15133:    nop     dword ptr [rax + rax]
0x15138:    mov     rdx, qword ptr [rsi + 0x18]
0x1513c:    mov     rcx, qword ptr [rip + 0x164d5]
0x15143:    lea     rsi, [rsp + 0x38]
0x15148:    lea     rdi, [rip + 0xc28d]
0x1514f:    call   0x83e0
0x15154:    test    rax, rax
0x15157:    je      0x15110

```

```

0x15159:      mov     rdi, qword ptr [rbp + 0x18]
0x1515d:      mov     rsi, qword ptr [rsp + 0x38]
0x15162:      xor     eax, eax
0x15164:      call    0x11190
0x15169:      test    eax, eax
0x1516b:      jne     0x15308
0x15171:      mov     r12, qword ptr [rsp + 0x38]
0x15176:      mov     rdi, r12
0x15179:      call    0x49b0
0x1517e:      mov     rdi, r12
0x15181:      mov     rbx, rax
0x15184:      lea     rax, [rax + 1]
0x15188:      mov     qword ptr [rsp + 0x18], rax
0x1518d:      lea     rax, [rbx + 2]
0x15191:      mov     rsi, rax
0x15194:      mov     qword ptr [rsp + 0x10], rax
0x15199:      call    0x1d690
0x1519e:      mov     qword ptr [rsp + 0x38], rax
0x151a3:      mov     byte ptr [rax + rbx], 0x2f
0x151a7:      mov     rax, qword ptr [rsp + 0x38]
0x151ac:      mov     byte ptr [rax + rbx + 1], 0
0x151b1:      mov     eax, dword ptr [rbp + 0x70]
0x151b4:      mov     dword ptr [rbp + 0x50], 0xffffffff
0x151bb:      mov     dword ptr [rbp + 0x54], eax
0x151be:      mov     rdi, rbp
0x151c1:      call    0x100e0
0x151c6:      cmp     eax, -1
0x151c9:      je      0x152ce
0x151cf:      cdq     rax
0x151d1:      lea     rax, [rax + rax*4]
0x151d5:      shl     rax, 4
0x151d9:      mov     qword ptr [rsp + 8], rax
0x151de:      add     rax, qword ptr [rbp + 0x38]
0x151e2:      mov     r12, qword ptr [rax]
0x151e5:      mov     r14d, dword ptr [rax + 0x10]
0x151e9:      call    0x49e0
0x151ee:      mov     ebx, eax
0x151f0:      call    0x4bf0
0x151f5:      mov     rdi, r12
0x151f8:      mov     r13d, eax
0x151fb:      call    0x49b0
0x15200:      mov     rcx, qword ptr [rsp + 0x10]
0x15205:      mov     rdi, qword ptr [rsp + 0x38]
0x1520a:      lea     rsi, [rcx + rax]
0x1520e:      call    0x1d690
0x15213:      mov     rcx, qword ptr [rsp + 0x18]
0x15218:      mov     rsi, r12
0x1521b:      mov     qword ptr [rsp + 0x38], rax

```

```

0x15220:    lea     rdi, [rax + rcx]
0x15224:    call   0x48b0
0x15229:    mov     eax, dword ptr [rip + 0x16881]
0x1522f:    test    eax, eax
0x15231:    jne     0x15338
0x15237:    mov     rdi, qword ptr [rip + 0x15252]
0x1523e:    call   0xdb60
0x15243:    xor     eax, eax
0x15245:    call   0xa820
0x1524a:    mov     rdx, qword ptr [rsp + 0x38]
0x1524f:    xor     eax, eax
0x15251:    mov     r9d, r13d
0x15254:    mov     r8d, ebx
0x15257:    mov     ecx, r14d
0x1525a:    mov     rsi, r12
0x1525d:    mov     rdi, rbp
0x15260:    call   0x11330
0x15265:    cmp     eax, 1
0x15268:    je      0x15320
0x1526e:    cmp     eax, 2
0x15271:    je      0x152ce
0x15273:    cmp     eax, 3
0x15276:    je      0x151be
0x1527c:    sub     eax, 1
0x1527f:    lea     rbx, [rip + 0x1519a]
0x15286:    xor     ecx, ecx
0x15288:    mov     rsi, r12
0x1528b:    cdq     rsi
0x1528d:    mov     r8d, 0xe
0x15293:    lea     rdi, [rip + 0xc00f]
0x1529a:    mov     rdx, qword ptr [rbx + rax*8]
0x1529e:    xor     eax, eax
0x152a0:    call   0xf7e0
0x152a5:    test    al, al
0x152a7:    je      0x152ce
0x152a9:    xor     eax, eax
0x152ab:    mov     rdi, rbp
0x152ae:    call   0x10240
0x152b3:    mov     rdi, r15
0x152b6:    xor     eax, eax
0x152b8:    call   0x10240
0x152bd:    mov     rdi, rbp
0x152c0:    call   0x100e0
0x152c5:    cmp     eax, -1
0x152c8:    jne     0x151cf
0x152ce:    mov     rdi, qword ptr [rsp + 0x38]
0x152d3:    test    rdi, rdi
0x152d6:    je      0x152dd

```

```

0x152d8:      call    0x18d10
0x152dd:      xor     eax, eax
0x152df:      call    0xe4e0
0x152e4:      xor     eax, eax
0x152e6:      call    0xa820
0x152eb:      jmp     0x1505b
0x1505b
0x152f0:      mov     rdi, qword ptr [rip + 0x151a1]
0x152f7:      call    0xdb60
0x152fc:      jmp     0x14f84
0x14f84
0x15301:      nop     dword ptr [rax]
0x15308:      mov     edx, 0xc
0x1530d:      xor     esi, esi
0x1530f:      lea     rdi, [rip + 0x14fea]
0x15316:      call    0x8070
0x1531b:      jmp     0x15110
0x15110
0x15320:      mov     rax, qword ptr [rbp + 0x38]
0x15324:      mov     rbx, qword ptr [rsp + 8]
0x15329:      mov     byte ptr [rax + rbx + 0x49], 0
0x1532e:      jmp     0x152a9
0x152a9
0x15333:      nop     dword ptr [rax + rax]
0x15338:      mov     dword ptr [rip + 0x1676e], 0
0x15342:      xor     eax, eax
0x15344:      call    0xc8c0
0x15349:      mov     edx, 4
0x1534e:      lea     rsi, [rip + 0xb1dd]
0x15355:      lea     rdi, [rip + 0xbe9d]
0x1535c:      mov     qword ptr [rsp + 0x28], rax
0x15361:      call    0x8070
0x15366:      mov     rdi, qword ptr [rsp + 0x28]
0x1536b:      mov     byte ptr [rsp + 0x27], al
0x1536f:      call    0xc970
0x15374:      xor     eax, eax
0x15376:      call    0xd870
0x1537b:      xor     eax, eax
0x1537d:      call    0xd740
0x15382:      xor     eax, eax
0x15384:      call    0xa820
0x15389:      movzx   eax, byte ptr [rsp + 0x27]
0x1538e:      and     eax, 0xffffffff
0x15391:      cmp     al, 0x4e
0x15393:      je      0x15237
0x15399:      jmp     0x152ce
0x152ce
0x1539e:      nop

```

```

0x153a0:    mov     rdx, qword ptr [rsp + 0x40]
0x153a5:    lea     rdi, [rip + 0xc8a4]
0x153ac:    xor     ecx, ecx
0x153ae:    xor     eax, eax
0x153b0:    mov     r8d, 0x1e
0x153b6:    mov     rsi, r12
0x153b9:    call    0xf7e0
0x153be:    mov     rdi, qword ptr [rsp + 0x40]
0x153c3:    call    0x18d10
0x153c8:    jmp     0x15110
0x15110:
0x153cd:    nop     dword ptr [rax]
0x153d0:    mov     rdi, rbp
0x153d3:    call    0x14d00
0x153d8:    jmp     0x150d1
0x150d1:
0x153dd:    nop     dword ptr [rax]
0x153e0:    mov     rdi, r15
0x153e3:    call    0x14d00
0x153e8:    jmp     0x15076
0x15076:
0x153ed:    call    0x49d0
0x153f2:    nop     word ptr cs:[rax + rax]
0x153fd:    nop     dword ptr [rax]
0x15400:    endbr64
0x15404:    push    r15
0x15406:    push    r14
0x15408:    push    r13
0x1540a:    push    r12
0x1540c:    push    rbp
0x1540d:    mov     rbp, rdi
0x15410:    push    rbx
0x15411:    sub     rsp, 0x28
0x15415:    mov     qword ptr [rsp + 0x10], rsi
0x1541a:    mov     esi, dword ptr [rdi + 0x70]
0x1541d:    test    esi, esi
0x1541f:    jne     0x15450
0x15421:    mov     ecx, dword ptr [rdi + 0x28]
0x15424:    test    ecx, ecx
0x15426:    jne     0x15450
0x15428:    mov     rax, qword ptr [rdi + 0x18]
0x1542c:    cmp     byte ptr [rax], 0x2f
0x1542f:    je      0x15440
0x15431:    add     rsp, 0x28
0x15435:    pop     rbx
0x15436:    pop     rbp
0x15437:    pop     r12
0x15439:    pop     r13

```

```

0x1543b:    pop     r14
0x1543d:    pop     r15
0x1543f:    ret
0x15440:    cmp     byte ptr [rax + 1], 0
0x15444:    jne     0x15431
0x15446:    nop     word ptr cs:[rax + rax]
0x15450:    xor     edx, edx
0x15452:    lea     rsi, [rip + 0xb0d9]
0x15459:    lea     rdi, [rip + 0xbf97]
0x15460:    call    0x8070
0x15465:    cmp     al, 0x79
0x15467:    jne     0x15431
0x15469:    mov     edx, dword ptr [rbp + 0x68]
0x1546c:    test    edx, edx
0x1546e:    jle     0x158fc
0x15474:    mov     rax, qword ptr [rbp + 0x38]
0x15478:    xor     ecx, ecx
0x1547a:    add     rax, 0x49
0x1547e:    jmp     0x1548b
0x1548b:
0x15480:    add     ecx, 1
0x15483:    add     rax, 0x50
0x15487:    cmp     ecx, edx
0x15489:    je      0x15490
0x1548b:    cmp     byte ptr [rax], 0
0x1548e:    je      0x15480
0x15490:    mov     dword ptr [rsp + 0x1c], ecx
0x15494:    mov     eax, dword ptr [rbp + 0x70]
0x15497:    mov     dword ptr [rbp + 0x50], 0xffffffff
0x1549e:    xor     r13d, r13d
0x154a1:    lea     r15, [rip + 0xbf69]
0x154a8:    mov     dword ptr [rsp + 0x18], 1
0x154b0:    mov     dword ptr [rbp + 0x54], eax
0x154b3:    nop     dword ptr [rax + rax]
0x154b8:    mov     rdi, rbp
0x154bb:    call    0x100e0
0x154c0:    cmp     eax, -1
0x154c3:    je      0x155c6
0x154c9:    cdqe
0x154cb:    lea     rbx, [rax + rax*4]
0x154cf:    mov     rax, qword ptr [rbp + 0x38]
0x154d3:    shl     rbx, 4
0x154d7:    mov     r12, qword ptr [rax + rbx]
0x154db:    xor     eax, eax
0x154dd:    call    0x18ab0
0x154e2:    mov     rdi, r12
0x154e5:    call    0x49b0
0x154ea:    lea     rdi, [rax + 0x20]

```



```

0x154ee:      call    0x1d670
0x154f3:      mov     r8, r12
0x154f6:      mov     rcx, r15
0x154f9:      mov     esi, 1
0x154fe:      mov     r14, rax
0x15501:      mov     rdi, rax
0x15504:      mov     rdx, -1
0x1550b:      xor     eax, eax
0x1550d:      call    0x4f10
0x15512:      mov     edx, 1
0x15517:      mov     esi, 2
0x1551c:      mov     rdi, r14
0x1551f:      call    0xe480
0x15524:      xor     eax, eax
0x15526:      call    0xa820
0x1552b:      mov     rdi, r14
0x1552e:      call    0x18d10
0x15533:      mov     edx, dword ptr [rip + 0x16577]
0x15539:      test    edx, edx
0x1553b:      jne     0x15760
0x15541:      mov     eax, dword ptr [rsp + 0x18]
0x15545:      test    eax, eax
0x15547:      jne     0x15738
0x1554d:      mov     rdi, qword ptr [rip + 0x14f54]
0x15554:      call    0xdb60
0x15559:      xor     eax, eax
0x1555b:      call    0xa820
0x15560:      cmp     r13d, 0x61
0x15564:      je      0x15728
0x1556a:      cmp     r13d, 0x73
0x1556e:      je      0x154b8
0x15574:      cmp     r13d, 0x79
0x15578:      jne     0x155c6
0x1557a:      mov     rax, qword ptr [rbp + 0x38]
0x1557e:      cmp     byte ptr [rax + rbx + 0x48], 0
0x15583:      jne     0x156e8
0x15589:      mov     rdi, qword ptr [rip + 0x14f10]
0x15590:      call    0xdb60
0x15595:      xor     eax, eax
0x15597:      call    0xa820
0x1559c:      mov     rdi, r12
0x1559f:      call    0x4990
0x155a4:      test    eax, eax
0x155a6:      jne     0x157c8
0x155ac:      mov     rax, qword ptr [rbp + 0x38]
0x155b0:      mov     rdi, rbp
0x155b3:      mov     byte ptr [rax + rbx + 0x49], 0
0x155b8:      call    0x100e0

```

```

0x155bd:      cmp     eax, -1
0x155c0:      jne     0x154c9
0x155c6:      mov     eax, dword ptr [rsp + 0x1c]
0x155ca:      cmp     dword ptr [rbp + 0x68], eax
0x155cd:      je      0x155e3
0x155cf:      mov     eax, dword ptr [rsp + 0x1c]
0x155d3:      mov     rdi, rbp
0x155d6:      mov     dword ptr [rbp + 0x28], eax
0x155d9:      xor     eax, eax
0x155db:      call    0xf110
0x155e0:      mov     dword ptr [rbp + 0x2c], eax
0x155e3:      xor     eax, eax
0x155e5:      mov     rdi, rbp
0x155e8:      call    0x10240
0x155ed:      mov     rdi, qword ptr [rsp + 0x10]
0x155f2:      xor     eax, eax
0x155f4:      call    0x10240
0x155f9:      xor     eax, eax
0x155fb:      call    0xe4e0
0x15600:      xor     eax, eax
0x15602:      call    0xa820
0x15607:      mov     rsi, qword ptr [rbp + 0x18]
0x1560b:      xor     eax, eax
0x1560d:      mov     edx, 1
0x15612:      mov     rdi, rbp
0x15615:      call    0xf880
0x1561a:      test    eax, eax
0x1561c:      je      0x158e0
0x15622:      xor     eax, eax
0x15624:      mov     rdi, rbp
0x15627:      call    0x10ec0
0x1562c:      xor     eax, eax
0x1562e:      mov     rdi, rbp
0x15631:      call    0x10160
0x15636:      cmp     byte ptr [rbp + 0x6d], 0
0x1563a:      jne     0x15890
0x15640:      xor     eax, eax
0x15642:      mov     rdi, rbp
0x15645:      call    0x10240
0x1564a:      cmp     byte ptr [rbp + 0x6d], 0
0x1564e:      jne     0x15880
0x15654:      mov     rax, qword ptr [rsp + 0x10]
0x15659:      mov     rdi, qword ptr [rbp + 0x18]
0x1565d:      mov     r12, qword ptr [rax + 0x18]
0x15661:      mov     rsi, r12
0x15664:      call    0x4b30
0x15669:      test    eax, eax
0x1566b:      jne     0x15431

```

```

0x15671:    mov     rdi, qword ptr [rsp + 0x10]
0x15676:    mov     edx, 1
0x1567b:    mov     rsi, r12
0x1567e:    call    0xf880
0x15683:    test    eax, eax
0x15685:    je      0x158ed
0x1568b:    mov     rbx, qword ptr [rsp + 0x10]
0x15690:    xor     eax, eax
0x15692:    mov     rdi, rbx
0x15695:    call    0x10ec0
0x1569a:    xor     eax, eax
0x1569c:    mov     rdi, rbx
0x1569f:    call    0x10160
0x156a4:    cmp     byte ptr [rbx + 0x6d], 0
0x156a8:    je      0x156b4
0x156aa:    mov     rdi, rbx
0x156ad:    xor     eax, eax
0x156af:    call    0xec50
0x156b4:    mov     rbx, qword ptr [rsp + 0x10]
0x156b9:    xor     eax, eax
0x156bb:    mov     rdi, rbx
0x156be:    call    0x10240
0x156c3:    cmp     byte ptr [rbx + 0x6d], 0
0x156c7:    je      0x15431
0x156cd:    add     rsp, 0x28
0x156d1:    mov     rdi, rbx
0x156d4:    xor     eax, eax
0x156d6:    pop     rbx
0x156d7:    pop     rbp
0x156d8:    pop     r12
0x156da:    pop     r13
0x156dc:    pop     r14
0x156de:    pop     r15
0x156e0:    jmp     0x10d70
0x10d70:
0x156e5:    nop     dword ptr [rax]
0x156e8:    mov     rdi, r12
0x156eb:    call    0x4870
0x156f0:    test    eax, eax
0x156f2:    je      0x155ac
0x156f8:    xor     eax, eax
0x156fa:    mov     ecx, 0xe
0x156ff:    mov     rsi, r12
0x15702:    lea     rdx, [rip + 0xae29]
0x15709:    lea     rdi, [rip + 0xc5e8]
0x15710:    call    0xf760
0x15715:    cmp     al, 0x79
0x15717:    je      0x154b8

```

```

0x1571d:      jmp      0x155c6
0x155c6
0x15722:      nop      word ptr [rax + rax]
0x15728:      mov      dword ptr [rsp + 0x18], 0
0x15730:      jmp      0x1557a
0x1557a
0x15735:      nop      dword ptr [rax]
0x15738:      mov      ecx, 2
0x1573d:      lea      rdx, [rip + 0xbcd9]
0x15744:      mov      rsi, r12
0x15747:      xor      eax, eax
0x15749:      lea      rdi, [rip + 0xc528]
0x15750:      call     0xf760
0x15755:      movsx    r13d, al
0x15759:      jmp      0x1554d
0x1554d
0x1575e:      nop
0x15760:      mov      dword ptr [rip + 0x16346], 0
0x1576a:      xor      eax, eax
0x1576c:      call     0xc8c0
0x15771:      mov      edx, 4
0x15776:      lea      rsi, [rip + 0xad5]
0x1577d:      lea      rdi, [rip + 0xba75]
0x15784:      mov      qword ptr [rsp + 8], rax
0x15789:      call     0x8070
0x1578e:      mov      rdi, qword ptr [rsp + 8]
0x15793:      mov      r14d, eax
0x15796:      call     0xc970
0x1579b:      xor      eax, eax
0x1579d:      and      r14d, 0xffffffff
0x157a1:      call     0xd870
0x157a6:      xor      eax, eax
0x157a8:      call     0xd740
0x157ad:      xor      eax, eax
0x157af:      call     0xa820
0x157b4:      cmp      r14b, 0x4e
0x157b8:      je       0x15541
0x157be:      jmp      0x155c6
0x155c6
0x157c3:      nop      dword ptr [rax + rax]
0x157c8:      xor      eax, eax
0x157ca:      mov      ecx, 0x12
0x157cf:      mov      rsi, r12
0x157d2:      lea      rdx, [rip + 0xad59]
0x157d9:      lea      rdi, [rip + 0xc4c0]
0x157e0:      call     0xf760
0x157e5:      cmp      al, 0x79
0x157e7:      jne      0x156f8

```

```

0x157ed:      mov     rdi, r12
0x157f0:      call    0x49b0
0x157f5:      lea     rdi, [rax + 0x21]
0x157f9:      call    0x1d670
0x157fe:      mov     rdx, -1
0x15805:      mov     r8, r12
0x15808:      lea     rcx, [rip + 0xbb38]
0x1580f:      mov     r14, rax
0x15812:      mov     rdi, rax
0x15815:      mov     esi, 1
0x1581a:      xor     eax, eax
0x1581c:      call    0x4f10
0x15821:      mov     esi, 1
0x15826:      mov     rdi, r14
0x15829:      call    0x193f0
0x1582e:      mov     rdi, r14
0x15831:      mov     dword ptr [rsp + 8], eax
0x15835:      call    0x18d10
0x1583a:      mov     rdi, qword ptr [rbp + 0x18]
0x1583e:      call    0xc4b0
0x15843:      mov     eax, dword ptr [rsp + 8]
0x15847:      mov     edx, eax
0x15849:      and     edx, 0x7f
0x1584c:      add     edx, 1
0x1584f:      cmp     dl, 1
0x15852:      jg      0x158b0
0x15854:      test    al, 0x7f
0x15856:      jne     0x156f8
0x1585c:      test    ah, 0xff
0x1585f:      je      0x155ac
0x15865:      lea     rdi, [rip + 0xb37e]
0x1586c:      call    0x195e0
0x15871:      jmp     0x156f8
0x156f8
0x15876:      nop     word ptr cs:[rax + rax]
0x15880:      mov     rdi, rbp
0x15883:      xor     eax, eax
0x15885:      call    0x10d70
0x1588a:      jmp     0x15654
0x15654
0x1588f:      nop
0x15890:      xor     eax, eax
0x15892:      mov     rdi, rbp
0x15895:      call    0xec50
0x1589a:      xor     eax, eax
0x1589c:      mov     rdi, rbp
0x1589f:      call    0x10240
0x158a4:      cmp     byte ptr [rbp + 0x6d], 0

```

```

0x158a8:      je      0x15654
0x158ae:      jmp     0x15880
0x15880:
0x158b0:      xor     eax, eax
0x158b2:      mov     ecx, 0xe
0x158b7:      mov     rsi, r12
0x158ba:      lea     rdx, [rip + 0xac71]
0x158c1:      lea     rdi, [rip + 0xc408]
0x158c8:      call    0xf760
0x158cd:      cmp     al, 0x79
0x158cf:      je      0x154b8
0x158d5:      jmp     0x155c6
0x155c6:
0x158da:      nop     word ptr [rax + rax]
0x158e0:      mov     rdi, rbp
0x158e3:      call    0x14d00
0x158e8:      jmp     0x15622
0x15622:
0x158ed:      mov     rdi, qword ptr [rsp + 0x10]
0x158f2:      call    0x14d00
0x158f7:      jmp     0x1568b
0x1568b:
0x158fc:      mov     dword ptr [rsp + 0x1c], 0
0x15904:      jmp     0x15494
0x15494:
0x15909:      nop     dword ptr [rax]
0x15910:      endbr64
0x15914:      push    r13
0x15916:      xor     edx, edx
0x15918:      push    r12
0x1591a:      push    rbp
0x1591b:      mov     rbp, rsi
0x1591e:      push    rbx
0x1591f:      mov     rbx, rdi
0x15922:      lea     rdi, [rip + 0xbaf9]
0x15929:      sub     rsp, 0x18
0x1592d:      mov     rcx, qword ptr [rip + 0x15cd4]
0x15934:      mov     rax, qword ptr fs:[0x28]
0x1593d:      mov     qword ptr [rsp + 8], rax
0x15942:      xor     eax, eax
0x15944:      mov     rsi, rsp
0x15947:      mov     qword ptr [rsp], 0
0x1594f:      call    0x83e0
0x15954:      test    rax, rax
0x15957:      je      0x15a0b
0x1595d:      mov     rdi, qword ptr [rsp]
0x15961:      cmp     byte ptr [rdi], 0
0x15964:      je      0x15a06

```

```

0x1596a:    xor     eax, eax
0x1596c:    call   0x19940
0x15971:    mov     rdi, qword ptr [rsp]
0x15975:    mov     r13, rax
0x15978:    call   0x18d10
0x1597d:    mov     rdi, r13
0x15980:    mov     qword ptr [rsp], r13
0x15984:    call   0x49b0
0x15989:    mov     rdi, r13
0x1598c:    lea     rsi, [rax + 2]
0x15990:    mov     r12, rax
0x15993:    call   0x4ce0
0x15998:    mov     esi, 0x41ff
0x1599d:    mov     qword ptr [rsp], rax
0x159a1:    mov     byte ptr [rax + r12], 0x2f
0x159a6:    mov     rax, qword ptr [rsp]
0x159aa:    mov     byte ptr [rax + r12 + 1], 0
0x159b0:    mov     rdi, qword ptr [rsp]
0x159b4:    xor     eax, eax
0x159b6:    call   0x124d0
0x159bb:    cmp     eax, -1
0x159be:    je      0x15a90
0x159c4:    mov     rsi, qword ptr [rbx + 0x18]
0x159c8:    xor     eax, eax
0x159ca:    mov     edx, 1
0x159cf:    mov     rdi, rbx
0x159d2:    call   0xf880
0x159d7:    test    eax, eax
0x159d9:    jne     0x15a30
0x159db:    mov     rdi, rbx
0x159de:    call   0x14d00
0x159e3:    mov     rsi, qword ptr [rbp + 0x18]
0x159e7:    mov     rdi, qword ptr [rbx + 0x18]
0x159eb:    call   0x4b30
0x159f0:    test    eax, eax
0x159f2:    je      0x15af0
0x159f8:    mov     rdi, rbp
0x159fb:    xor     eax, eax
0x159fd:    call   0x10240
0x15a02:    mov     rdi, qword ptr [rsp]
0x15a06:    call   0x18d10
0x15a0b:    mov     rax, qword ptr [rsp + 8]
0x15a10:    sub     rax, qword ptr fs:[0x28]
0x15a19:    jne     0x15ba4
0x15a1f:    add     rsp, 0x18
0x15a23:    pop     rbx
0x15a24:    pop     rbp
0x15a25:    pop     r12

```

```

0x15a27:    pop     r13
0x15a29:    ret
0x15a2a:    nop     word ptr [rax + rax]
0x15a30:    mov     rsi, qword ptr [rsp]
0x15a34:    mov     rdi, rbx
0x15a37:    xor     eax, eax
0x15a39:    call    0x122d0
0x15a3e:    mov     edx, dword ptr [rbx + 0x60]
0x15a41:    mov     esi, dword ptr [rbx + 0x68]
0x15a44:    mov     edi, 0
0x15a49:    mov     dword ptr [rbx + 0x28], eax
0x15a4c:    sub     edx, 2
0x15a4f:    mov     ecx, edx
0x15a51:    shr     ecx, 0x1f
0x15a54:    add     ecx, edx
0x15a56:    sar     ecx, 1
0x15a58:    sub     eax, ecx
0x15a5a:    lea     ecx, [rdx + rax]
0x15a5d:    cmovs   eax, edi
0x15a60:    cmp     esi, ecx
0x15a62:    jg      0x15a70
0x15a64:    mov     eax, esi
0x15a66:    sub     eax, edx
0x15a68:    mov     edx, 0
0x15a6d:    cmovs   eax, edx
0x15a70:    cmp     byte ptr [rbx + 0x6d], 0
0x15a74:    mov     dword ptr [rbx + 0x2c], eax
0x15a77:    jne     0x15ac0
0x15a79:    mov     rdi, rbx
0x15a7c:    xor     eax, eax
0x15a7e:    call    0x10240
0x15a83:    jmp     0x159e3
0x159e3
0x15a88:    nop     dword ptr [rax + rax]
0x15a90:    mov     rsi, qword ptr [rsp]
0x15a94:    mov     ecx, 0x1c
0x15a99:    xor     edx, edx
0x15a9b:    xor     eax, eax
0x15a9d:    lea     rdi, [rip + 0xb993]
0x15aa4:    call    0xf760
0x15aa9:    mov     rdi, qword ptr [rsp]
0x15aad:    call    0x18d10
0x15ab2:    jmp     0x15a0b
0x15a0b
0x15ab7:    nop     word ptr [rax + rax]
0x15ac0:    xor     eax, eax
0x15ac2:    mov     rdi, rbx
0x15ac5:    call    0x10d70

```



```

0x15aca:    cmp     byte ptr [rbx + 0x6d], 0
0x15ace:    je      0x15a79
0x15ad0:    mov     rdi, rbx
0x15ad3:    xor     eax, eax
0x15ad5:    call    0xec50
0x15ada:    mov     rdi, rbx
0x15add:    xor     eax, eax
0x15adf:    call    0x10240
0x15ae4:    jmp     0x159e3
0x159e3:
0x15ae9:    nop     dword ptr [rax]
0x15af0:    movsxd  rax, dword ptr [rbp + 0x28]
0x15af4:    lea     rax, [rax + rax*4]
0x15af8:    shl     rax, 4
0x15afc:    add     rax, qword ptr [rbp + 0x38]
0x15b00:    mov     rdi, qword ptr [rax]
0x15b03:    call    0x1d7a0
0x15b08:    mov     rsi, qword ptr [rbp + 0x18]
0x15b0c:    mov     edx, 1
0x15b11:    mov     rdi, rbp
0x15b14:    mov     r12, rax
0x15b17:    xor     eax, eax
0x15b19:    call    0xf880
0x15b1e:    test    eax, eax
0x15b20:    jne     0x15b40
0x15b22:    mov     rdi, rbp
0x15b25:    call    0x14d00
0x15b2a:    mov     rdi, r12
0x15b2d:    call    0x18d10
0x15b32:    jmp     0x159f8
0x159f8:
0x15b37:    nop     word ptr [rax + rax]
0x15b40:    mov     rsi, r12
0x15b43:    mov     rdi, rbp
0x15b46:    xor     eax, eax
0x15b48:    call    0x122d0
0x15b4d:    mov     edx, dword ptr [rbp + 0x60]
0x15b50:    mov     esi, dword ptr [rbp + 0x68]
0x15b53:    mov     edi, 0
0x15b58:    mov     dword ptr [rbp + 0x28], eax
0x15b5b:    sub     edx, 2
0x15b5e:    mov     ecx, edx
0x15b60:    shr     ecx, 0x1f
0x15b63:    add     ecx, edx
0x15b65:    sar     ecx, 1
0x15b67:    sub     eax, ecx
0x15b69:    lea     ecx, [rdx + rax]
0x15b6c:    cmovs   eax, edi

```

```

0x15b6f:    cmp     esi, ecx
0x15b71:    jg      0x15b7f
0x15b73:    mov     eax, esi
0x15b75:    sub     eax, edx
0x15b77:    mov     edx, 0
0x15b7c:    cmovs   eax, edx
0x15b7f:    cmp     byte ptr [rbp + 0x6d], 0
0x15b83:    mov     dword ptr [rbp + 0x2c], eax
0x15b86:    je      0x15b2a
0x15b88:    xor     eax, eax
0x15b8a:    mov     rdi, rbp
0x15b8d:    call    0x10d70
0x15b92:    cmp     byte ptr [rbp + 0x6d], 0
0x15b96:    je      0x15b2a
0x15b98:    mov     rdi, rbp
0x15b9b:    xor     eax, eax
0x15b9d:    call    0xec50
0x15ba2:    jmp     0x15b2a
0x15b2a:
0x15ba4:    call    0x49d0
0x15ba9:    nop     dword ptr [rax]
0x15bb0:    endbr64
0x15bb4:    push    r15
0x15bb6:    mov     r15, rdi
0x15bb9:    push    r14
0x15bbb:    push    r13
0x15bbd:    push    r12
0x15bbf:    mov     r12, rsi
0x15bc2:    push    rbp
0x15bc3:    push    rbx
0x15bc4:    sub     rsp, 0x48
0x15bc8:    mov     edx, dword ptr [rdi + 0x70]
0x15bcb:    mov     rax, qword ptr fs:[0x28]
0x15bd4:    mov     qword ptr [rsp + 0x38], rax
0x15bd9:    xor     eax, eax
0x15bdb:    mov     qword ptr [rsp + 0x28], 0
0x15be4:    mov     qword ptr [rsp + 0x30], 0
0x15bed:    mov     dword ptr [rdi + 0x90], 1
0x15bf7:    test    edx, edx
0x15bf9:    jne     0x15ec8
0x15bff:    movsxd  rax, dword ptr [rdi + 0x28]
0x15c03:    mov     rdx, rax
0x15c06:    lea     rax, [rax + rax*4]
0x15c0a:    shl     rax, 4
0x15c0e:    add     rax, qword ptr [rdi + 0x38]
0x15c12:    mov     rbp, qword ptr [rax]
0x15c15:    test    edx, edx
0x15c17:    jne     0x15c30

```

```

0x15c19:    mov     rax, qword ptr [rdi + 0x18]
0x15c1d:    cmp     byte ptr [rax], 0x2f
0x15c20:    jne     0x15ea0
0x15c26:    cmp     byte ptr [rax + 1], 0
0x15c2a:    jne     0x15ea0
0x15c30:    mov     rdi, rbp
0x15c33:    call    0x49b0
0x15c38:    lea     rdi, [rax + 0x11]
0x15c3c:    call    0x1d670
0x15c41:    mov     edx, 0xe
0x15c46:    mov     rsi, rbp
0x15c49:    lea     rdi, [rip + 0x15990]
0x15c50:    mov     r13, rax
0x15c53:    call    0x4880
0x15c58:    lea     rcx, [rip + 0xb7ef]
0x15c5f:    mov     esi, 1
0x15c64:    mov     rdi, r13
0x15c67:    mov     r8, rax
0x15c6a:    mov     rdx, -1
0x15c71:    xor     eax, eax
0x15c73:    call    0x4f10
0x15c78:    mov     rdi, rbp
0x15c7b:    call    0x49b0
0x15c80:    mov     rdi, qword ptr [r12 + 0x18]
0x15c85:    mov     rbx, rax
0x15c88:    call    0x49b0
0x15c8d:    lea     rdi, [rbx + rax + 2]
0x15c92:    call    0x1d670
0x15c97:    mov     r8, qword ptr [r12 + 0x18]
0x15c9c:    mov     r9, rbp
0x15c9f:    lea     rcx, [rip + 0xaa5b]
0x15ca6:    mov     rdi, rax
0x15ca9:    mov     r14, rax
0x15cac:    mov     esi, 1
0x15cb1:    xor     eax, eax
0x15cb3:    mov     rdx, -1
0x15cba:    call    0x4f10
0x15cbf:    mov     rdi, r13
0x15cc2:    lea     rsi, [rsp + 0x30]
0x15cc7:    mov     rdx, r14
0x15cca:    mov     rcx, qword ptr [rip + 0x1593f]
0x15cd1:    call    0x83e0
0x15cd6:    mov     rdi, r13
0x15cd9:    test    rax, rax
0x15cdc:    je      0x16173
0x15ce2:    call    0x18d10
0x15ce7:    movsxd  rax, dword ptr [r15 + 0x28]
0x15ceb:    lea     rax, [rax + rax*4]

```

```

0x15cef:    shl     rax, 4
0x15cf3:    add     rax, qword ptr [r15 + 0x38]
0x15cf7:    mov     eax, dword ptr [rax + 0x10]
0x15cfa:    and     eax, 0xf000
0x15cff:    cmp     eax, 0x4000
0x15d04:    je      0x160a0
0x15d0a:    mov     rdi, qword ptr [rip + 0x1479f]
0x15d11:    call    0xdb60
0x15d16:    xor     eax, eax
0x15d18:    call    0xa820
0x15d1d:    mov     rdi, qword ptr [rsp + 0x30]
0x15d22:    xor     eax, eax
0x15d24:    call    0x19940
0x15d29:    mov     rdi, qword ptr [rsp + 0x30]
0x15d2e:    mov     r13, rax
0x15d31:    call    0x18d10
0x15d36:    mov     rdi, rbp
0x15d39:    xor     eax, eax
0x15d3b:    mov     rsi, r13
0x15d3e:    mov     qword ptr [rsp + 0x30], r13
0x15d43:    call    0x11190
0x15d48:    mov     rdi, r14
0x15d4b:    mov     ebx, eax
0x15d4d:    call    0x18d10
0x15d52:    test    ebx, ebx
0x15d54:    jne     0x16150
0x15d5a:    movsxd  rax, dword ptr [r15 + 0x28]
0x15d5e:    mov     rdx, qword ptr [rsp + 0x30]
0x15d63:    mov     rsi, rbp
0x15d66:    mov     rdi, r15
0x15d69:    lea     rax, [rax + rax*4]
0x15d6d:    shl     rax, 4
0x15d71:    add     rax, qword ptr [r15 + 0x38]
0x15d75:    mov     ecx, dword ptr [rax + 0x10]
0x15d78:    xor     eax, eax
0x15d7a:    call    0x11d10
0x15d7f:    cmp     eax, 1
0x15d82:    je      0x15db6
0x15d84:    cmp     eax, 2
0x15d87:    je      0x161a0
0x15d8d:    sub     eax, 1
0x15d90:    lea     rdx, [rip + 0x14629]
0x15d97:    xor     ecx, ecx
0x15d99:    mov     rsi, rbp
0x15d9c:    cdqe
0x15d9e:    mov     r8d, 0xe
0x15da4:    lea     rdi, [rip + 0xb6b0]
0x15dab:    mov     rdx, qword ptr [rdx + rax*8]

```

```

0x15daf:    xor     eax, eax
0x15db1:    call   0xf7e0
0x15db6:    mov     rdi, qword ptr [rsp + 0x30]
0x15dbb:    call   0x18d10
0x15dc0:    xor     eax, eax
0x15dc2:    call   0xe4e0
0x15dc7:    mov     rdi, r15
0x15dca:    xor     eax, eax
0x15dcc:    call   0x10240
0x15dd1:    xor     eax, eax
0x15dd3:    mov     rdi, r12
0x15dd6:    call   0x10240
0x15ddb:    xor     eax, eax
0x15ddd:    call   0xa820
0x15de2:    mov     rsi, qword ptr [r12 + 0x18]
0x15de7:    xor     eax, eax
0x15de9:    mov     edx, 1
0x15dee:    mov     rdi, r12
0x15df1:    call   0xf880
0x15df6:    test    eax, eax
0x15df8:    je      0x16190
0x15dfe:    xor     eax, eax
0x15e00:    mov     rdi, r12
0x15e03:    call   0x10ec0
0x15e08:    xor     eax, eax
0x15e0a:    mov     rdi, r12
0x15e0d:    call   0x10160
0x15e12:    cmp     byte ptr [r12 + 0x6d], 0
0x15e18:    je      0x15e24
0x15e1a:    mov     rdi, r12
0x15e1d:    xor     eax, eax
0x15e1f:    call   0xec50
0x15e24:    xor     eax, eax
0x15e26:    mov     rdi, r12
0x15e29:    call   0x10240
0x15e2e:    cmp     byte ptr [r12 + 0x6d], 0
0x15e34:    je      0x15e40
0x15e36:    mov     rdi, r12
0x15e39:    xor     eax, eax
0x15e3b:    call   0x10d70
0x15e40:    mov     rsi, qword ptr [r15 + 0x18]
0x15e44:    xor     eax, eax
0x15e46:    mov     edx, 1
0x15e4b:    mov     rdi, r15
0x15e4e:    call   0xf880
0x15e53:    test    eax, eax
0x15e55:    je      0x16180
0x15e5b:    xor     eax, eax

```

```

0x15e5d:    mov     rdi, r15
0x15e60:    call   0x10ec0
0x15e65:    xor     eax, eax
0x15e67:    mov     rdi, r15
0x15e6a:    call   0x10160
0x15e6f:    cmp     byte ptr [r15 + 0x6d], 0
0x15e74:    je      0x15e80
0x15e76:    mov     rdi, r15
0x15e79:    xor     eax, eax
0x15e7b:    call   0xec50
0x15e80:    xor     eax, eax
0x15e82:    mov     rdi, r15
0x15e85:    call   0x10240
0x15e8a:    cmp     byte ptr [r15 + 0x6d], 0
0x15e8f:    je      0x15ea0
0x15e91:    mov     rdi, r15
0x15e94:    xor     eax, eax
0x15e96:    call   0x10d70
0x15e9b:    nop     dword ptr [rax + rax]
0x15ea0:    mov     rax, qword ptr [rsp + 0x38]
0x15ea5:    sub     rax, qword ptr fs:[0x28]
0x15eae:    jne     0x161bd
0x15eb4:    add     rsp, 0x48
0x15eb8:    pop     rbx
0x15eb9:    pop     rbp
0x15eba:    pop     r12
0x15ebc:    pop     r13
0x15ebe:    pop     r14
0x15ec0:    pop     r15
0x15ec2:    ret
0x15ec3:    nop     dword ptr [rax + rax]
0x15ec8:    mov     rdx, qword ptr [rsi + 0x18]
0x15ecc:    mov     rcx, qword ptr [rip + 0x1573d]
0x15ed3:    lea     rsi, [rsp + 0x28]
0x15ed8:    lea     rdi, [rip + 0xb591]
0x15edf:    call   0x83e0
0x15ee4:    test    rax, rax
0x15ee7:    je      0x15ea0
0x15ee9:    mov     rdi, qword ptr [r15 + 0x18]
0x15eed:    mov     rsi, qword ptr [rsp + 0x28]
0x15ef2:    xor     eax, eax
0x15ef4:    call   0x11190
0x15ef9:    mov     r13d, eax
0x15efc:    test    eax, eax
0x15efe:    jne     0x160b8
0x15f04:    mov     rbp, qword ptr [rsp + 0x28]
0x15f09:    mov     rdi, rbp
0x15f0c:    call   0x49b0

```

```

0x15f11:    mov     rdi, rbp
0x15f14:    mov     rbx, rax
0x15f17:    lea     rax, [rax + 1]
0x15f1b:    lea     r14, [rbx + 2]
0x15f1f:    mov     qword ptr [rsp + 8], rax
0x15f24:    mov     rsi, r14
0x15f27:    call    0x1d690
0x15f2c:    mov     qword ptr [rsp + 0x28], rax
0x15f31:    mov     byte ptr [rax + rbx], 0x2f
0x15f35:    mov     rax, qword ptr [rsp + 0x28]
0x15f3a:    mov     byte ptr [rax + rbx + 1], 0
0x15f3f:    mov     edx, dword ptr [r15 + 0x68]
0x15f43:    test    edx, edx
0x15f45:    jle     0x15f6a
0x15f47:    mov     rax, qword ptr [r15 + 0x38]
0x15f4b:    add     rax, 0x49
0x15f4f:    jmp     0x15f65
0x15f65:
0x15f51:    nop     dword ptr [rax]
0x15f58:    add     r13d, 1
0x15f5c:    add     rax, 0x50
0x15f60:    cmp     r13d, edx
0x15f63:    je      0x15f6a
0x15f65:    cmp     byte ptr [rax], 0
0x15f68:    je      0x15f58
0x15f6a:    mov     eax, dword ptr [r15 + 0x70]
0x15f6e:    mov     dword ptr [r15 + 0x50], 0xffffffff
0x15f76:    mov     dword ptr [r15 + 0x54], eax
0x15f7a:    nop     word ptr [rax + rax]
0x15f80:    mov     rdi, r15
0x15f83:    call    0x100e0
0x15f88:    cmp     eax, -1
0x15f8b:    je      0x16060
0x15f91:    cdqe
0x15f93:    lea     rbx, [rax + rax*4]
0x15f97:    mov     rax, qword ptr [r15 + 0x38]
0x15f9b:    shl     rbx, 4
0x15f9f:    mov     rbp, qword ptr [rax + rbx]
0x15fa3:    xor     eax, eax
0x15fa5:    call    0x18ab0
0x15faa:    mov     rdi, rbp
0x15fad:    call    0x49b0
0x15fb2:    mov     rdi, qword ptr [rsp + 0x28]
0x15fb7:    lea     rsi, [r14 + rax]
0x15fbb:    call    0x1d690
0x15fc0:    mov     rcx, qword ptr [rsp + 8]
0x15fc5:    mov     rsi, rbp
0x15fc8:    mov     qword ptr [rsp + 0x28], rax

```

```

0x15fcd:    lea     rdi, [rax + rcx]
0x15fd1:    call   0x48b0
0x15fd6:    mov     eax, dword ptr [rip + 0x15ad4]
0x15fdc:    test    eax, eax
0x15fde:    jne     0x160e0
0x15fe4:    mov     rdi, qword ptr [rip + 0x144cd]
0x15feb:    call   0xdb60
0x15ff0:    xor     eax, eax
0x15ff2:    call   0xa820
0x15ff7:    mov     rax, qword ptr [r15 + 0x38]
0x15ffb:    mov     rsi, rbp
0x15ffe:    mov     rdi, r15
0x16001:    mov     rdx, qword ptr [rsp + 0x28]
0x16006:    mov     ecx, dword ptr [rax + rbx + 0x10]
0x1600a:    xor     eax, eax
0x1600c:    call   0x11d10
0x16011:    cmp     eax, 1
0x16014:    je      0x160d0
0x1601a:    cmp     eax, 2
0x1601d:    je      0x16060
0x1601f:    cmp     eax, 3
0x16022:    je      0x15f80
0x16028:    sub     eax, 1
0x1602b:    lea     rdx, [rip + 0x1438e]
0x16032:    xor     ecx, ecx
0x16034:    mov     rsi, rbp
0x16037:    cdqe
0x16039:    mov     r8d, 0xe
0x1603f:    lea     rdi, [rip + 0xb415]
0x16046:    mov     rdx, qword ptr [rdx + rax*8]
0x1604a:    xor     eax, eax
0x1604c:    call   0xf7e0
0x16051:    test    al, al
0x16053:    jne     0x15f80
0x16059:    nop     dword ptr [rax]
0x16060:    mov     rdi, qword ptr [rsp + 0x28]
0x16065:    test    rdi, rdi
0x16068:    je      0x1606f
0x1606a:    call   0x18d10
0x1606f:    cmp     dword ptr [r15 + 0x68], r13d
0x16073:    je      0x16087
0x16075:    mov     dword ptr [r15 + 0x28], r13d
0x16079:    mov     rdi, r15
0x1607c:    xor     eax, eax
0x1607e:    call   0xf110
0x16083:    mov     dword ptr [r15 + 0x2c], eax
0x16087:    xor     eax, eax
0x16089:    call   0xe4e0

```



```

0x1608e:      xor     eax, eax
0x16090:      call    0xa820
0x16095:      jmp     0x15de2
0x15de2
0x1609a:      nop     word ptr [rax + rax]
0x160a0:      mov     rdi, qword ptr [rip + 0x14419]
0x160a7:      call    0xdb60
0x160ac:      jmp     0x15d16
0x15d16
0x160b1:      nop     dword ptr [rax]
0x160b8:      mov     edx, 0xc
0x160bd:      xor     esi, esi
0x160bf:      lea     rdi, [rip + 0x1423a]
0x160c6:      call    0x8070
0x160cb:      jmp     0x15ea0
0x15ea0
0x160d0:      mov     rax, qword ptr [r15 + 0x38]
0x160d4:      mov     byte ptr [rax + rbx + 0x49], 0
0x160d9:      jmp     0x15f80
0x15f80
0x160de:      nop
0x160e0:      mov     dword ptr [rip + 0x159c6], 0
0x160ea:      xor     eax, eax
0x160ec:      call    0xc8c0
0x160f1:      mov     edx, 4
0x160f6:      lea     rsi, [rip + 0xa435]
0x160fd:      lea     rdi, [rip + 0xb0f5]
0x16104:      mov     qword ptr [rsp + 0x18], rax
0x16109:      call    0x8070
0x1610e:      mov     rdi, qword ptr [rsp + 0x18]
0x16113:      mov     byte ptr [rsp + 0x17], al
0x16117:      call    0xc970
0x1611c:      xor     eax, eax
0x1611e:      call    0xd870
0x16123:      xor     eax, eax
0x16125:      call    0xd740
0x1612a:      xor     eax, eax
0x1612c:      call    0xa820
0x16131:      movzx   eax, byte ptr [rsp + 0x17]
0x16136:      and     eax, 0xffffffff
0x16139:      cmp     al, 0x4e
0x1613b:      je      0x15fe4
0x16141:      jmp     0x16060
0x16060
0x16146:      nop     word ptr cs:[rax + rax]
0x16150:      mov     rdx, qword ptr [rsp + 0x30]
0x16155:      lea     rdi, [rip + 0xbaf4]
0x1615c:      xor     ecx, ecx

```

```

0x1615e:    xor     eax, eax
0x16160:    mov     r8d, 0x1e
0x16166:    mov     rsi, rbp
0x16169:    call    0xf7e0
0x1616e:    mov     rdi, qword ptr [rsp + 0x30]
0x16173:    call    0x18d10
0x16178:    jmp     0x15ea0
0x15ea0:
0x1617d:    nop     dword ptr [rax]
0x16180:    mov     rdi, r15
0x16183:    call    0x14d00
0x16188:    jmp     0x15e5b
0x15e5b:
0x1618d:    nop     dword ptr [rax]
0x16190:    mov     rdi, r12
0x16193:    call    0x14d00
0x16198:    jmp     0x15dfe
0x15dfe:
0x1619d:    nop     dword ptr [rax]
0x161a0:    mov     rdi, qword ptr [rsp + 0x30]
0x161a5:    call    0x18d10
0x161aa:    xor     eax, eax
0x161ac:    call    0xe4e0
0x161b1:    xor     eax, eax
0x161b3:    call    0xa820
0x161b8:    jmp     0x15ea0
0x15ea0:
0x161bd:    call    0x49d0
0x161c2:    nop     word ptr cs:[rax + rax]
0x161cd:    nop     dword ptr [rax]
0x161d0:    endbr64
0x161d4:    push    r13
0x161d6:    xor     r13d, r13d
0x161d9:    push    r12
0x161db:    lea     r12, [rip + 0xc203]
0x161e2:    push    rbp
0x161e3:    mov     rbp, rdi
0x161e6:    push    rbx
0x161e7:    mov     rbx, rsi
0x161ea:    sub     rsp, 8
0x161ee:    mov     rdx, qword ptr [rdi + 0x38]
0x161f2:    test    rdx, rdx
0x161f5:    je      0x1621a
0x161f7:    movsxd  rax, dword ptr [rdi + 0x28]
0x161fb:    lea     rax, [rax + rax*4]
0x161ff:    shl     rax, 4
0x16203:    mov     rdi, qword ptr [rdx + rax]
0x16207:    test    rdi, rdi

```

```

0x1620a:      je      0x1621a
0x1620c:      call   0x1d7a0
0x16211:      mov     r13d, 1
0x16217:      mov     r12, rax
0x1621a:      xor     edx, edx
0x1621c:      mov     rsi, qword ptr [rbp + 0x18]
0x16220:      cmp     rbx, -1
0x16224:      mov     rdi, rbp
0x16227:      sete    dl
0x1622a:      xor     eax, eax
0x1622c:      call    0xf880
0x16231:      test    eax, eax
0x16233:      je      0x16278
0x16235:      cmp     rbx, -1
0x16239:      je      0x16288
0x1623b:      mov     dword ptr [rbp + 0x28], 0
0x16242:      mov     rdi, rbp
0x16245:      xor     eax, eax
0x16247:      call    0xf110
0x1624c:      mov     dword ptr [rbp + 0x2c], eax
0x1624f:      test    r13d, r13d
0x16252:      jne     0x16260
0x16254:      add     rsp, 8
0x16258:      pop     rbx
0x16259:      pop     rbp
0x1625a:      pop     r12
0x1625c:      pop     r13
0x1625e:      ret
0x1625f:      nop
0x16260:      add     rsp, 8
0x16264:      mov     rdi, r12
0x16267:      pop     rbx
0x16268:      pop     rbp
0x16269:      pop     r12
0x1626b:      pop     r13
0x1626d:      jmp     0x18d10
0x18d10
0x16272:      nop     word ptr [rax + rax]
0x16278:      mov     rdi, rbp
0x1627b:      xor     eax, eax
0x1627d:      call    0x14d00
0x16282:      jmp     0x1624f
0x1624f
0x16284:      nop     dword ptr [rax]
0x16288:      mov     rsi, r12
0x1628b:      mov     rdi, rbp
0x1628e:      xor     eax, eax
0x16290:      call    0x122d0

```

```

0x16295:      mov     rdi, rbp
0x16298:      mov     dword ptr [rbp + 0x28], eax
0x1629b:      xor     eax, eax
0x1629d:      call    0xf110
0x162a2:      mov     dword ptr [rbp + 0x2c], eax
0x162a5:      jmp     0x1624f
0x1624f:
0x162a7:      nop     word ptr [rax + rax]
0x162b0:      endbr64
0x162b4:      push    r15
0x162b6:      push    r14
0x162b8:      push    r13
0x162ba:      push    r12
0x162bc:      mov     r12, rdi
0x162bf:      push    rbp
0x162c0:      movsxd  rax, dword ptr [rdi + 0x28]
0x162c4:      mov     rbp, rsi
0x162c7:      lea     rax, [rax + rax*4]
0x162cb:      shl     rax, 4
0x162cf:      add     rax, qword ptr [rdi + 0x38]
0x162d3:      movzx   edx, byte ptr [rax + 0x48]
0x162d7:      mov     r14, qword ptr [rax]
0x162da:      test    dl, dl
0x162dc:      je      0x16300
0x162de:      cmp     dl, 1
0x162e1:      jne     0x162ed
0x162e3:      cmp     byte ptr [rax + 0x4a], 0
0x162e7:      jne     0x16420
0x162ed:      xor     r13d, r13d
0x162f0:      pop     rbp
0x162f1:      mov     eax, r13d
0x162f4:      pop     r12
0x162f6:      pop     r13
0x162f8:      pop     r14
0x162fa:      pop     r15
0x162fc:      ret
0x162fd:      nop     dword ptr [rax]
0x16300:      cmp     byte ptr [r14], 0x2e
0x16304:      jne     0x163c0
0x1630a:      cmp     byte ptr [r14 + 1], 0x2e
0x1630f:      jne     0x163c0
0x16315:      cmp     byte ptr [r14 + 2], 0
0x1631a:      jne     0x163c0
0x16320:      mov     rax, qword ptr [rdi + 0x18]
0x16324:      cmp     byte ptr [rax], 0x2f
0x16327:      jne     0x16330
0x16329:      cmp     byte ptr [rax + 1], 0
0x1632d:      je      0x162ed

```

```

0x1632f:      nop
0x16330:      mov     rax, qword ptr [r12 + 0x78]
0x16335:      lea     rdi, [rax + 1]
0x16339:      call    0x1d670
0x1633e:      mov     rsi, qword ptr [r12 + 0x18]
0x16343:      mov     rdi, rax
0x16346:      mov     r15, rax
0x16349:      call    0x48b0
0x1634e:      mov     edx, 1
0x16353:      mov     rsi, r14
0x16356:      mov     rdi, r12
0x16359:      xor     eax, eax
0x1635b:      call    0xf880
0x16360:      mov     r13d, eax
0x16363:      test    eax, eax
0x16365:      jne     0x165c0
0x1636b:      xor     eax, eax
0x1636d:      mov     rdi, r12
0x16370:      call    0x14d00
0x16375:      xor     eax, eax
0x16377:      mov     rdi, r12
0x1637a:      call    0x10ec0
0x1637f:      xor     eax, eax
0x16381:      mov     rdi, r12
0x16384:      call    0x10160
0x16389:      cmp     byte ptr [r12 + 0x6d], 0
0x1638f:      jne     0x165b0
0x16395:      xor     eax, eax
0x16397:      mov     rdi, r12
0x1639a:      call    0x10240
0x1639f:      cmp     byte ptr [r12 + 0x6d], 0
0x163a5:      je      0x162ed
0x163ab:      mov     rdi, r12
0x163ae:      xor     eax, eax
0x163b0:      call    0x10d70
0x163b5:      jmp     0x162f0
0x162f0:
0x163ba:      nop     word ptr [rax + rax]
0x163c0:      mov     rax, qword ptr [r12 + 0x78]
0x163c5:      lea     rdi, [rax + 1]
0x163c9:      call    0x1d670
0x163ce:      mov     rsi, qword ptr [r12 + 0x18]
0x163d3:      mov     rdi, rax
0x163d6:      mov     r15, rax
0x163d9:      call    0x48b0
0x163de:      mov     edx, 1
0x163e3:      mov     rsi, r14
0x163e6:      mov     rdi, r12

```

```

0x163e9:      xor     eax, eax
0x163eb:      call    0xf880
0x163f0:      mov     r13d, eax
0x163f3:      test    eax, eax
0x163f5:      jne     0x164b0
0x163fb:      mov     ecx, 0x1c
0x16400:      xor     edx, edx
0x16402:      mov     rsi, r14
0x16405:      xor     eax, eax
0x16407:      lea     rdi, [rip + 0xaf94]
0x1640e:      call    0xf760
0x16413:      jmp     0x162f0
0x162f0
0x16418:      nop     dword ptr [rax + rax]
0x16420:      mov     rdi, r14
0x16423:      call    0x49b0
0x16428:      lea     rdi, [rax + 0x21]
0x1642c:      call    0x1d670
0x16431:      mov     r8, r14
0x16434:      mov     esi, 1
0x16439:      lea     rcx, [rip + 0xb04b]
0x16440:      mov     rdx, -1
0x16447:      mov     r13, rax
0x1644a:      mov     rdi, rax
0x1644d:      xor     eax, eax
0x1644f:      call    0x4f10
0x16454:      xor     esi, esi
0x16456:      mov     rdi, r13
0x16459:      call    0x193f0
0x1645e:      mov     rdi, r13
0x16461:      mov     r13d, 1
0x16467:      mov     dword ptr [rip + 0x1446b], 1
0x16471:      call    0x18d10
0x16476:      xor     eax, eax
0x16478:      call    0xa480
0x1647d:      mov     rdi, qword ptr [rip + 0x14444]
0x16484:      call    0xbb10
0x16489:      mov     rdi, r14
0x1648c:      mov     dword ptr [r12 + 0x14], 0xffffffff
0x16495:      mov     dword ptr [rbp + 0x14], 0xffffffff
0x1649c:      call    0xd610
0x164a1:      mov     rdi, qword ptr [r12 + 0x18]
0x164a6:      call    0xc4b0
0x164ab:      jmp     0x162f0
0x162f0
0x164b0:      mov     dword ptr [r12 + 0x28], 0
0x164b9:      mov     rdi, r12
0x164bc:      xor     eax, eax

```

```

0x164be:      call    0xf110
0x164c3:      mov     dword ptr [r12 + 0x2c], eax
0x164c8:      mov     rdi, r15
0x164cb:      call    0x18d10
0x164d0:      xor     eax, eax
0x164d2:      mov     rdi, r12
0x164d5:      call    0x10ec0
0x164da:      xor     eax, eax
0x164dc:      mov     rdi, r12
0x164df:      call    0x10160
0x164e4:      cmp     byte ptr [r12 + 0x6d], 0
0x164ea:      je      0x164f6
0x164ec:      mov     rdi, r12
0x164ef:      xor     eax, eax
0x164f1:      call    0xec50
0x164f6:      xor     eax, eax
0x164f8:      mov     rdi, r12
0x164fb:      call    0x10240
0x16500:      cmp     byte ptr [r12 + 0x6d], 0
0x16506:      je      0x16512
0x16508:      mov     rdi, r12
0x1650b:      xor     eax, eax
0x1650d:      call    0x10d70
0x16512:      mov     rsi, qword ptr [rbp + 0x18]
0x16516:      mov     rdi, qword ptr [r12 + 0x18]
0x1651b:      call    0x4b30
0x16520:      mov     r13d, eax
0x16523:      test    eax, eax
0x16525:      jne     0x16588
0x16527:      mov     r8d, 1
0x1652d:      mov     rdx, r12
0x16530:      mov     esi, 0x22
0x16535:      mov     rdi, rbp
0x16538:      mov     rcx, -1
0x1653f:      call    0x13e30
0x16544:      xor     eax, eax
0x16546:      mov     rdi, rbp
0x16549:      call    0x10ec0
0x1654e:      xor     eax, eax
0x16550:      mov     rdi, rbp
0x16553:      call    0x10160
0x16558:      cmp     byte ptr [rbp + 0x6d], 0
0x1655c:      jne     0x165a0
0x1655e:      xor     eax, eax
0x16560:      mov     rdi, rbp
0x16563:      call    0x10240
0x16568:      cmp     byte ptr [rbp + 0x6d], 0
0x1656c:      je      0x162ed

```

```

0x16572:      mov     rdi, rbp
0x16575:      xor     eax, eax
0x16577:      call    0x10d70
0x1657c:      jmp     0x162f0
0x162f0
0x16581:      nop     dword ptr [rax]
0x16588:      mov     rdi, rbp
0x1658b:      xor     eax, eax
0x1658d:      call    0x10240
0x16592:      jmp     0x162ed
0x162ed
0x16597:      nop     word ptr [rax + rax]
0x165a0:      mov     rdi, rbp
0x165a3:      xor     eax, eax
0x165a5:      call    0xec50
0x165aa:      jmp     0x1655e
0x1655e
0x165ac:      nop     dword ptr [rax]
0x165b0:      mov     rdi, r12
0x165b3:      xor     eax, eax
0x165b5:      call    0xec50
0x165ba:      jmp     0x16395
0x16395
0x165bf:      nop
0x165c0:      mov     rdi, r15
0x165c3:      mov     esi, 0x2f
0x165c8:      call    0x4a20
0x165cd:      mov     rdi, r12
0x165d0:      lea     rsi, [rax + 1]
0x165d4:      xor     eax, eax
0x165d6:      call    0x122d0
0x165db:      mov     rdi, r12
0x165de:      mov     dword ptr [r12 + 0x28], eax
0x165e3:      xor     eax, eax
0x165e5:      call    0xf110
0x165ea:      jmp     0x164c3
0x164c3
0x165ef:      nop
0x165f0:      endbr64
0x165f4:      push    r13
0x165f6:      push    r12
0x165f8:      mov     r12, rdx
0x165fb:      push    rbp
0x165fc:      mov     rbp, rsi
0x165ff:      push    rbx
0x16600:      mov     rbx, rdi
0x16603:      mov     rdi, rdx
0x16606:      sub     rsp, 8

```



```

0x1660a:    call    0x49b0
0x1660f:    cmp     byte ptr [r12], 0x2f
0x16614:    je      0x166f0
0x1661a:    add     rax, qword ptr [rbx + 0x78]
0x1661e:    mov     rdi, qword ptr [rbx + 0x18]
0x16622:    lea     rdx, [rax + 1]
0x16626:    lea     rsi, [rax + 2]
0x1662a:    mov     qword ptr [rbx + 0x78], rdx
0x1662e:    call    0x1d690
0x16633:    mov     qword ptr [rbx + 0x18], rax
0x16637:    mov     rdi, rax
0x1663a:    mov     r13, rax
0x1663d:    call    0x49b0
0x16642:    mov     edx, 0x2f
0x16647:    mov     rsi, r12
0x1664a:    mov     word ptr [r13 + rax], dx
0x16650:    mov     rdi, qword ptr [rbx + 0x18]
0x16654:    call    0x4df0
0x16659:    mov     rdi, qword ptr [rbx + 0x18]
0x1665d:    call    0x18040
0x16662:    mov     dword ptr [rbx + 0x28], 0
0x16669:    mov     rdi, rbx
0x1666c:    xor     eax, eax
0x1666e:    mov     r8d, 1
0x16674:    call    0xf110
0x16679:    xor     ecx, ecx
0x1667b:    xor     edx, edx
0x1667d:    mov     esi, 0x22
0x16682:    mov     dword ptr [rbx + 0x2c], eax
0x16685:    mov     rdi, rbx
0x16688:    call    0x13e30
0x1668d:    xor     eax, eax
0x1668f:    mov     rdi, rbx
0x16692:    call    0x10ec0
0x16697:    xor     eax, eax
0x16699:    mov     rdi, rbx
0x1669c:    call    0x10160
0x166a1:    cmp     byte ptr [rbx + 0x6d], 0
0x166a5:    je      0x166b1
0x166a7:    mov     rdi, rbx
0x166aa:    xor     eax, eax
0x166ac:    call    0xec50
0x166b1:    xor     eax, eax
0x166b3:    mov     rdi, rbx
0x166b6:    call    0x10240
0x166bb:    cmp     byte ptr [rbx + 0x6d], 0
0x166bf:    je      0x166cb
0x166c1:    mov     rdi, rbx

```

```

0x166c4:      xor     eax, eax
0x166c6:      call    0x10d70
0x166cb:      mov     rsi, qword ptr [rbp + 0x18]
0x166cf:      mov     rdi, qword ptr [rbx + 0x18]
0x166d3:      call    0x4b30
0x166d8:      test    eax, eax
0x166da:      je       0x16718
0x166dc:      add     rsp, 8
0x166e0:      pop     rbx
0x166e1:      pop     rbp
0x166e2:      pop     r12
0x166e4:      pop     r13
0x166e6:      ret
0x166e7:      nop     word ptr [rax + rax]
0x166f0:      mov     qword ptr [rbx + 0x78], rax
0x166f4:      mov     rdi, qword ptr [rbx + 0x18]
0x166f8:      lea     rsi, [rax + 1]
0x166fc:      call    0x1d690
0x16701:      mov     rsi, r12
0x16704:      mov     qword ptr [rbx + 0x18], rax
0x16708:      mov     rdi, rax
0x1670b:      call    0x48b0
0x16710:      jmp     0x16662
0x16662
0x16715:      nop     dword ptr [rax]
0x16718:      xor     edx, edx
0x1671a:      mov     r8d, 1
0x16720:      mov     esi, 0x22
0x16725:      mov     rdi, rbp
0x16728:      mov     rcx, -1
0x1672f:      call    0x13e30
0x16734:      xor     eax, eax
0x16736:      mov     rdi, rbp
0x16739:      call    0x10ec0
0x1673e:      xor     eax, eax
0x16740:      mov     rdi, rbp
0x16743:      call    0x10160
0x16748:      cmp     byte ptr [rbp + 0x6d], 0
0x1674c:      jne     0x16780
0x1674e:      xor     eax, eax
0x16750:      mov     rdi, rbp
0x16753:      call    0x10240
0x16758:      cmp     byte ptr [rbp + 0x6d], 0
0x1675c:      je       0x166dc
0x16762:      add     rsp, 8
0x16766:      mov     rdi, rbp
0x16769:      xor     eax, eax
0x1676b:      pop     rbx

```

```

0x1676c:      pop     rbp
0x1676d:      pop     r12
0x1676f:      pop     r13
0x16771:      jmp     0x10d70
0x10d70
0x16776:      nop     word ptr cs:[rax + rax]
0x16780:      xor     eax, eax
0x16782:      mov     rdi, rbp
0x16785:      call    0xec50
0x1678a:      xor     eax, eax
0x1678c:      mov     rdi, rbp
0x1678f:      call    0x10240
0x16794:      cmp     byte ptr [rbp + 0x6d], 0
0x16798:      je      0x166dc
0x1679e:      jmp     0x16762
0x16762
0x167a0:      endbr64
0x167a4:      push    r15
0x167a6:      lea     r15, [rip + 0x13bd3]
0x167ad:      push    r14
0x167af:      lea     r14, [rip + 0xacdc]
0x167b6:      push    r13
0x167b8:      mov     r13, rsi
0x167bb:      push    r12
0x167bd:      mov     r12d, edx
0x167c0:      push    rbp
0x167c1:      mov     rbp, rdi
0x167c4:      push    rbx
0x167c5:      sub     rsp, 8
0x167c9:      mov     rdi, qword ptr [rip + 0x13d08]
0x167d0:      call    0xdb60
0x167d5:      xor     eax, eax
0x167d7:      call    0xa820
0x167dc:      mov     eax, dword ptr [rbp + 0x70]
0x167df:      mov     dword ptr [rbp + 0x50], 0xffffffff
0x167e6:      mov     dword ptr [rbp + 0x90], 1
0x167f0:      mov     dword ptr [rbp + 0x54], eax
0x167f3:      nop     dword ptr [rax + rax]
0x167f8:      mov     rdi, rbp
0x167fb:      call    0x100e0
0x16800:      movsxd  rbx, eax
0x16803:      cmp     ebx, -1
0x16806:      je      0x1686b
0x16808:      xor     eax, eax
0x1680a:      call    0x18ab0
0x1680f:      xor     eax, eax
0x16811:      mov     edx, r12d
0x16814:      mov     esi, ebx

```

```

0x16816:    mov     rdi, rbp
0x16819:    call   0x134d0
0x1681e:    cmp     eax, 1
0x16821:    je      0x168f0
0x16827:    cmp     eax, 2
0x1682a:    je      0x1686b
0x1682c:    cmp     eax, 3
0x1682f:    je      0x167f8
0x16831:    sub     eax, 1
0x16834:    xor     ecx, ecx
0x16836:    mov     rdi, r14
0x16839:    mov     r8d, 0x1e
0x1683f:    cdqe
0x16841:    mov     rdx, qword ptr [r15 + rax*8]
0x16845:    lea     rax, [rbx + rbx*4]
0x16849:    shl     rax, 4
0x1684d:    add     rax, qword ptr [rbp + 0x38]
0x16851:    mov     rsi, qword ptr [rax]
0x16854:    xor     eax, eax
0x16856:    call   0xf7e0
0x1685b:    mov     rdi, rbp
0x1685e:    call   0x100e0
0x16863:    movsxd  rbx, eax
0x16866:    cmp     ebx, -1
0x16869:    jne     0x16808
0x1686b:    xor     eax, eax
0x1686d:    call   0xe4e0
0x16872:    xor     eax, eax
0x16874:    call   0xa820
0x16879:    xor     ecx, ecx
0x1687b:    xor     edx, edx
0x1687d:    mov     r8d, 1
0x16883:    mov     esi, 0x22
0x16888:    mov     rdi, rbp
0x1688b:    call   0x13e30
0x16890:    xor     eax, eax
0x16892:    mov     rdi, rbp
0x16895:    call   0x10ec0
0x1689a:    xor     eax, eax
0x1689c:    mov     rdi, rbp
0x1689f:    call   0x10160
0x168a4:    cmp     byte ptr [rbp + 0x6d], 0
0x168a8:    je      0x168b4
0x168aa:    mov     rdi, rbp
0x168ad:    xor     eax, eax
0x168af:    call   0xec50
0x168b4:    xor     eax, eax
0x168b6:    mov     rdi, rbp

```

```

0x168b9:      call    0x10240
0x168be:      cmp     byte ptr [rbp + 0x6d], 0
0x168c2:      je      0x168ce
0x168c4:      mov     rdi, rbp
0x168c7:      xor     eax, eax
0x168c9:      call    0x10d70
0x168ce:      mov     rsi, qword ptr [r13 + 0x18]
0x168d2:      mov     rdi, qword ptr [rbp + 0x18]
0x168d6:      call    0x4b30
0x168db:      test    eax, eax
0x168dd:      je      0x16908
0x168df:      add     rsp, 8
0x168e3:      pop     rbx
0x168e4:      pop     rbp
0x168e5:      pop     r12
0x168e7:      pop     r13
0x168e9:      pop     r14
0x168eb:      pop     r15
0x168ed:      ret
0x168ee:      nop
0x168f0:      lea     rax, [rbx + rbx*4]
0x168f4:      shl     rax, 4
0x168f8:      add     rax, qword ptr [rbp + 0x38]
0x168fc:      mov     byte ptr [rax + 0x49], 0
0x16900:      jmp     0x167f8
0x167f8:
0x16905:      nop     dword ptr [rax]
0x16908:      xor     ecx, ecx
0x1690a:      xor     edx, edx
0x1690c:      mov     r8d, 1
0x16912:      mov     esi, 0x22
0x16917:      mov     rdi, r13
0x1691a:      call    0x13e30
0x1691f:      xor     eax, eax
0x16921:      mov     rdi, r13
0x16924:      call    0x10ec0
0x16929:      xor     eax, eax
0x1692b:      mov     rdi, r13
0x1692e:      call    0x10160
0x16933:      cmp     byte ptr [r13 + 0x6d], 0
0x16938:      jne     0x16968
0x1693a:      xor     eax, eax
0x1693c:      mov     rdi, r13
0x1693f:      call    0x10240
0x16944:      cmp     byte ptr [r13 + 0x6d], 0
0x16949:      je      0x168df
0x1694b:      add     rsp, 8
0x1694f:      mov     rdi, r13

```

```

0x16952:    xor     eax, eax
0x16954:    pop     rbx
0x16955:    pop     rbp
0x16956:    pop     r12
0x16958:    pop     r13
0x1695a:    pop     r14
0x1695c:    pop     r15
0x1695e:    jmp     0x10d70
0x10d70
0x16963:    nop     dword ptr [rax + rax]
0x16968:    mov     rdi, r13
0x1696b:    xor     eax, eax
0x1696d:    call    0xec50
0x16972:    jmp     0x1693a
0x1693a
0x16974:    nop     word ptr cs:[rax + rax]
0x1697e:    nop
0x16980:    push    r15
0x16982:    push    r14
0x16984:    push    r13
0x16986:    push    r12
0x16988:    push    rbp
0x16989:    push    rbx
0x1698a:    sub     rsp, 0x8c8
0x16991:    mov     rdi, qword ptr [rip + 0x14ca8]
0x16998:    mov     r13, qword ptr [rip + 0x13cc1]
0x1699f:    mov     rax, qword ptr fs:[0x28]
0x169a8:    mov     qword ptr [rsp + 0x8b8], rax
0x169b0:    xor     eax, eax
0x169b2:    lea     rax, [rip + 0x14c8f]
0x169b9:    mov     qword ptr [rip + 0x13ca0], rax
0x169c0:    test    rdi, rdi
0x169c3:    je      0x16a78
0x169c9:    lea     rsi, [rsp + 0x20]
0x169ce:    call    0x18f40
0x169d3:    cmp     eax, -1
0x169d6:    je      0x16a78
0x169dc:    mov     rax, qword ptr [rsp + 0x68]
0x169e1:    mov     rdi, qword ptr [rip + 0x14c58]
0x169e8:    lea     rsi, [rip + 0xb5da]
0x169ef:    mov     qword ptr [rsp + 0x10], rax
0x169f4:    mov     rax, qword ptr [rsp + 0x78]
0x169f9:    mov     qword ptr [rsp + 0x18], rax
0x169fe:    call    0x4da0
0x16a03:    mov     r14, rax
0x16a06:    test    rax, rax
0x16a09:    je      0x16a78
0x16a0b:    mov     dword ptr [rsp + 4], 0

```

```

0x16a13:    xor     ebp, ebp
0x16a15:    xor     ebx, ebx
0x16a17:    xor     r12d, r12d
0x16a1a:    lea     r15, [rsp + 0xb0]
0x16a22:    nop     word ptr [rax + rax]
0x16a28:    mov     rdx, r14
0x16a2b:    mov     esi, 0x800
0x16a30:    mov     rdi, r15
0x16a33:    call    0x4b00
0x16a38:    test    rax, rax
0x16a3b:    je      0x16ad0
0x16a41:    cmp     byte ptr [rsp + 0xb0], 0
0x16a49:    je      0x16aa8
0x16a4b:    cmp     dword ptr [r15], 0x6d6f7246
0x16a52:    je      0x16ab0
0x16a54:    test    ebp, ebp
0x16a56:    jne     0x16a28
0x16a58:    test    ebx, ebx
0x16a5a:    je      0x16a28
0x16a5c:    cmp     dword ptr [r15], 0x74617453
0x16a63:    je      0x16b60
0x16a69:    xor     ebp, ebp
0x16a6b:    mov     ebx, 1
0x16a70:    jmp     0x16a28
0x16a28:
0x16a72:    nop     word ptr [rax + rax]
0x16a78:    xor     eax, eax
0x16a7a:    mov     rdx, qword ptr [rsp + 0x8b8]
0x16a82:    sub     rdx, qword ptr fs:[0x28]
0x16a8b:    jne     0x16c02
0x16a91:    add     rsp, 0x8c8
0x16a98:    pop     rbx
0x16a99:    pop     rbp
0x16a9a:    pop     r12
0x16a9c:    pop     r13
0x16a9e:    pop     r14
0x16aa0:    pop     r15
0x16aa2:    ret
0x16aa3:    nop     dword ptr [rax + rax]
0x16aa8:    xor     ebx, ebx
0x16aaa:    jmp     0x16a28
0x16a28:
0x16aaf:    nop
0x16ab0:    cmp     byte ptr [r15 + 4], 0x20
0x16ab5:    jne     0x16a54
0x16ab7:    add     r12d, 1
0x16abb:    xor     ebp, ebp
0x16abd:    mov     ebx, 1

```

```

0x16ac2:      jmp      0x16a28
0x16a28
0x16ac7:      nop      word ptr [rax + rax]
0x16ad0:      mov      rdi, r14
0x16ad3:      call     0x4940
0x16ad8:      mov      rdi, qword ptr [rip + 0x14b61]
0x16adf:      lea      rsi, [rsp + 0x10]
0x16ae4:      call     0x47f0
0x16ae9:      mov      rbp, qword ptr [rip + 0x13b70]
0x16af0:      test     r12d, r12d
0x16af3:      je       0x16b0a
0x16af5:      cmp      r12d, dword ptr [rsp + 4]
0x16afa:      je       0x16b48
0x16afc:      lea      rbp, [rip + 0x13b1d]
0x16b03:      mov      qword ptr [rip + 0x13b56], rbp
0x16b0a:      mov      rdi, rbp
0x16b0d:      call     0x49b0
0x16b12:      mov      rsi, r13
0x16b15:      mov      rdi, rbp
0x16b18:      mov      r8, rax
0x16b1b:      mov      eax, dword ptr [rip + 0x14b2f]
0x16b21:      add      eax, dword ptr [rip + 0x14b2d]
0x16b27:      lea      eax, [rax + r8 + 0x15]
0x16b2c:      mov      dword ptr [rip + 0x14b1a], eax
0x16b32:      call     0x4b30
0x16b37:      test     eax, eax
0x16b39:      setne    al
0x16b3c:      movzx    eax, al
0x16b3f:      jmp      0x16a7a
0x16a7a
0x16b44:      nop      dword ptr [rax]
0x16b48:      lea      rbp, [rip + 0x13adc]
0x16b4f:      mov      qword ptr [rip + 0x13b0a], rbp
0x16b56:      jmp      0x16b0a
0x16b0a
0x16b58:      nop      dword ptr [rax + rax]
0x16b60:      cmp      word ptr [r15 + 4], 0x7375
0x16b67:      jne      0x16a69
0x16b6d:      cmp      byte ptr [r15 + 6], 0x3a
0x16b72:      jne      0x16a69
0x16b78:      mov      rdi, r15
0x16b7b:      mov      esi, 0x3a
0x16b80:      call     0x4a00
0x16b85:      movsx    rbx, byte ptr [rax + 1]
0x16b8a:      lea      rdi, [rax + 1]
0x16b8e:      test     bl, bl
0x16b90:      je       0x16bf8
0x16b92:      mov      qword ptr [rsp + 8], rdi

```



```

0x16b97:      call    0x4f00
0x16b9c:      mov     rdi, qword ptr [rsp + 8]
0x16ba1:      mov     rax, qword ptr [rax]
0x16ba4:      jmp     0x16bbd
0x16bbd
0x16ba6:      nop     word ptr cs:[rax + rax]
0x16bb0:      movsx   rbx, byte ptr [rdi + 1]
0x16bb5:      add     rdi, 1
0x16bb9:      test    bl, bl
0x16bbb:      je      0x16bf8
0x16bbd:      test    byte ptr [rax + rbx*2 + 1], 0x20
0x16bc2:      jne     0x16bb0
0x16bc4:      mov     esi, 0x52
0x16bc9:      mov     ebp, 1
0x16bce:      mov     ebx, 1
0x16bd3:      call    0x4a00
0x16bd8:      test    rax, rax
0x16bdb:      je      0x16a28
0x16be1:      add     dword ptr [rsp + 4], 1
0x16be6:      mov     ebp, 1
0x16beb:      mov     ebx, 1
0x16bf0:      jmp     0x16a28
0x16a28
0x16bf5:      nop     dword ptr [rax]
0x16bf8:      mov     ebx, 1
0x16bfd:      jmp     0x16a28
0x16a28
0x16c02:      call    0x49d0
0x16c07:      nop     word ptr [rax + rax]
0x16c10:      endbr64
0x16c14:      sub     rsp, 0xa8
0x16c1b:      mov     edi, 0x1d
0x16c20:      mov     rax, qword ptr fs:[0x28]
0x16c29:      mov     qword ptr [rsp + 0x98], rax
0x16c31:      xor     eax, eax
0x16c33:      call    0x1d670
0x16c38:      lea     r9, [rip + 0x9950]
0x16c3f:      lea     r8, [rip + 0x991f]
0x16c46:      mov     rdx, -1
0x16c4d:      lea     rcx, [rip + 0xb219]
0x16c54:      mov     esi, 1
0x16c59:      mov     rdi, rax
0x16c5c:      mov     qword ptr [rip + 0x149fd], rax
0x16c63:      xor     eax, eax
0x16c65:      call    0x4f10
0x16c6a:      mov     rdi, qword ptr [rip + 0x149ef]
0x16c71:      call    0x49b0
0x16c76:      mov     rdi, qword ptr [rip + 0x14e1b]

```

```

0x16c7d:      mov     dword ptr [rip + 0x149d5], eax
0x16c83:      call   0x49b0
0x16c88:      mov     rdi, qword ptr [rip + 0x13d01]
0x16c8f:      mov     dword ptr [rip + 0x149bf], eax
0x16c95:      call   0x49b0
0x16c9a:      lea     rdi, [rip + 0x9999]
0x16ca1:      mov     dword ptr [rip + 0x149a9], eax
0x16ca7:      call   0x47e0
0x16cac:      mov     qword ptr [rip + 0x1498d], rax
0x16cb3:      test    rax, rax
0x16cb6:      je      0x16cc3
0x16cb8:      mov     rdi, rax
0x16cbb:      mov     rsi, rsp
0x16cbe:      call   0x18f40
0x16cc3:      xor     eax, eax
0x16cc5:      call   0x16980
0x16cca:      mov     rdi, qword ptr [rip + 0x1398f]
0x16cd1:      call   0x49b0
0x16cd6:      mov     r8, rax
0x16cd9:      mov     eax, dword ptr [rip + 0x14971]
0x16cdf:      add     eax, dword ptr [rip + 0x1496f]
0x16ce5:      lea     eax, [rax + r8 + 0x15]
0x16cea:      mov     dword ptr [rip + 0x1495c], eax
0x16cf0:      xor     eax, eax
0x16cf2:      call   0xc590
0x16cf7:      mov     qword ptr [rip + 0x1493a], rax
0x16cfe:      mov     rax, qword ptr [rsp + 0x98]
0x16d06:      sub     rax, qword ptr fs:[0x28]
0x16d0f:      jne     0x16d19
0x16d11:      add     rsp, 0xa8
0x16d18:      ret
0x16d19:      call   0x49d0
0x16d1e:      nop
0x16d20:      endbr64
0x16d24:      mov     rdi, qword ptr [rip + 0x1490d]
0x16d2b:      jmp     0xc5c0
0xc5c0
0x16d30:      endbr64
0x16d34:      mov     r8d, edi
0x16d37:      mov     rdi, qword ptr [rip + 0x148fa]
0x16d3e:      mov     edx, esi
0x16d40:      mov     ecx, 1
0x16d45:      xor     esi, esi
0x16d47:      jmp     0xc5e0
0xc5e0
0x16d4c:      nop     dword ptr [rax]
0x16d50:      endbr64
0x16d54:      push    r14

```

```

0x16d56:    push    r13
0x16d58:    push    r12
0x16d5a:    push    rbp
0x16d5b:    push    rbx
0x16d5c:    sub     rsp, 0x10
0x16d60:    mov     rax, qword ptr fs:[0x28]
0x16d69:    mov     qword ptr [rsp + 8], rax
0x16d6e:    xor     eax, eax
0x16d70:    lea     rbp, [rsp + 7]
0x16d75:    mov     rdi, rbp
0x16d78:    call    0xa6a0
0x16d7d:    mov     edx, dword ptr [rip + 0x138c1]
0x16d83:    mov     esi, dword ptr [rip + 0x138b7]
0x16d89:    mov     edi, dword ptr [rip + 0x138b9]
0x16d8f:    call    0xa570
0x16d94:    mov     rdi, qword ptr [rip + 0x1489d]
0x16d9b:    xor     edx, edx
0x16d9d:    xor     esi, esi
0x16d9f:    call    0xc680
0x16da4:    mov     edx, dword ptr [rip + 0x148ae]
0x16daa:    mov     rsi, qword ptr [rip + 0x148af]
0x16db1:    mov     rdi, qword ptr [rip + 0x14880]
0x16db8:    call    0xc600
0x16dbd:    mov     rax, qword ptr [rip + 0x14874]
0x16dc4:    mov     edi, dword ptr [rax + 0xc]
0x16dc7:    add     edi, 1
0x16dca:    movsxd  rdi, edi
0x16dcd:    call    0x1d670
0x16dd2:    mov     r13, rax
0x16dd5:    xor     eax, eax
0x16dd7:    call    0x16980
0x16ddc:    mov     r14, qword ptr [rip + 0x14855]
0x16de3:    mov     eax, dword ptr [rip + 0x1486f]
0x16de9:    mov     ecx, dword ptr [rip + 0x1485d]
0x16def:    mov     edx, dword ptr [r14 + 0xc]
0x16df3:    lea     esi, [rax + rcx + 2]
0x16df7:    cmp     esi, edx
0x16df9:    jl      0x16e58
0x16dfb:    cmp     eax, edx
0x16dfd:    jl      0x16e30
0x16dff:    mov     rdi, r13
0x16e02:    call    0x18d10
0x16e07:    mov     rdi, rbp
0x16e0a:    call    0xa6b0
0x16e0f:    mov     rax, qword ptr [rsp + 8]
0x16e14:    sub     rax, qword ptr fs:[0x28]
0x16e1d:    jne     0x16fc7
0x16e23:    add     rsp, 0x10

```

```

0x16e27:    pop     rbx
0x16e28:    pop     rbp
0x16e29:    pop     r12
0x16e2b:    pop     r13
0x16e2d:    pop     r14
0x16e2f:    ret
0x16e30:    sub     edx, eax
0x16e32:    mov     rdi, r13
0x16e35:    mov     esi, 0x20
0x16e3a:    mov     r12d, edx
0x16e3d:    movsxd  rdx, edx
0x16e40:    call    0x4a50
0x16e45:    mov     edx, r12d
0x16e48:    mov     rsi, r13
0x16e4b:    mov     rdi, r14
0x16e4e:    call    0xc600
0x16e53:    jmp     0x16dff
0x16dff:
0x16e55:    nop     dword ptr [rax]
0x16e58:    sub     edx, eax
0x16e5a:    mov     rdi, r13
0x16e5d:    mov     esi, 0x20
0x16e62:    mov     ebx, edx
0x16e64:    sub     ebx, ecx
0x16e66:    movsxd  rdx, ebx
0x16e69:    call    0x4a50
0x16e6e:    mov     edx, ebx
0x16e70:    mov     rsi, r13
0x16e73:    mov     rdi, r14
0x16e76:    call    0xc600
0x16e7b:    add     ebx, dword ptr [rip + 0x147d7]
0x16e81:    mov     rdi, qword ptr [rip + 0x147b0]
0x16e88:    xor     esi, esi
0x16e8a:    mov     edx, ebx
0x16e8c:    call    0xc680
0x16e91:    mov     rdi, qword ptr [rip + 0x147a0]
0x16e98:    mov     edx, 5
0x16e9d:    lea     rsi, [rip + 0x13793]
0x16ea4:    call    0xc600
0x16ea9:    mov     rdi, qword ptr [rip + 0x14788]
0x16eb0:    mov     esi, 0x20
0x16eb5:    call    0xc650
0x16eba:    mov     edi, dword ptr [rip + 0x1378c]
0x16ec0:    call    0xa4f0
0x16ec5:    mov     edx, dword ptr [rip + 0x14789]
0x16ecb:    mov     rsi, qword ptr [rip + 0x14bc6]
0x16ed2:    mov     rdi, qword ptr [rip + 0x1475f]
0x16ed9:    call    0xc600

```

```

0x16ede:    mov     rdi, qword ptr [rip + 0x14753]
0x16ee5:    mov     esi, 0x20
0x16eea:    call    0xc650
0x16eef:    mov     r12, qword ptr [rip + 0x1376a]
0x16ef6:    mov     rdi, r12
0x16ef9:    call    0x49b0
0x16efe:    mov     rdi, qword ptr [rip + 0x14733]
0x16f05:    mov     rsi, r12
0x16f08:    mov     edx, eax
0x16f0a:    call    0xc600
0x16f0f:    mov     rdi, qword ptr [rip + 0x14722]
0x16f16:    mov     esi, 0x20
0x16f1b:    call    0xc650
0x16f20:    mov     edi, dword ptr [rip + 0x1371a]
0x16f26:    call    0xa4f0
0x16f2b:    mov     rdi, qword ptr [rip + 0x14706]
0x16f32:    mov     edx, 4
0x16f37:    lea     rsi, [rip + 0x136f4]
0x16f3e:    call    0xc600
0x16f43:    mov     rdi, qword ptr [rip + 0x146ee]
0x16f4a:    mov     esi, 0x20
0x16f4f:    call    0xc650
0x16f54:    mov     edi, dword ptr [rip + 0x136f6]
0x16f5a:    call    0xa4f0
0x16f5f:    mov     edx, dword ptr [rip + 0x146eb]
0x16f65:    mov     rsi, qword ptr [rip + 0x13a24]
0x16f6c:    mov     rdi, qword ptr [rip + 0x146c5]
0x16f73:    call    0xc600
0x16f78:    mov     edi, dword ptr [rip + 0x136c2]
0x16f7e:    call    0xa4f0
0x16f83:    mov     rdi, qword ptr [rip + 0x146ae]
0x16f8a:    mov     esi, 0x20
0x16f8f:    call    0xc650
0x16f94:    xor     edi, edi
0x16f96:    xor     eax, eax
0x16f98:    call    0x16fd0
0x16f9d:    mov     rdi, qword ptr [rip + 0x14694]
0x16fa4:    xor     esi, esi
0x16fa6:    mov     eax, dword ptr [rdi + 0xc]
0x16fa9:    lea     edx, [rax - 1]
0x16fac:    call    0xc680
0x16fb1:    mov     rdi, qword ptr [rip + 0x14680]
0x16fb8:    mov     esi, 0x20
0x16fbd:    call    0xc650
0x16fc2:    jmp     0x16dff
0x16dff:
0x16fc7:    call    0x49d0
0x16fcc:    nop     dword ptr [rax]

```

```

0x16fd0:      endbr64
0x16fd4:      push    r12
0x16fd6:      push    rbp
0x16fd7:      push    rbx
0x16fd8:      mov     ebx, edi
0x16fda:      sub     rsp, 0x30
0x16fde:      mov     rax, qword ptr fs:[0x28]
0x16fe7:      mov     qword ptr [rsp + 0x28], rax
0x16fec:      xor     eax, eax
0x16fee:      call    0x7870
0x16ff3:      test    eax, eax
0x16ff5:      jne     0x17114
0x16ffb:      mov     eax, dword ptr [rip + 0x1464b]
0x17001:      mov     edx, dword ptr [rip + 0x14651]
0x17007:      lea     edx, [rdx + rax + 2]
0x1700b:      mov     rax, qword ptr [rip + 0x14626]
0x17012:      cmp     edx, dword ptr [rax + 0xc]
0x17015:      jge     0x17114
0x1701b:      test    ebx, ebx
0x1701d:      jne     0x17140
0x17023:      lea     r12, [rsp + 7]
0x17028:      xor     eax, eax
0x1702a:      call    0x18330
0x1702f:      mov     rdi, r12
0x17032:      mov     rbp, rax
0x17035:      call    0xa6a0
0x1703a:      lea     rsi, [rsp + 0xc]
0x1703f:      lea     rdi, [rsp + 8]
0x17044:      call    0xa4d0
0x17049:      xor     edi, edi
0x1704b:      call    0xa660
0x17050:      movsxd  r8, dword ptr [rbp + 8]
0x17054:      mov     r9d, dword ptr [rbp + 4]
0x17058:      lea     rbp, [rsp + 0x10]
0x1705d:      lea     rcx, [rip + 0xa0d4]
0x17064:      mov     esi, 1
0x17069:      mov     rdi, rbp
0x1706c:      mov     rax, r8
0x1706f:      imul    r8, r8, 0x2aaaaaab
0x17076:      cdq
0x17077:      sar     r8, 0x21
0x1707b:      sub     r8d, edx
0x1707e:      lea     edx, [r8 + r8*2]
0x17082:      mov     r8d, eax
0x17085:      shl     edx, 2
0x17088:      sub     r8d, edx
0x1708b:      mov     edx, 0xc
0x17090:      cmov    r8d, edx

```

```

0x17094:      cmp     eax, 0xc
0x17097:      mov     edx, 0x70
0x1709c:      mov     eax, 0x61
0x170a1:      cmovge  eax, edx
0x170a4:      sub     rsp, 8
0x170a8:      mov     edx, 0x10
0x170ad:      push    rax
0x170ae:      xor     eax, eax
0x170b0:      call    0x4f10
0x170b5:      mov     rdi, qword ptr [rip + 0x1457c]
0x170bc:      xor     esi, esi
0x170be:      mov     eax, dword ptr [rdi + 0xc]
0x170c1:      lea     edx, [rax - 7]
0x170c4:      call    0xc680
0x170c9:      mov     edx, dword ptr [rip + 0x13589]
0x170cf:      mov     esi, dword ptr [rip + 0x1357f]
0x170d5:      mov     edi, dword ptr [rip + 0x13581]
0x170db:      call    0xa570
0x170e0:      mov     rdi, rbp
0x170e3:      call    0x49b0
0x170e8:      mov     rdi, qword ptr [rip + 0x14549]
0x170ef:      mov     rsi, rbp
0x170f2:      mov     edx, eax
0x170f4:      call    0xc600
0x170f9:      mov     esi, dword ptr [rsp + 0x1c]
0x170fd:      mov     edi, dword ptr [rsp + 0x18]
0x17101:      call    0xa4b0
0x17106:      mov     rdi, r12
0x17109:      call    0xa6b0
0x1710e:      pop     rax
0x1710f:      pop     rdx
0x17110:      test    ebx, ebx
0x17112:      jne     0x17130
0x17114:      mov     rax, qword ptr [rsp + 0x28]
0x17119:      sub     rax, qword ptr fs:[0x28]
0x17122:      jne     0x1715b
0x17124:      add     rsp, 0x30
0x17128:      pop     rbx
0x17129:      pop     rbp
0x1712a:      pop     r12
0x1712c:      ret
0x1712d:      nop     dword ptr [rax]
0x17130:      xor     eax, eax
0x17132:      call    0xa820
0x17137:      jmp     0x17114
0x17114:
0x17139:      nop     dword ptr [rax]
0x17140:      xor     eax, eax

```

```

0x17142:      call    0x16980
0x17147:      test    eax, eax
0x17149:      je      0x17023
0x1714f:      xor     eax, eax
0x17151:      call    0x16d50
0x17156:      jmp     0x17023
0x17023:
0x1715b:      call    0x49d0
0x17160:      push    rbp
0x17161:      mov     esi, 0x400
0x17166:      push    rbx
0x17167:      lea     rbx, [rip + 0x14512]
0x1716e:      mov     rdi, rbx
0x17171:      sub     rsp, 8
0x17175:      mov     rdx, qword ptr [rip + 0x1490c]
0x1717c:      call    0x4b00
0x17181:      test    rax, rax
0x17184:      je      0x171d8
0x17186:      mov     rdi, rbx
0x17189:      call    0x49b0
0x1718e:      mov     rbp, rax
0x17191:      cmp     rax, 0x3ff
0x17197:      je      0x171e8
0x17199:      mov     esi, 0x23
0x1719e:      mov     rdi, rbx
0x171a1:      call    0x4a00
0x171a6:      test    rax, rax
0x171a9:      je      0x171c0
0x171ab:      mov     byte ptr [rax], 0
0x171ae:      mov     eax, 1
0x171b3:      add     rsp, 8
0x171b7:      pop     rbx
0x171b8:      pop     rbp
0x171b9:      ret
0x171ba:      nop     word ptr [rax + rax]
0x171c0:      cmp     byte ptr [rbx + rbp - 1], 0xa
0x171c5:      mov     eax, 1
0x171ca:      jne     0x171b3
0x171cc:      mov     byte ptr [rbx + rbp - 1], 0
0x171d1:      jmp     0x171b3
0x171b3:
0x171d3:      nop     dword ptr [rax + rax]
0x171d8:      add     rsp, 8
0x171dc:      xor     eax, eax
0x171de:      pop     rbx
0x171df:      pop     rbp
0x171e0:      ret
0x171e1:      nop     dword ptr [rax]

```



```

0x171e8:      mov     rcx, qword ptr [rip + 0x136f9]
0x171ef:      mov     rdi, qword ptr [rip + 0x1368a]
0x171f6:      lea     rdx, [rip + 0xaceb]
0x171fd:      xor     eax, eax
0x171ff:      mov     esi, 1
0x17204:      call    0x4e60
0x17209:      nop     dword ptr [rax]
0x17210:      mov     rdi, qword ptr [rip + 0x14871]
0x17217:      call    0x4a90
0x1721c:      cmp     eax, 0xa
0x1721f:      je      0x17199
0x17225:      cmp     eax, -1
0x17228:      jne     0x17210
0x1722a:      jmp     0x17199
0x17199:
0x1722f:      nop
0x17230:      endbr64
0x17234:      test    rdi, rdi
0x17237:      je      0x17270
0x17239:      sub     rsp, 8
0x1723d:      lea     rsi, [rip + 0xad85]
0x17244:      call    0x4da0
0x17249:      mov     qword ptr [rip + 0x14838], rax
0x17250:      test    rax, rax
0x17253:      je      0x17269
0x17255:      mov     dword ptr [rip + 0x14821], 0xffffffff
0x1725f:      mov     eax, 1
0x17264:      add     rsp, 8
0x17268:      ret
0x17269:      xor     eax, eax
0x1726b:      add     rsp, 8
0x1726f:      ret
0x17270:      xor     eax, eax
0x17272:      ret
0x17273:      nop     word ptr cs:[rax + rax]
0x1727e:      nop
0x17280:      endbr64
0x17284:      mov     rdi, qword ptr [rip + 0x147fd]
0x1728b:      test    rdi, rdi
0x1728e:      je      0x17298
0x17290:      jmp     0x4940
0x4940:
0x17295:      nop     dword ptr [rax]
0x17298:      ret
0x17299:      nop     dword ptr [rax]
0x172a0:      endbr64
0x172a4:      push    rbp
0x172a5:      xor     edx, edx

```

```

0x172a7:    xor     esi, esi
0x172a9:    lea     rbp, [rip + 0x143d0]
0x172b0:    push    rbx
0x172b1:    mov     rbx, rdi
0x172b4:    sub     rsp, 8
0x172b8:    mov     rdi, qword ptr [rip + 0x147c9]
0x172bf:    call    0x4c80
0x172c4:    jmp     0x172df
0x172df:
0x172c6:    nop     word ptr cs:[rax + rax]
0x172d0:    mov     rsi, rbp
0x172d3:    mov     rdi, rbx
0x172d6:    call    0x4b30
0x172db:    test    eax, eax
0x172dd:    je      0x17300
0x172df:    xor     eax, eax
0x172e1:    call    0x17160
0x172e6:    test    eax, eax
0x172e8:    jne     0x172d0
0x172ea:    mov     eax, 0xffffffff
0x172ef:    mov     dword ptr [rip + 0x1478b], eax
0x172f5:    add     rsp, 8
0x172f9:    pop     rbx
0x172fa:    pop     rbp
0x172fb:    ret
0x172fc:    nop     dword ptr [rax]
0x17300:    mov     rdi, qword ptr [rip + 0x14781]
0x17307:    call    0x4b80
0x1730c:    mov     dword ptr [rip + 0x1476e], eax
0x17312:    add     rsp, 8
0x17316:    pop     rbx
0x17317:    pop     rbp
0x17318:    ret
0x17319:    nop     dword ptr [rax]
0x17320:    endbr64
0x17324:    push    r15
0x17326:    push    r14
0x17328:    push    r13
0x1732a:    mov     r13, rdi
0x1732d:    push    r12
0x1732f:    push    rbp
0x17330:    push    rbx
0x17331:    mov     ebx, ecx
0x17333:    sub     rsp, 0x438
0x1733a:    mov     qword ptr [rsp + 0x10], rsi
0x1733f:    mov     dword ptr [rsp + 0xc], edx
0x17343:    mov     rax, qword ptr fs:[0x28]
0x1734c:    mov     qword ptr [rsp + 0x428], rax

```

```

0x17354:    xor     eax, eax
0x17356:    test    ecx, ecx
0x17358:    jne     0x17618
0x1735e:    cmp     dword ptr [rsp + 0xc], 1
0x17363:    jne     0x175e0
0x17369:    mov     rax, qword ptr [rsp + 0x10]
0x1736e:    mov     qword ptr [rax], 0
0x17375:    lea     rbp, [rip + 0x14304]
0x1737c:    lea     r12, [rsp + 0x20]
0x17381:    lea     r14, [rip + 0xa092]
0x17388:    nop     dword ptr [rax + rax]
0x17390:    xor     eax, eax
0x17392:    call    0x17160
0x17397:    test    eax, eax
0x17399:    je      0x17600
0x1739f:    cmp     byte ptr [rbp], 0
0x173a3:    je      0x17600
0x173a9:    mov     rdx, r12
0x173ac:    mov     rsi, r14
0x173af:    mov     rdi, rbp
0x173b2:    xor     eax, eax
0x173b4:    mov     byte ptr [rsp + 0x20], 0
0x173b9:    call    0x4c60
0x173be:    call    0x4f00
0x173c3:    mov     r15, rax
0x173c6:    test    ebx, ebx
0x173c8:    jne     0x17410
0x173ca:    movsx   rcx, byte ptr [rsp + 0x20]
0x173d0:    mov     rax, qword ptr [rax]
0x173d3:    mov     byte ptr [rsp + 0x40], 0
0x173d8:    test    byte ptr [rax + rcx*2 + 1], 0x40
0x173dd:    jne     0x17433
0x173df:    mov     rax, qword ptr [rsp + 0x428]
0x173e7:    sub     rax, qword ptr fs:[0x28]
0x173f0:    jne     0x17657
0x173f6:    add     rsp, 0x438
0x173fd:    pop     rbx
0x173fe:    pop     rbp
0x173ff:    pop     r12
0x17401:    pop     r13
0x17403:    pop     r14
0x17405:    pop     r15
0x17407:    ret
0x17408:    nop     dword ptr [rax + rax]
0x17410:    movsx   rcx, byte ptr [rsp + 0x20]
0x17416:    mov     rax, qword ptr [rax]
0x17419:    test    byte ptr [rax + rcx*2 + 1], 0x40
0x1741e:    je      0x173df

```

```

0x17420:    mov     rsi, r12
0x17423:    mov     rdi, r13
0x17426:    call    0x4b30
0x1742b:    test    eax, eax
0x1742d:    jne     0x17390
0x17433:    mov     esi, 0x3d
0x17438:    mov     rdi, rbp
0x1743b:    call    0x4a00
0x17440:    test    rax, rax
0x17443:    je      0x174e3
0x17449:    cmp     byte ptr [rax + 1], 0
0x1744d:    je      0x174e3
0x17453:    mov     rdx, qword ptr [rsp + 0x10]
0x17458:    lea     rdi, [rax + 1]
0x1745c:    mov     qword ptr [rdx], rdi
0x1745f:    movzx   ecx, byte ptr [rax + 1]
0x17463:    test    cl, cl
0x17465:    je      0x1763f
0x1746b:    cmp     dword ptr [rsp + 0xc], 1
0x17470:    jle     0x1763f
0x17476:    mov     r8d, 1
0x1747c:    jmp     0x1748e
0x1748e:
0x1747e:    nop
0x17480:    mov     ecx, eax
0x17482:    cmp     r8d, dword ptr [rsp + 0xc]
0x17487:    jge     0x174d1
0x17489:    test    sil, sil
0x1748c:    je      0x174d1
0x1748e:    movzx   eax, byte ptr [rdi + 1]
0x17492:    test    al, al
0x17494:    setne   sil
0x17498:    add     rdi, 1
0x1749c:    cmp     cl, 0x3b
0x1749f:    jne     0x17480
0x174a1:    mov     byte ptr [rdi - 1], 0
0x174a5:    cmp     al, 0x3b
0x174a7:    je      0x17638
0x174ad:    test    sil, sil
0x174b0:    je      0x17638
0x174b6:    mov     rdx, qword ptr [rsp + 0x10]
0x174bb:    movsxd  rax, r8d
0x174be:    mov     qword ptr [rdx + rax*8], rdi
0x174c2:    movzx   ecx, byte ptr [rdi]
0x174c5:    test    cl, cl
0x174c7:    setne   sil
0x174cb:    add     r8d, 1
0x174cf:    jmp     0x17482

```

```

0x17482
0x174d1:      mov     esi, 0x3b
0x174d6:      call    0x4a00
0x174db:      test    rax, rax
0x174de:      je       0x174ef
0x174e0:      mov     byte ptr [rax], 0
0x174e3:      mov     eax, dword ptr [rsp + 0xc]
0x174e7:      test    eax, eax
0x174e9:      jle      0x175c0
0x174ef:      mov     rsi, qword ptr [rsp + 0x10]
0x174f4:      xor     r8d, r8d
0x174f7:      jmp     0x17513
0x17513
0x174f9:      nop     dword ptr [rax]
0x17500:      add     r8d, 1
0x17504:      add     rsi, 8
0x17508:      cmp     dword ptr [rsp + 0xc], r8d
0x1750d:      jle      0x175c0
0x17513:      mov     rbp, qword ptr [rsi]
0x17516:      test    rbp, rbp
0x17519:      je       0x17500
0x1751b:      mov     r14, qword ptr [r15]
0x1751e:      movsx   rax, byte ptr [rbp]
0x17523:      test    byte ptr [r14 + rax*2 + 1], 0x20
0x17529:      je       0x17547
0x1752b:      lea     rax, [rbp + 1]
0x1752f:      nop
0x17530:      mov     rbp, rax
0x17533:      mov     qword ptr [rsi], rax
0x17536:      add     rax, 1
0x1753a:      movsx   rcx, byte ptr [rbp]
0x1753f:      test    byte ptr [r14 + rcx*2 + 1], 0x20
0x17545:      jne      0x17530
0x17547:      mov     rdi, rbp
0x1754a:      mov     qword ptr [rsp + 0x18], rsi
0x1754f:      mov     dword ptr [rsp + 0x10], r8d
0x17554:      call    0x49b0
0x17559:      mov     r8d, dword ptr [rsp + 0x10]
0x1755e:      mov     rsi, qword ptr [rsp + 0x18]
0x17563:      lea     rax, [rbp + rax - 1]
0x17568:      movsx   rcx, byte ptr [rax]
0x1756c:      test    byte ptr [r14 + rcx*2 + 1], 0x20
0x17572:      jne      0x17591
0x17574:      jmp     0x17596
0x17596
0x17576:      nop     word ptr cs:[rax + rax]
0x17580:      movsx   rcx, byte ptr [rax - 1]
0x17585:      sub     rax, 1

```

```

0x17589:    test    byte ptr [r14 + rcx*2 + 1], 0x20
0x1758f:    je      0x17596
0x17591:    cmp     rax, rbp
0x17594:    jae     0x17580
0x17596:    mov     byte ptr [rax + 1], 0
0x1759a:    mov     rax, qword ptr [rsi]
0x1759d:    cmp     byte ptr [rax], 0
0x175a0:    jne     0x17500
0x175a6:    mov     qword ptr [rsi], 0
0x175ad:    add     r8d, 1
0x175b1:    add     rsi, 8
0x175b5:    cmp     dword ptr [rsp + 0xc], r8d
0x175ba:    jg      0x17513
0x175c0:    test    ebx, ebx
0x175c2:    jne     0x173df
0x175c8:    mov     rsi, r12
0x175cb:    mov     rdi, r13
0x175ce:    call    0x48b0
0x175d3:    jmp     0x173df
0x173df
0x175d8:    nop     dword ptr [rax + rax]
0x175e0:    movsxd  rdx, dword ptr [rsp + 0xc]
0x175e5:    mov     rdi, qword ptr [rsp + 0x10]
0x175ea:    xor     esi, esi
0x175ec:    shl     rdx, 3
0x175f0:    call    0x4a50
0x175f5:    jmp     0x17375
0x17375
0x175fa:    nop     word ptr [rax + rax]
0x17600:    test    ebx, ebx
0x17602:    jne     0x173df
0x17608:    mov     byte ptr [r13], 0
0x1760d:    jmp     0x173df
0x173df
0x17612:    nop     word ptr [rax + rax]
0x17618:    movsxd  rsi, dword ptr [rip + 0x14461]
0x1761f:    mov     rdi, qword ptr [rip + 0x14462]
0x17626:    xor     edx, edx
0x17628:    call    0x4c80
0x1762d:    jmp     0x1735e
0x1735e
0x17632:    nop     word ptr [rax + rax]
0x17638:    mov     ecx, eax
0x1763a:    jmp     0x174cb
0x174cb
0x1763f:    mov     esi, 0x3b
0x17644:    call    0x4a00
0x17649:    test    rax, rax

```

```

0x1764c:      jne      0x174e0
0x17652:      jmp      0x174e3
0x174e3:
0x17657:      call     0x49d0
0x1765c:      nop      dword ptr [rax]
0x17660:      endbr64
0x17664:      push     r12
0x17666:      mov      edi, 0xf
0x1766b:      lea      r12, [rip + 0xa8b4]
0x17672:      call     0x1d670
0x17677:      mov      rcx, r12
0x1767a:      lea      r9, [rip + 0xa895]
0x17681:      lea      r8, [rip + 0xa893]
0x17688:      mov      rdi, rax
0x1768b:      mov      rdx, -1
0x17692:      mov      esi, 1
0x17697:      mov      qword ptr [rip + 0x14402], rax
0x1769e:      xor      eax, eax
0x176a0:      call     0x4f10
0x176a5:      mov      edi, 0x17
0x176aa:      call     0x1d670
0x176af:      mov      rcx, r12
0x176b2:      pop      r12
0x176b4:      lea      r9, [rip + 0xa870]
0x176bb:      mov      qword ptr [rip + 0x143e6], rax
0x176c2:      mov      rdi, rax
0x176c5:      xor      eax, eax
0x176c7:      lea      r8, [rip + 0xa84d]
0x176ce:      mov      rdx, -1
0x176d5:      mov      esi, 1
0x176da:      jmp      0x4f10
0x4f10:
0x176df:      nop
0x176e0:      endbr64
0x176e4:      push     rbp
0x176e5:      mov      rbp, rdi
0x176e8:      xor      edi, edi
0x176ea:      push     rbx
0x176eb:      sub      rsp, 0x18
0x176ef:      mov      rax, qword ptr fs:[0x28]
0x176f8:      mov      qword ptr [rsp + 8], rax
0x176fd:      xor      eax, eax
0x176ff:      call     0x4be0
0x17704:      mov      rdi, rsp
0x17707:      mov      qword ptr [rsp], rax
0x1770b:      call     0x4840
0x17710:      mov      rbx, rax
0x17713:      call     0x4920

```

```

0x17718:      movsxd  rcx, dword ptr [rbx + 0x10]
0x1771c:      sub     rsp, 8
0x17720:      lea     rdx, [rip + 0x12fd9]
0x17727:      mov     r9d, dword ptr [rbx + 8]
0x1772b:      mov     r8d, dword ptr [rbx + 0xc]
0x1772f:      mov     esi, 1
0x17734:      mov     rcx, qword ptr [rdx + rcx*8]
0x17738:      push    rbp
0x17739:      lea     rdx, [rip + 0xa8d0]
0x17740:      push    rax
0x17741:      mov     rdi, qword ptr [rip + 0x13138]
0x17748:      push    qword ptr [rip + 0x1319a]
0x1774e:      mov     eax, dword ptr [rbx]
0x17750:      push    rax
0x17751:      mov     eax, dword ptr [rbx + 4]
0x17754:      push    rax
0x17755:      xor     eax, eax
0x17757:      call    0x4e60
0x1775c:      add     rsp, 0x30
0x17760:      mov     rax, qword ptr [rsp + 8]
0x17765:      sub     rax, qword ptr fs:[0x28]
0x1776e:      jne     0x17777
0x17770:      add     rsp, 0x18
0x17774:      pop     rbx
0x17775:      pop     rbp
0x17776:      ret
0x17777:      call    0x49d0
0x1777c:      nop     dword ptr [rax]
0x17780:      endbr64
0x17784:      push    rbp
0x17785:      xor     eax, eax
0x17787:      mov     ebp, edi
0x17789:      call    0x8ad0
0x1778e:      cmp     ebp, 0xb
0x17791:      je      0x177d3
0x17793:      jg      0x177c0
0x17795:      lea     eax, [rbp - 1]
0x17798:      cmp     eax, 1
0x1779b:      ja      0x17802
0x1779d:      cmp     ebp, 1
0x177a0:      lea     rdi, [rip + 0xa791]
0x177a7:      lea     rax, [rip + 0xa78e]
0x177ae:      cmovne  rdi, rax
0x177b2:      call    0x176e0
0x177b7:      mov     edi, ebp
0x177b9:      call    0x4e20
0x177be:      nop
0x177c0:      cmp     ebp, 0xf

```



```

0x177c3:      jne      0x17802
0x177c5:      lea      rdi, [rip + 0x8dbe]
0x177cc:      call    0x176e0
0x177d1:      jmp      0x177b7
0x177b7
0x177d3:      lea      rdi, [rip + 0xa766]
0x177da:      call    0x176e0
0x177df:      mov      rcx, qword ptr [rip + 0x13102]
0x177e6:      mov      rdi, qword ptr [rip + 0x13093]
0x177ed:      lea      rdx, [rip + 0xa88c]
0x177f4:      xor      eax, eax
0x177f6:      mov      esi, 1
0x177fb:      call    0x4e60
0x17800:      jmp      0x177b7
0x177b7
0x17802:      mov      rcx, qword ptr [rip + 0x130df]
0x17809:      mov      rdi, qword ptr [rip + 0x13070]
0x17810:      mov      r8d, ebp
0x17813:      xor      eax, eax
0x17815:      lea      rdx, [rip + 0xa82c]
0x1781c:      mov      esi, 1
0x17821:      call    0x4e60
0x17826:      jmp      0x177df
0x177df
0x17828:      nop      dword ptr [rax + rax]
0x17830:      endbr64
0x17834:      push     rbx
0x17835:      mov      rcx, qword ptr [rip + 0x130ac]
0x1783c:      mov      rbx, rdi
0x1783f:      mov      esi, 1
0x17844:      mov      rdi, qword ptr [rip + 0x13035]
0x1784b:      lea      rdx, [rip + 0xa6f3]
0x17852:      xor      eax, eax
0x17854:      call    0x4e60
0x17859:      mov      r8, rbx
0x1785c:      mov      rcx, qword ptr [rip + 0x13085]
0x17863:      pop      rbx
0x17864:      mov      rdi, qword ptr [rip + 0x13015]
0x1786b:      lea      rdx, [rip + 0xa836]
0x17872:      mov      esi, 1
0x17877:      xor      eax, eax
0x17879:      jmp      0x4e60
0x4e60
0x1787e:      nop
0x17880:      endbr64
0x17884:      sub      rsp, 8
0x17888:      mov      r8, rdi
0x1788b:      mov      rcx, qword ptr [rip + 0x13056]

```

```

0x17892:    xor     eax, eax
0x17894:    mov     rdi, qword ptr [rip + 0x12fe5]
0x1789b:    lea     rdx, [rip + 0xa836]
0x178a2:    mov     esi, 1
0x178a7:    call    0x4e60
0x178ac:    mov     rcx, qword ptr [rip + 0x13035]
0x178b3:    mov     esi, 1
0x178b8:    xor     eax, eax
0x178ba:    mov     rdi, qword ptr [rip + 0x12fbf]
0x178c1:    lea     rdx, [rip + 0xa840]
0x178c8:    call    0x4e60
0x178cd:    mov     rcx, qword ptr [rip + 0x13014]
0x178d4:    mov     esi, 1
0x178d9:    xor     eax, eax
0x178db:    mov     rdi, qword ptr [rip + 0x12f9e]
0x178e2:    lea     rdx, [rip + 0xa84f]
0x178e9:    call    0x4e60
0x178ee:    mov     rcx, qword ptr [rip + 0x12ff3]
0x178f5:    mov     esi, 1
0x178fa:    xor     eax, eax
0x178fc:    mov     rdi, qword ptr [rip + 0x12f7d]
0x17903:    lea     rdx, [rip + 0xa86e]
0x1790a:    call    0x4e60
0x1790f:    mov     rcx, qword ptr [rip + 0x12fd2]
0x17916:    mov     esi, 1
0x1791b:    xor     eax, eax
0x1791d:    mov     rdi, qword ptr [rip + 0x12f5c]
0x17924:    lea     rdx, [rip + 0xa885]
0x1792b:    call    0x4e60
0x17930:    mov     rcx, qword ptr [rip + 0x12fb1]
0x17937:    mov     esi, 1
0x1793c:    xor     eax, eax
0x1793e:    mov     rdi, qword ptr [rip + 0x12f3b]
0x17945:    lea     rdx, [rip + 0xa8a4]
0x1794c:    call    0x4e60
0x17951:    mov     rcx, qword ptr [rip + 0x12f90]
0x17958:    mov     esi, 1
0x1795d:    xor     eax, eax
0x1795f:    mov     rdi, qword ptr [rip + 0x12f1a]
0x17966:    lea     rdx, [rip + 0xa8c3]
0x1796d:    call    0x4e60
0x17972:    mov     rcx, qword ptr [rip + 0x12f6f]
0x17979:    mov     esi, 1
0x1797e:    xor     eax, eax
0x17980:    mov     rdi, qword ptr [rip + 0x12ef9]
0x17987:    lea     rdx, [rip + 0xa8ea]
0x1798e:    add     rsp, 8
0x17992:    jmp     0x4e60

```

```

0x4e60
0x17997:      nop      word ptr [rax + rax]
0x179a0:      endbr64
0x179a4:      push     r12
0x179a6:      lea      r12, [rip + 0x12d33]
0x179ad:      push     rbp
0x179ae:      sub      rsp, 8
0x179b2:      mov      rdi, qword ptr [rip + 0x140ef]
0x179b9:      call     0x49b0
0x179be:      mov      rdi, r12
0x179c1:      mov      rbp, rax
0x179c4:      call     0x49b0
0x179c9:      lea      rdi, [rbp + rax + 9]
0x179ce:      call     0x1d670
0x179d3:      mov      rsi, qword ptr [rip + 0x140ce]
0x179da:      mov      rdi, rax
0x179dd:      mov      rbp, rax
0x179e0:      call     0x4980
0x179e5:      mov      rsi, r12
0x179e8:      mov      rdi, rax
0x179eb:      call     0x4980
0x179f0:      mov      edx, 0x6e6f
0x179f5:      mov      rdi, rbp
0x179f8:      mov      dword ptr [rax], 0x6d6d6f63
0x179fe:      mov      word ptr [rax + 4], dx
0x17a02:      mov      byte ptr [rax + 6], 0
0x17a06:      call     0x17230
0x17a0b:      test     eax, eax
0x17a0d:      je       0x17a17
0x17a0f:      add      rsp, 8
0x17a13:      pop      rbp
0x17a14:      pop      r12
0x17a16:      ret
0x17a17:      mov      rdi, rbp
0x17a1a:      call     0x17830
0x17a1f:      mov      edi, 1
0x17a24:      call     0x4e20
0x17a29:      nop      dword ptr [rax]
0x17a30:      endbr64
0x17a34:      push     r13
0x17a36:      push     r12
0x17a38:      lea      r12, [rip + 0x12c91]
0x17a3f:      push     rbp
0x17a40:      push     rbx
0x17a41:      sub      rsp, 8
0x17a45:      mov      rdi, qword ptr [rip + 0x12ea4]
0x17a4c:      call     0x49b0
0x17a51:      mov      rdi, r12

```

```

0x17a54:    mov     rbx, rax
0x17a57:    call   0x49b0
0x17a5c:    mov     rdi, qword ptr [rip + 0x12eed]
0x17a63:    mov     rbp, rax
0x17a66:    call   0x49b0
0x17a6b:    lea     rdi, [rbx + rbp + 2]
0x17a70:    add     rdi, rax
0x17a73:    call   0x1d670
0x17a78:    mov     rsi, qword ptr [rip + 0x12e71]
0x17a7f:    mov     rdi, rax
0x17a82:    mov     rbp, rax
0x17a85:    call   0x4980
0x17a8a:    mov     rsi, r12
0x17a8d:    mov     rdi, rax
0x17a90:    call   0x4980
0x17a95:    mov     rsi, qword ptr [rip + 0x12eb4]
0x17a9c:    mov     rdi, rax
0x17a9f:    call   0x48b0
0x17aa4:    mov     rdi, rbp
0x17aa7:    call   0x17230
0x17aac:    test    eax, eax
0x17aae:    jne     0x17b2a
0x17ab0:    mov     rdi, rbp
0x17ab3:    lea     r12, [rip + 0x12c06]
0x17aba:    call   0x18d10
0x17abf:    mov     rdi, qword ptr [rip + 0x12e2a]
0x17ac6:    call   0x49b0
0x17acb:    mov     rdi, r12
0x17ace:    mov     rbp, rax
0x17ad1:    call   0x49b0
0x17ad6:    mov     rdi, qword ptr [rip + 0x12e73]
0x17add:    mov     rbx, rax
0x17ae0:    call   0x49b0
0x17ae5:    lea     rdi, [rbp + rbx + 2]
0x17aea:    add     rdi, rax
0x17aed:    call   0x1d670
0x17af2:    mov     rsi, qword ptr [rip + 0x12df7]
0x17af9:    mov     rdi, rax
0x17afc:    mov     rbp, rax
0x17aff:    call   0x4980
0x17b04:    mov     rsi, r12
0x17b07:    mov     rdi, rax
0x17b0a:    call   0x4980
0x17b0f:    mov     rsi, qword ptr [rip + 0x12e3a]
0x17b16:    mov     rdi, rax
0x17b19:    call   0x48b0
0x17b1e:    mov     rdi, rbp
0x17b21:    call   0x17230

```

```

0x17b26:    test    eax, eax
0x17b28:    je      0x17b50
0x17b2a:    mov     rdi, rbp
0x17b2d:    mov     r12d, 1
0x17b33:    call    0x18d10
0x17b38:    add     rsp, 8
0x17b3c:    mov     eax, r12d
0x17b3f:    pop     rbx
0x17b40:    pop     rbp
0x17b41:    pop     r12
0x17b43:    pop     r13
0x17b45:    ret
0x17b46:    nop     word ptr cs:[rax + rax]
0x17b50:    mov     rdi, rbp
0x17b53:    lea     r13, [rip + 0x12b86]
0x17b5a:    call    0x18d10
0x17b5f:    mov     rdi, qword ptr [rip + 0x13f42]
0x17b66:    call    0x49b0
0x17b6b:    mov     rdi, r13
0x17b6e:    mov     rbp, rax
0x17b71:    call    0x49b0
0x17b76:    mov     rdi, qword ptr [rip + 0x12dd3]
0x17b7d:    mov     rbx, rax
0x17b80:    call    0x49b0
0x17b85:    lea     rdi, [rbp + rbx + 2]
0x17b8a:    add     rdi, rax
0x17b8d:    call    0x1d670
0x17b92:    mov     rsi, qword ptr [rip + 0x13f0f]
0x17b99:    mov     rdi, rax
0x17b9c:    mov     rbp, rax
0x17b9f:    call    0x4980
0x17ba4:    mov     rsi, r13
0x17ba7:    mov     rdi, rax
0x17baa:    call    0x4980
0x17baf:    mov     rsi, qword ptr [rip + 0x12d9a]
0x17bb6:    mov     rdi, rax
0x17bb9:    call    0x48b0
0x17bbe:    mov     rdi, rbp
0x17bc1:    call    0x17230
0x17bc6:    mov     r12d, eax
0x17bc9:    test    eax, eax
0x17bcb:    jne     0x17b2a
0x17bd1:    xor     eax, eax
0x17bd3:    mov     rdi, rbp
0x17bd6:    call    0x17880
0x17bdb:    mov     rdi, rbp
0x17bde:    call    0x18d10
0x17be3:    mov     rdi, qword ptr [rip + 0x13ebe]

```

```

0x17bea:      call    0x49b0
0x17bef:      mov     rdi, r13
0x17bf2:      mov     rbx, rax
0x17bf5:      call    0x49b0
0x17bfa:      lea     rdi, [rbx + rax + 0xa]
0x17bff:      call    0x1d670
0x17c04:      mov     rsi, qword ptr [rip + 0x13e9d]
0x17c0b:      mov     rdi, rax
0x17c0e:      mov     rbp, rax
0x17c11:      call    0x4980
0x17c16:      mov     rsi, r13
0x17c19:      mov     rdi, rax
0x17c1c:      call    0x4980
0x17c21:      mov     rdi, rbp
0x17c24:      movabs  rdx, 0x636972656e6567
0x17c2e:      mov     qword ptr [rax], rdx
0x17c31:      call    0x17230
0x17c36:      test    eax, eax
0x17c38:      jne     0x17b38
0x17c3e:      mov     rdi, rbp
0x17c41:      call    0x17830
0x17c46:      mov     edi, 1
0x17c4b:      call    0x4e20
0x17c50:      endbr64
0x17c54:      push    rbp
0x17c55:      push    rbx
0x17c56:      mov     rbx, rdi
0x17c59:      sub     rsp, 8
0x17c5d:      call    0x172a0
0x17c62:      cmp     eax, -1
0x17c65:      je      0x17c70
0x17c67:      add     rsp, 8
0x17c6b:      pop     rbx
0x17c6c:      pop     rbp
0x17c6d:      ret
0x17c6e:      nop
0x17c70:      mov     edx, 7
0x17c75:      lea     rsi, [rip + 0xa2e5]
0x17c7c:      mov     rdi, rbx
0x17c7f:      call    0x4890
0x17c84:      test    eax, eax
0x17c86:      jne     0x17cd4
0x17c88:      lea     rdi, [rbx + 5]
0x17c8c:      call    0x49b0
0x17c91:      lea     rdi, [rax + 6]
0x17c95:      call    0x1d670
0x17c9a:      lea     rsi, [rbx + 7]
0x17c9e:      mov     dword ptr [rax], 0x5449475b

```

```

0x17ca4:    mov     rbp, rax
0x17ca7:    lea     rdi, [rax + 5]
0x17cab:    mov     byte ptr [rax + 4], 0x2d
0x17caf:    call    0x48b0
0x17cb4:    mov     rdi, rbp
0x17cb7:    call    0x172a0
0x17cbc:    mov     rdi, rbp
0x17cbf:    cmp     eax, -1
0x17cc2:    je      0x17ccf
0x17cc4:    add     rsp, 8
0x17cc8:    pop     rbx
0x17cc9:    pop     rbp
0x17cca:    jmp     0x18d10
0x18d10:
0x17ccf:    call    0x18d10
0x17cd4:    mov     rdi, qword ptr [rip + 0x12ba5]
0x17cdb:    mov     rcx, qword ptr [rip + 0x12c06]
0x17ce2:    mov     r8, rbx
0x17ce5:    xor     eax, eax
0x17ce7:    lea     rdx, [rip + 0xa5ba]
0x17cee:    mov     esi, 1
0x17cf3:    call    0x4e60
0x17cf8:    mov     edi, 1
0x17cfd:    call    0x4e20
0x17d02:    nop     word ptr cs:[rax + rax]
0x17d0d:    nop     dword ptr [rax]
0x17d10:    endbr64
0x17d14:    push    rbx
0x17d15:    mov     ecx, 1
0x17d1a:    mov     ebx, esi
0x17d1c:    mov     edx, 1
0x17d21:    sub     rsp, 0x10
0x17d25:    mov     rax, qword ptr fs:[0x28]
0x17d2e:    mov     qword ptr [rsp + 8], rax
0x17d33:    xor     eax, eax
0x17d35:    mov     rsi, rsp
0x17d38:    call    0x17320
0x17d3d:    mov     rdi, qword ptr [rsp]
0x17d41:    mov     eax, ebx
0x17d43:    test    rdi, rdi
0x17d46:    je      0x17d54
0x17d48:    mov     edx, 0xa
0x17d4d:    xor     esi, esi
0x17d4f:    call    0x4ba0
0x17d54:    mov     rdx, qword ptr [rsp + 8]
0x17d59:    sub     rdx, qword ptr fs:[0x28]
0x17d62:    jne     0x17d6a
0x17d64:    add     rsp, 0x10

```

```

0x17d68:    pop     rbx
0x17d69:    ret
0x17d6a:    call    0x49d0
0x17d6f:    nop
0x17d70:    endbr64
0x17d74:    push    r15
0x17d76:    push    r14
0x17d78:    mov     r14, rdi
0x17d7b:    push    r13
0x17d7d:    mov     r13d, ecx
0x17d80:    mov     ecx, 1
0x17d85:    push    r12
0x17d87:    mov     r12d, r13d
0x17d8a:    push    rbp
0x17d8b:    mov     ebp, edx
0x17d8d:    mov     edx, 1
0x17d92:    push    rbx
0x17d93:    mov     rbx, rsi
0x17d96:    sub     rsp, 0x18
0x17d9a:    mov     rax, qword ptr fs:[0x28]
0x17da3:    mov     qword ptr [rsp + 8], rax
0x17da8:    xor     eax, eax
0x17daa:    mov     rsi, rsp
0x17dad:    call    0x17320
0x17db2:    mov     r15, qword ptr [rsp]
0x17db6:    test    r15, r15
0x17db9:    je      0x17de4
0x17dbb:    test    ebp, ebp
0x17dbd:    jle     0x17e3c
0x17dbf:    xor     r12d, r12d
0x17dc2:    jmp     0x17dd5
0x17dd5:
0x17dc4:    nop     dword ptr [rax]
0x17dc8:    add     r12d, 1
0x17dcc:    add     rbx, 8
0x17dd0:    cmp     ebp, r12d
0x17dd3:    je      0x17e10
0x17dd5:    mov     rsi, qword ptr [rbx]
0x17dd8:    mov     rdi, r15
0x17ddb:    call    0x4b30
0x17de0:    test    eax, eax
0x17de2:    jne     0x17dc8
0x17de4:    mov     rax, qword ptr [rsp + 8]
0x17de9:    sub     rax, qword ptr fs:[0x28]
0x17df2:    jne     0x17e43
0x17df4:    add     rsp, 0x18
0x17df8:    mov     eax, r12d
0x17dfb:    pop     rbx

```



```

0x17dfc:      pop     rbp
0x17dfd:      pop     r12
0x17dff:      pop     r13
0x17e01:      pop     r14
0x17e03:      pop     r15
0x17e05:      ret
0x17e06:      nop     word ptr cs:[rax + rax]
0x17e10:      mov     rcx, qword ptr [rip + 0x12ad1]
0x17e17:      mov     r9, r15
0x17e1a:      mov     r8, r14
0x17e1d:      xor     eax, eax
0x17e1f:      mov     rdi, qword ptr [rip + 0x12a5a]
0x17e26:      lea     rdx, [rip + 0xa13c]
0x17e2d:      mov     esi, 1
0x17e32:      mov     r12d, r13d
0x17e35:      call    0x4e60
0x17e3a:      jmp     0x17de4
0x17de4
0x17e3c:      je      0x17e10
0x17e3e:      xor     r12d, r12d
0x17e41:      jmp     0x17de4
0x17de4
0x17e43:      call    0x49d0
0x17e48:      nop     dword ptr [rax + rax]
0x17e50:      endbr64
0x17e54:      push    r12
0x17e56:      mov     ecx, 1
0x17e5b:      mov     edx, 1
0x17e60:      mov     r12, rdi
0x17e63:      push    rbp
0x17e64:      push    rbx
0x17e65:      mov     ebx, esi
0x17e67:      sub     rsp, 0x10
0x17e6b:      mov     rax, qword ptr fs:[0x28]
0x17e74:      mov     qword ptr [rsp + 8], rax
0x17e79:      xor     eax, eax
0x17e7b:      mov     rsi, rsp
0x17e7e:      call    0x17320
0x17e83:      mov     rbp, qword ptr [rsp]
0x17e87:      mov     eax, ebx
0x17e89:      test    rbp, rbp
0x17e8c:      je      0x17ea7
0x17e8e:      cmp     byte ptr [rbp], 0x4f
0x17e92:      je      0x17ec0
0x17e94:      lea     rsi, [rip + 0x8e4b]
0x17e9b:      mov     rdi, rbp
0x17e9e:      call    0x4b30
0x17ea3:      test    eax, eax

```

```

0x17ea5:    jne     0x17ed8
0x17ea7:    mov     rdx, qword ptr [rsp + 8]
0x17eac:    sub     rdx, qword ptr fs:[0x28]
0x17eb5:    jne     0x17f03
0x17eb7:    add     rsp, 0x10
0x17ebb:    pop     rbx
0x17ebc:    pop     rbp
0x17ebd:    pop     r12
0x17ebf:    ret
0x17ec0:    cmp     byte ptr [rbp + 1], 0x4e
0x17ec4:    jne     0x17e94
0x17ec6:    cmp     byte ptr [rbp + 2], 0
0x17eca:    mov     eax, 1
0x17ecf:    je      0x17ea7
0x17ed1:    jmp     0x17e94
0x17e94:
0x17ed3:    nop     dword ptr [rax + rax]
0x17ed8:    mov     rcx, qword ptr [rip + 0x12a09]
0x17edf:    mov     r9, rbp
0x17ee2:    mov     r8, r12
0x17ee5:    xor     eax, eax
0x17ee7:    mov     rdi, qword ptr [rip + 0x12992]
0x17eee:    lea     rdx, [rip + 0xa074]
0x17ef5:    mov     esi, 1
0x17efa:    call    0x4e60
0x17eff:    mov     eax, ebx
0x17f01:    jmp     0x17ea7
0x17ea7:
0x17f03:    call    0x49d0
0x17f08:    nop     dword ptr [rax + rax]
0x17f10:    endbr64
0x17f14:    push    r12
0x17f16:    mov     ecx, 1
0x17f1b:    mov     r12, rsi
0x17f1e:    mov     edx, 1
0x17f23:    sub     rsp, 0x10
0x17f27:    mov     rax, qword ptr fs:[0x28]
0x17f30:    mov     qword ptr [rsp + 8], rax
0x17f35:    xor     eax, eax
0x17f37:    mov     rsi, rsp
0x17f3a:    call    0x17320
0x17f3f:    mov     rdi, qword ptr [rsp]
0x17f43:    test    rdi, rdi
0x17f46:    je      0x17f50
0x17f48:    call    0x1d7a0
0x17f4d:    mov     r12, rax
0x17f50:    mov     rax, qword ptr [rsp + 8]
0x17f55:    sub     rax, qword ptr fs:[0x28]

```

```

0x17f5e:    jne     0x17f6a
0x17f60:    add     rsp, 0x10
0x17f64:    mov     rax, r12
0x17f67:    pop     r12
0x17f69:    ret
0x17f6a:    call    0x49d0
0x17f6f:    nop
0x17f70:    endbr64
0x17f74:    push    r14
0x17f76:    push    r13
0x17f78:    push    r12
0x17f7a:    push    rbp
0x17f7b:    push    rbx
0x17f7c:    sub     rsp, 0x10
0x17f80:    mov     rax, qword ptr fs:[0x28]
0x17f89:    mov     qword ptr [rsp + 8], rax
0x17f8e:    xor     eax, eax
0x17f90:    test    edx, edx
0x17f92:    jle     0x17fe8
0x17f94:    lea     eax, [rdx - 1]
0x17f97:    mov     rbx, rsi
0x17f9a:    mov     rbp, rdi
0x17f9d:    mov     r12, rsp
0x17fa0:    lea     r13, [rsi + rax*8 + 8]
0x17fa5:    lea     r14, [rip + 0x9fbd]
0x17fac:    nop     dword ptr [rax]
0x17fb0:    mov     rdi, qword ptr [rbx]
0x17fb3:    mov     ecx, 1
0x17fb8:    mov     edx, 1
0x17fbd:    mov     rsi, r12
0x17fc0:    call    0x17320
0x17fc5:    mov     rdi, qword ptr [rsp]
0x17fc9:    test    rdi, rdi
0x17fcc:    je      0x17fdb
0x17fce:    call    0xbd00
0x17fd3:    cmp     eax, -1
0x17fd6:    je      0x18008
0x17fd8:    mov     dword ptr [rbp], eax
0x17fdb:    add     rbx, 8
0x17fdf:    add     rbp, 4
0x17fe3:    cmp     rbx, r13
0x17fe6:    jne     0x17fb0
0x17fe8:    mov     rax, qword ptr [rsp + 8]
0x17fed:    sub     rax, qword ptr fs:[0x28]
0x17ff6:    jne     0x1802e
0x17ff8:    add     rsp, 0x10
0x17ffc:    pop     rbx
0x17ffd:    pop     rbp

```

```

0x17ffe:    pop     r12
0x18000:    pop     r13
0x18002:    pop     r14
0x18004:    ret
0x18005:    nop     dword ptr [rax]
0x18008:    mov     r9, qword ptr [rsp]
0x1800c:    mov     r8, qword ptr [rbx]
0x1800f:    mov     rdx, r14
0x18012:    mov     esi, 1
0x18017:    mov     rcx, qword ptr [rip + 0x128ca]
0x1801e:    mov     rdi, qword ptr [rip + 0x1285b]
0x18025:    xor     eax, eax
0x18027:    call    0x4e60
0x1802c:    jmp     0x17fdb
0x17fdb
0x1802e:    call    0x49d0
0x18033:    nop     word ptr cs:[rax + rax]
0x1803e:    nop
0x18040:    endbr64
0x18044:    push    r12
0x18046:    cmp     byte ptr [rdi], 0x2f
0x18049:    mov     r12, rdi
0x1804c:    lea     rax, [rdi + 1]
0x18050:    je      0x18061
0x18052:    lea     rdi, [rip + 0x9f26]
0x18059:    call    0x8b00
0x1805e:    mov     rax, r12
0x18061:    movzx   edx, byte ptr [rax]
0x18064:    mov     rsi, rax
0x18067:    test    dl, dl
0x18069:    jne     0x1808c
0x1806b:    jmp     0x180f0
0x180f0
0x18070:    cmp     dl, 0x2e
0x18073:    jne     0x18118
0x18079:    movzx   ecx, byte ptr [rax + 1]
0x1807d:    cmp     cl, 0x2f
0x18080:    je      0x18086
0x18082:    test    cl, cl
0x18084:    jne     0x180a0
0x18086:    add     rax, 1
0x1808a:    mov     edx, ecx
0x1808c:    cmp     dl, 0x2f
0x1808f:    jne     0x18070
0x18091:    movzx   ecx, byte ptr [rax + 1]
0x18095:    jmp     0x18086
0x18086
0x18097:    nop     word ptr [rax + rax]

```

```

0x180a0:    cmp     cl, 0x2e
0x180a3:    jne     0x180c0
0x180a5:    movzx   edx, byte ptr [rax + 2]
0x180a9:    cmp     dl, 0x2f
0x180ac:    je      0x18120
0x180ae:    test    dl, dl
0x180b0:    je      0x18120
0x180b2:    mov     edx, 0x2e
0x180b7:    nop     word ptr [rax + rax]
0x180c0:    mov     rdi, rax
0x180c3:    mov     rcx, rsi
0x180c6:    add     rax, 1
0x180ca:    add     rsi, 1
0x180ce:    mov     byte ptr [rsi - 1], dl
0x180d1:    movzx   edx, byte ptr [rax]
0x180d4:    test    dl, dl
0x180d6:    je      0x180f0
0x180d8:    cmp     dl, 0x2f
0x180db:    jne     0x180c0
0x180dd:    test    dl, dl
0x180df:    je      0x180f0
0x180e1:    mov     byte ptr [rsi], 0x2f
0x180e4:    movzx   edx, byte ptr [rdi + 1]
0x180e8:    lea     rsi, [rcx + 2]
0x180ec:    test    dl, dl
0x180ee:    jne     0x1808c
0x180f0:    cmp     byte ptr [rsi - 1], 0x2f
0x180f4:    jne     0x18109
0x180f6:    mov     rax, rsi
0x180f9:    sub     rax, r12
0x180fc:    cmp     rax, 1
0x18100:    setg    al
0x18103:    movzx   eax, al
0x18106:    sub     rsi, rax
0x18109:    mov     rax, r12
0x1810c:    mov     byte ptr [rsi], 0
0x1810f:    pop     r12
0x18111:    ret
0x18112:    nop     word ptr [rax + rax]
0x18118:    test    dl, dl
0x1811a:    jne     0x180c0
0x1811c:    jmp     0x180f0
0x180f0:
0x1811e:    nop
0x18120:    lea     rdi, [rsi - 1]
0x18124:    cmp     rdi, r12
0x18127:    jne     0x1812f
0x18129:    add     rax, 2

```

```

0x1812d:      jmp      0x180ec
0x180ec
0x1812f:      cmp      byte ptr [rsi - 2], 0x2f
0x18133:      lea      rcx, [rsi - 2]
0x18137:      je       0x1814e
0x18139:      nop      dword ptr [rax]
0x18140:      mov      rsi, rcx
0x18143:      sub      rcx, 1
0x18147:      cmp      byte ptr [rcx], 0x2f
0x1814a:      jne      0x18140
0x1814c:      jmp      0x18129
0x18129
0x1814e:      mov      rsi, rdi
0x18151:      jmp      0x18129
0x18129
0x18153:      nop      word ptr cs:[rax + rax]
0x1815e:      nop
0x18160:      endbr64
0x18164:      push     rbx
0x18165:      call     0x4a60
0x1816a:      mov      edi, eax
0x1816c:      mov      ebx, eax
0x1816e:      call     0x4970
0x18173:      test     rax, rax
0x18176:      je       0x18189
0x18178:      mov      rdi, qword ptr [rax]
0x1817b:      call     0x1d7a0
0x18180:      pop      rbx
0x18181:      mov      qword ptr [rip + 0x13910], rax
0x18188:      ret
0x18189:      mov      rcx, qword ptr [rip + 0x12758]
0x18190:      mov      r8d, ebx
0x18193:      mov      rdi, qword ptr [rip + 0x126e6]
0x1819a:      lea      rdx, [rip + 0xa13f]
0x181a1:      mov      esi, 1
0x181a6:      call     0x4e60
0x181ab:      mov      rcx, qword ptr [rip + 0x12736]
0x181b2:      xor      eax, eax
0x181b4:      mov      rdi, qword ptr [rip + 0x126c5]
0x181bb:      lea      r8, [rip + 0x9dd7]
0x181c2:      lea      rdx, [rip + 0xa14f]
0x181c9:      mov      esi, 1
0x181ce:      call     0x4e60
0x181d3:      mov      rdi, qword ptr [rip + 0x126a6]
0x181da:      mov      esi, 1
0x181df:      xor      eax, eax
0x181e1:      mov      rcx, qword ptr [rip + 0x12700]
0x181e8:      lea      rdx, [rip + 0xa161]

```

```

0x181ef:      call    0x4e60
0x181f4:      mov     edi, 1
0x181f9:      call    0x4e20
0x181fe:      nop
0x18200:      endbr64
0x18204:      push    r12
0x18206:      push    rbp
0x18207:      mov     rbp, rsi
0x1820a:      push    rbx
0x1820b:      movsxd  rbx, edx
0x1820e:      cmp     ebx, 2
0x18211:      je      0x18250
0x18213:      mov     r12, rdi
0x18216:      jg      0x18230
0x18218:      test    ebx, ebx
0x1821a:      je      0x18224
0x1821c:      cmp     ebx, 1
0x1821f:      jne     0x18260
0x18221:      mov     byte ptr [rsi], 0x20
0x18224:      pop     rbx
0x18225:      pop     rbp
0x18226:      pop     r12
0x18228:      ret
0x18229:      nop     dword ptr [rax]
0x18230:      cmp     ebx, 3
0x18233:      jne     0x18260
0x18235:      mov     edx, 0x2020
0x1823a:      mov     byte ptr [rsi + 2], 0x20
0x1823e:      mov     word ptr [rsi], dx
0x18241:      pop     rbx
0x18242:      pop     rbp
0x18243:      pop     r12
0x18245:      ret
0x18246:      nop     word ptr cs:[rax + rax]
0x18250:      mov     ecx, 0x2020
0x18255:      mov     word ptr [rsi], cx
0x18258:      pop     rbx
0x18259:      pop     rbp
0x1825a:      pop     r12
0x1825c:      ret
0x1825d:      nop     dword ptr [rax]
0x18260:      mov     rdi, r12
0x18263:      call    0x49b0
0x18268:      movsxd  rdx, eax
0x1826b:      cmp     ebx, eax
0x1826d:      jge     0x182a0
0x1826f:      mov     eax, 0x2e2e
0x18274:      lea     r8d, [rbx - 3]

```

```

0x18278:      mov     byte ptr [rbp + 2], 0x2e
0x1827c:      sub     rdx, rbx
0x1827f:      mov     word ptr [rbp], ax
0x18283:      movsxd  r8, r8d
0x18286:      lea     rsi, [r12 + rdx + 3]
0x1828b:      pop     rbx
0x1828c:      lea     rdi, [rbp + 3]
0x18290:      mov     rdx, r8
0x18293:      pop     rbp
0x18294:      pop     r12
0x18296:      jmp     0x4bb0
0x4bb0:
0x1829b:      nop     dword ptr [rax + rax]
0x182a0:      pop     rbx
0x182a1:      mov     rsi, r12
0x182a4:      mov     rdi, rbp
0x182a7:      pop     rbp
0x182a8:      pop     r12
0x182aa:      jmp     0x4bb0
0x4bb0:
0x182af:      nop
0x182b0:      endbr64
0x182b4:      push    r12
0x182b6:      mov     r12, rsi
0x182b9:      push    rbp
0x182ba:      mov     rbp, rdi
0x182bd:      push    rbx
0x182be:      lea     ebx, [rdx - 1]
0x182c1:      mov     edx, ebx
0x182c3:      call    0x18200
0x182c8:      mov     rdi, rbp
0x182cb:      call    0x49b0
0x182d0:      movsxd  rcx, ebx
0x182d3:      cmp     ebx, eax
0x182d5:      movsxd  rdx, eax
0x182d8:      mov     rax, rcx
0x182db:      pop     rbx
0x182dc:      cmovg   rax, rdx
0x182e0:      pop     rbp
0x182e1:      mov     byte ptr [r12 + rax], 0
0x182e6:      mov     rax, r12
0x182e9:      pop     r12
0x182eb:      ret
0x182ec:      nop     dword ptr [rax]
0x182f0:      endbr64
0x182f4:      push    r13
0x182f6:      mov     edx, 1
0x182fb:      xor     esi, esi

```



```

0x182fd:    push    r12
0x182ff:    push    rbp
0x18300:    mov     ebp, edi
0x18302:    call    0x4e50
0x18307:    mov     edi, ebp
0x18309:    mov     edx, 2
0x1830e:    xor     esi, esi
0x18310:    mov     r13, rax
0x18313:    call    0x4e50
0x18318:    mov     rsi, r13
0x1831b:    mov     edi, ebp
0x1831d:    xor     edx, edx
0x1831f:    mov     r12, rax
0x18322:    call    0x4e50
0x18327:    mov     rax, r12
0x1832a:    pop     rbp
0x1832b:    pop     r12
0x1832d:    pop     r13
0x1832f:    ret
0x18330:    endbr64
0x18334:    sub     rsp, 0x18
0x18338:    xor     edi, edi
0x1833a:    mov     rax, qword ptr fs:[0x28]
0x18343:    mov     qword ptr [rsp + 8], rax
0x18348:    xor     eax, eax
0x1834a:    call    0x4be0
0x1834f:    mov     rdi, rsp
0x18352:    mov     qword ptr [rsp], rax
0x18356:    call    0x4840
0x1835b:    mov     rdx, qword ptr [rsp + 8]
0x18360:    sub     rdx, qword ptr fs:[0x28]
0x18369:    jne     0x18370
0x1836b:    add     rsp, 0x18
0x1836f:    ret
0x18370:    call    0x49d0
0x18375:    nop     word ptr cs:[rax + rax]
0x18380:    endbr64
0x18384:    push    r12
0x18386:    push    rbp
0x18387:    mov     rbp, rsi
0x1838a:    push    rbx
0x1838b:    mov     rbx, rdi
0x1838e:    call    0x49b0
0x18393:    mov     rdi, rbp
0x18396:    mov     r12, rax
0x18399:    call    0x49b0
0x1839e:    lea     rdi, [r12 + rax + 2]
0x183a3:    call    0x1d670

```

```

0x183a8:    mov     r9, rbp
0x183ab:    mov     r8, rbx
0x183ae:    mov     esi, 1
0x183b3:    mov     r12, rax
0x183b6:    mov     rdi, rax
0x183b9:    xor     eax, eax
0x183bb:    mov     rdx, -1
0x183c2:    lea     rcx, [rip + 0x9be5]
0x183c9:    call    0x4f10
0x183ce:    mov     rdi, r12
0x183d1:    call    0x4b20
0x183d6:    mov     r12d, eax
0x183d9:    cmp     eax, -1
0x183dc:    je      0x183f0
0x183de:    mov     eax, r12d
0x183e1:    pop     rbx
0x183e2:    pop     rbp
0x183e3:    pop     r12
0x183e5:    ret
0x183e6:    nop     word ptr cs:[rax + rax]
0x183f0:    mov     rcx, qword ptr [rip + 0x124f1]
0x183f7:    mov     rdi, qword ptr [rip + 0x12482]
0x183fe:    mov     r8, rbx
0x18401:    xor     eax, eax
0x18403:    lea     rdx, [rip + 0x9f6e]
0x1840a:    mov     esi, 1
0x1840f:    call    0x4e60
0x18414:    mov     eax, r12d
0x18417:    pop     rbx
0x18418:    pop     rbp
0x18419:    pop     r12
0x1841b:    ret
0x1841c:    nop     dword ptr [rax]
0x18420:    endbr64
0x18424:    push    r13
0x18426:    push    r12
0x18428:    lea     r12, [rip + 0x9b85]
0x1842f:    push    rbp
0x18430:    mov     rdi, r12
0x18433:    push    rbx
0x18434:    sub     rsp, 8
0x18438:    call    0x47e0
0x1843d:    mov     r13, qword ptr [rip + 0x1365c]
0x18444:    test    rax, rax
0x18447:    je      0x184b0
0x18449:    mov     rdi, rax
0x1844c:    mov     rbx, rax
0x1844f:    call    0x49b0

```

```

0x18454:      mov     rdi, r13
0x18457:      mov     rbp, rax
0x1845a:      call    0x49b0
0x1845f:      lea     rdi, [rbp + rax + 2]
0x18464:      call    0x1d670
0x18469:      mov     r9, qword ptr [rip + 0x13630]
0x18470:      mov     r8, rbx
0x18473:      lea     rcx, [rip + 0x9b3f]
0x1847a:      mov     rbp, rax
0x1847d:      mov     rdi, rax
0x18480:      mov     esi, 1
0x18485:      xor     eax, eax
0x18487:      mov     rdx, -1
0x1848e:      call    0x4f10
0x18493:      mov     rdi, r12
0x18496:      mov     rsi, rbp
0x18499:      call    0x18380
0x1849e:      add     rsp, 8
0x184a2:      mov     rdi, rbp
0x184a5:      pop     rbx
0x184a6:      pop     rbp
0x184a7:      pop     r12
0x184a9:      pop     r13
0x184ab:      jmp     0x18d10
0x18d10
0x184b0:      add     rsp, 8
0x184b4:      mov     rsi, r13
0x184b7:      mov     rdi, r12
0x184ba:      pop     rbx
0x184bb:      pop     rbp
0x184bc:      pop     r12
0x184be:      pop     r13
0x184c0:      jmp     0x18380
0x18380
0x184c5:      nop     word ptr cs:[rax + rax]
0x184d0:      endbr64
0x184d4:      test    rsi, rsi
0x184d7:      je      0x18520
0x184d9:      push    rbp
0x184da:      mov     rbp, rsi
0x184dd:      push    rbx
0x184de:      mov     rbx, rdi
0x184e1:      sub     rsp, 8
0x184e5:      call    0x4f00
0x184ea:      mov     rdi, rbx
0x184ed:      lea     rsi, [rbx + rbp]
0x184f1:      nop     dword ptr [rax]
0x184f8:      movsx   rcx, byte ptr [rdi]

```

```

0x184fc:    mov     rdx, qword ptr [rax]
0x184ff:    test    byte ptr [rdx + rcx*2 + 1], 0x40
0x18504:    jne     0x18509
0x18506:    mov     byte ptr [rdi], 0x3f
0x18509:    add     rdi, 1
0x1850d:    cmp     rsi, rdi
0x18510:    jne     0x184f8
0x18512:    add     rsp, 8
0x18516:    pop     rbx
0x18517:    pop     rbp
0x18518:    ret
0x18519:    nop     dword ptr [rax]
0x18520:    ret
0x18521:    nop     word ptr cs:[rax + rax]
0x1852c:    nop     dword ptr [rax]
0x18530:    endbr64
0x18534:    test    rsi, rsi
0x18537:    je      0x185c2
0x1853d:    push    rbp
0x1853e:    mov     rbp, rsi
0x18541:    push    rbx
0x18542:    mov     rbx, rdi
0x18545:    sub     rsp, 8
0x18549:    call    0x4f00
0x1854e:    mov     rdi, rbx
0x18551:    lea     rsi, [rbx + rbp]
0x18555:    movabs  r9, 0x68000000c0000001
0x1855f:    movabs  r8, 0x2400f82000000000
0x18569:    mov     rcx, qword ptr [rax]
0x1856c:    nop     dword ptr [rax]
0x18570:    movsx   rdx, byte ptr [rdi]
0x18574:    mov     rax, rdx
0x18577:    test    byte ptr [rcx + rdx*2], 8
0x1857b:    jne     0x185b0
0x1857d:    cmp     dl, 0x3d
0x18580:    jg      0x185a0
0x18582:    cmp     dl, 0x24
0x18585:    jle     0x1858d
0x18587:    bt      r8, rdx
0x1858b:    jb      0x185b0
0x1858d:    add     rsp, 8
0x18591:    mov     eax, 1
0x18596:    pop     rbx
0x18597:    pop     rbp
0x18598:    ret
0x18599:    nop     dword ptr [rax]
0x185a0:    sub     eax, 0x40
0x185a3:    cmp     al, 0x3e

```

```

0x185a5:    ja     0x1858d
0x185a7:    bt     r9, rax
0x185ab:    jae     0x1858d
0x185ad:    nop     dword ptr [rax]
0x185b0:    add     rdi, 1
0x185b4:    cmp     rsi, rdi
0x185b7:    jne     0x18570
0x185b9:    add     rsp, 8
0x185bd:    xor     eax, eax
0x185bf:    pop     rbx
0x185c0:    pop     rbp
0x185c1:    ret
0x185c2:    xor     eax, eax
0x185c4:    ret
0x185c5:    nop     word ptr cs:[rax + rax]
0x185d0:    endbr64
0x185d4:    push    rbx
0x185d5:    mov     rbx, rdi
0x185d8:    call    0x49b0
0x185dd:    sub     eax, 1
0x185e0:    js      0x18618
0x185e2:    cdqe
0x185e4:    jmp     0x18602
0x18602:
0x185e6:    nop     word ptr cs:[rax + rax]
0x185f0:    cmp     dl, 0x20
0x185f3:    je      0x185fa
0x185f5:    cmp     dl, 9
0x185f8:    jne     0x18618
0x185fa:    sub     rax, 1
0x185fe:    test    eax, eax
0x18600:    js      0x18618
0x18602:    movzx   edx, byte ptr [rbx + rax]
0x18606:    cmp     dl, 0x26
0x18609:    jne     0x185f0
0x1860b:    mov     eax, 1
0x18610:    pop     rbx
0x18611:    ret
0x18612:    nop     word ptr [rax + rax]
0x18618:    xor     eax, eax
0x1861a:    pop     rbx
0x1861b:    ret
0x1861c:    nop     dword ptr [rax]
0x18620:    endbr64
0x18624:    jmp     0x1863c
0x1863c:
0x18626:    nop     word ptr cs:[rax + rax]
0x18630:    cmp     al, 0x20

```

```

0x18632:    je      0x18638
0x18634:    cmp     al, 0x3b
0x18636:    jne     0x18650
0x18638:    add     rdi, 1
0x1863c:    movzx   eax, byte ptr [rdi]
0x1863f:    test    al, al
0x18641:    jne     0x18630
0x18643:    mov     eax, 1
0x18648:    ret
0x18649:    nop     dword ptr [rax]
0x18650:    xor     eax, eax
0x18652:    ret
0x18653:    nop     word ptr cs:[rax + rax]
0x1865e:    nop
0x18660:    endbr64
0x18664:    push    r15
0x18666:    push    r14
0x18668:    push    r13
0x1866a:    push    r12
0x1866c:    push    rbp
0x1866d:    push    rbx
0x1866e:    sub     rsp, 0x98
0x18675:    mov     rax, qword ptr fs:[0x28]
0x1867e:    mov     qword ptr [rsp + 0x88], rax
0x18686:    xor     eax, eax
0x18688:    lea     r15, [rsp + 0x10]
0x1868d:    lea     r13, [rsp + 0x30]
0x18692:    mov     qword ptr [rsp + 8], 0
0x1869b:    mov     qword ptr [rsp], 0
0x186a3:    jmp     0x186d3
0x186d3:
0x186a5:    nop     dword ptr [rax]
0x186a8:    mov     rax, qword ptr [rsp]
0x186ac:    mov     qword ptr [rax + 0x18], r14
0x186b0:    mov     rdi, r13
0x186b3:    call    0x1d7a0
0x186b8:    mov     dword ptr [r14 + 8], r12d
0x186bc:    mov     qword ptr [r14], rax
0x186bf:    mov     dword ptr [r14 + 0xc], ebp
0x186c3:    mov     dword ptr [r14 + 0x10], ebx
0x186c7:    mov     qword ptr [r14 + 0x18], 0
0x186cf:    mov     qword ptr [rsp], r14
0x186d3:    xor     ecx, ecx
0x186d5:    mov     edx, 3
0x186da:    mov     rsi, r15
0x186dd:    mov     rdi, r13
0x186e0:    call    0x17320
0x186e5:    cmp     byte ptr [rsp + 0x30], 0

```

```

0x186ea:    je      0x18758
0x186ec:    mov     rdi, qword ptr [rsp + 0x10]
0x186f1:    mov     r12d, 0xffffffff
0x186f7:    test    rdi, rdi
0x186fa:    je      0x18704
0x186fc:    call    0xbd00
0x18701:    mov     r12d, eax
0x18704:    mov     rdi, qword ptr [rsp + 0x18]
0x18709:    mov     ebp, 0xffffffff
0x1870e:    test    rdi, rdi
0x18711:    je      0x1871a
0x18713:    call    0xbd00
0x18718:    mov     ebp, eax
0x1871a:    mov     rdi, qword ptr [rsp + 0x20]
0x1871f:    mov     ebx, 0xffffffff
0x18724:    test    rdi, rdi
0x18727:    je      0x18730
0x18729:    call    0xbd00
0x1872e:    mov     ebx, eax
0x18730:    mov     edi, 0x20
0x18735:    call    0x1d670
0x1873a:    cmp     qword ptr [rsp + 8], 0
0x18740:    mov     r14, rax
0x18743:    jne     0x186a8
0x18749:    mov     qword ptr [rsp + 8], rax
0x1874e:    jmp     0x186b0
0x186b0:
0x18753:    nop     dword ptr [rax + rax]
0x18758:    mov     rdx, qword ptr [rsp + 8]
0x1875d:    test    rdx, rdx
0x18760:    je      0x1877d
0x18762:    mov     rax, qword ptr [rip + 0x13327]
0x18769:    test    rax, rax
0x1876c:    je      0x187a8
0x1876e:    mov     rcx, qword ptr [rsp]
0x18772:    mov     qword ptr [rip + 0x13317], rdx
0x18779:    mov     qword ptr [rcx + 0x18], rax
0x1877d:    mov     rax, qword ptr [rsp + 0x88]
0x18785:    sub     rax, qword ptr fs:[0x28]
0x1878e:    jne     0x187b6
0x18790:    add     rsp, 0x98
0x18797:    pop     rbx
0x18798:    pop     rbp
0x18799:    pop     r12
0x1879b:    pop     r13
0x1879d:    pop     r14
0x1879f:    pop     r15
0x187a1:    ret

```

```

0x187a2:      nop      word ptr [rax + rax]
0x187a8:      mov      rax, qword ptr [rsp + 8]
0x187ad:      mov      qword ptr [rip + 0x132dc], rax
0x187b4:      jmp      0x1877d
0x1877d:
0x187b6:      call     0x49d0
0x187bb:      nop      dword ptr [rax + rax]
0x187c0:      endbr64
0x187c4:      sub      rsp, 8
0x187c8:      lea      rsi, [rip - 0xf]
0x187cf:      call     0x4b50
0x187d4:      mov      dword ptr [rip + 0x132d2], 1
0x187de:      add      rsp, 8
0x187e2:      ret
0x187e3:      nop      word ptr cs:[rax + rax]
0x187ee:      nop
0x187f0:      cmp      edi, 0x14
0x187f3:      je       0x18850
0x187f5:      jg       0x18818
0x187f7:      cmp      edi, 0xe
0x187fa:      je       0x18838
0x187fc:      cmp      edi, 0x12
0x187ff:      jne      0x18861
0x18801:      lea      rsi, [rip + 0x148]
0x18808:      mov      edi, 0x12
0x1880d:      jmp      0x4b50
0x4b50:
0x18812:      nop      word ptr [rax + rax]
0x18818:      cmp      edi, 0x1c
0x1881b:      jne      0x18830
0x1881d:      lea      rsi, [rip + 0x1cc]
0x18824:      mov      edi, 0x1c
0x18829:      jmp      0x4b50
0x4b50:
0x1882e:      nop
0x18830:      ret
0x18831:      nop      dword ptr [rax]
0x18838:      lea      rsi, [rip + 0x31]
0x1883f:      mov      edi, 0xe
0x18844:      jmp      0x4b50
0x4b50:
0x18849:      nop      dword ptr [rax]
0x18850:      lea      rsi, [rip + 0x89]
0x18857:      mov      edi, 0x14
0x1885c:      jmp      0x4b50
0x4b50:
0x18861:      ret
0x18862:      nop      word ptr cs:[rax + rax]

```



```

0x1886d:      nop      dword ptr [rax]
0x18870:      endbr64
0x18874:      mov      edx, dword ptr [rip + 0x1323a]
0x1887a:      push     rbp
0x1887b:      mov      eax, 1
0x18880:      mov      ebp, edi
0x18882:      test     edx, edx
0x18884:      jne      0x188b0
0x18886:      mov      edi, ebp
0x18888:      mov      dword ptr [rip + 0x1322e], eax
0x1888e:      call     0x187f0
0x18893:      xor      eax, eax
0x18895:      call     0x18330
0x1889a:      mov      edi, 0x3c
0x1889f:      pop      rbp
0x188a0:      sub      edi, dword ptr [rax]
0x188a2:      jmp      0x4aa0
0x4aa0
0x188a7:      nop      word ptr [rax + rax]
0x188b0:      xor      eax, eax
0x188b2:      call     0x18330
0x188b7:      mov      eax, dword ptr [rax]
0x188b9:      test     eax, eax
0x188bb:      je       0x188d0
0x188bd:      xor      eax, eax
0x188bf:      call     0xb4e0
0x188c4:      xor      eax, eax
0x188c6:      jmp      0x18886
0x18886
0x188c8:      nop      dword ptr [rax + rax]
0x188d0:      mov      edi, ebp
0x188d2:      call     0x16fd0
0x188d7:      jmp      0x188bd
0x188bd
0x188d9:      nop      dword ptr [rax]
0x188e0:      endbr64
0x188e4:      mov      eax, dword ptr [rip + 0x131ca]
0x188ea:      push     rbp
0x188eb:      mov      ebp, edi
0x188ed:      test     eax, eax
0x188ef:      jne      0x18908
0x188f1:      mov      eax, 1
0x188f6:      mov      edi, ebp
0x188f8:      pop      rbp
0x188f9:      mov      dword ptr [rip + 0x131c5], eax
0x188ff:      jmp      0x187f0
0x187f0
0x18904:      nop      dword ptr [rax]

```

```

0x18908:      xor     eax, eax
0x1890a:      call    0x9c90
0x1890f:      cmp     eax, 1
0x18912:      sete    al
0x18915:      movzx   eax, al
0x18918:      mov     dword ptr [rip + 0x1319a], eax
0x1891e:      xor     eax, eax
0x18920:      call    0x9610
0x18925:      call    0x4920
0x1892a:      mov     esi, 0x13
0x1892f:      mov     edi, eax
0x18931:      call    0x4bd0
0x18936:      xor     eax, eax
0x18938:      mov     edi, ebp
0x1893a:      pop     rbp
0x1893b:      mov     dword ptr [rip + 0x13183], eax
0x18941:      jmp     0x187f0
0x187f0
0x18946:      nop     word ptr cs:[rax + rax]
0x18950:      endbr64
0x18954:      mov     eax, dword ptr [rip + 0x1315e]
0x1895a:      push    rbx
0x1895b:      mov     ebx, edi
0x1895d:      test    eax, eax
0x1895f:      jne     0x189a0
0x18961:      cmp     ebx, 0x14
0x18964:      je      0x189b4
0x18966:      jg      0x18988
0x18968:      cmp     ebx, 0xe
0x1896b:      je      0x189d8
0x1896d:      cmp     ebx, 0x12
0x18970:      jne     0x189d0
0x18972:      lea     rsi, [rip - 0x29]
0x18979:      mov     edi, 0x12
0x1897e:      pop     rbx
0x1897f:      jmp     0x4b50
0x4b50
0x18984:      nop     dword ptr [rax]
0x18988:      cmp     ebx, 0x1c
0x1898b:      jne     0x189d0
0x1898d:      lea     rsi, [rip + 0x5c]
0x18994:      mov     edi, 0x1c
0x18999:      pop     rbx
0x1899a:      jmp     0x4b50
0x4b50
0x1899f:      nop
0x189a0:      call    0x7bd0
0x189a5:      mov     dword ptr [rip + 0x13111], 0

```

```

0x189af:      cmp     ebx, 0x14
0x189b2:      jne     0x18966
0x189b4:      lea     rsi, [rip - 0xdb]
0x189bb:      mov     edi, 0x14
0x189c0:      pop     rbx
0x189c1:      jmp     0x4b50
0x4b50
0x189c6:      nop     word ptr cs:[rax + rax]
0x189d0:      pop     rbx
0x189d1:      ret
0x189d2:      nop     word ptr [rax + rax]
0x189d8:      lea     rsi, [rip - 0x16f]
0x189df:      mov     edi, 0xe
0x189e4:      pop     rbx
0x189e5:      jmp     0x4b50
0x4b50
0x189ea:      nop     word ptr [rax + rax]
0x189f0:      endbr64
0x189f4:      mov     eax, dword ptr [rip + 0x130ba]
0x189fa:      push    rbx
0x189fb:      mov     ebx, edi
0x189fd:      test    eax, eax
0x189ff:      jne     0x18a50
0x18a01:      mov     eax, 1
0x18a06:      mov     dword ptr [rip + 0x130b4], eax
0x18a0c:      cmp     ebx, 0x14
0x18a0f:      je      0x18a72
0x18a11:      jg      0x18a30
0x18a13:      cmp     ebx, 0xe
0x18a16:      je      0x18a90
0x18a18:      cmp     ebx, 0x12
0x18a1b:      jne     0x18a88
0x18a1d:      lea     rsi, [rip - 0xd4]
0x18a24:      mov     edi, 0x12
0x18a29:      pop     rbx
0x18a2a:      jmp     0x4b50
0x4b50
0x18a2f:      nop
0x18a30:      cmp     ebx, 0x1c
0x18a33:      jne     0x18a88
0x18a35:      lea     rsi, [rip - 0x4c]
0x18a3c:      mov     edi, 0x1c
0x18a41:      pop     rbx
0x18a42:      jmp     0x4b50
0x4b50
0x18a47:      nop     word ptr [rax + rax]
0x18a50:      xor     eax, eax
0x18a52:      call    0xa6c0

```

```

0x18a57:    xor     eax, eax
0x18a59:    call    0x9d80
0x18a5e:    mov     edi, ebx
0x18a60:    call    0x7bd0
0x18a65:    xor     eax, eax
0x18a67:    mov     dword ptr [rip + 0x13053], eax
0x18a6d:    cmp     ebx, 0x14
0x18a70:    jne     0x18a11
0x18a72:    lea     rsi, [rip - 0x199]
0x18a79:    mov     edi, 0x14
0x18a7e:    pop     rbx
0x18a7f:    jmp     0x4b50
0x4b50:
0x18a84:    nop     dword ptr [rax]
0x18a88:    pop     rbx
0x18a89:    ret
0x18a8a:    nop     word ptr [rax + rax]
0x18a90:    lea     rsi, [rip - 0x227]
0x18a97:    mov     edi, 0xe
0x18a9c:    pop     rbx
0x18a9d:    jmp     0x4b50
0x4b50:
0x18aa2:    nop     word ptr cs:[rax + rax]
0x18aad:    nop     dword ptr [rax]
0x18ab0:    endbr64
0x18ab4:    sub     rsp, 8
0x18ab8:    mov     ecx, dword ptr [rip + 0x13006]
0x18abe:    test    ecx, ecx
0x18ac0:    jne     0x18af0
0x18ac2:    mov     edx, dword ptr [rip + 0x12ff8]
0x18ac8:    test    edx, edx
0x18aca:    jne     0x18b30
0x18acc:    mov     eax, dword ptr [rip + 0x12fea]
0x18ad2:    test    eax, eax
0x18ad4:    jne     0x18ae0
0x18ad6:    add     rsp, 8
0x18ada:    ret
0x18adb:    nop     dword ptr [rax + rax]
0x18ae0:    mov     dword ptr [rip + 0x12fd2], 0
0x18aea:    add     rsp, 8
0x18aee:    ret
0x18aef:    nop
0x18af0:    xor     eax, eax
0x18af2:    call    0x9610
0x18af7:    call    0x4920
0x18afc:    mov     esi, 0x13
0x18b01:    mov     edi, eax
0x18b03:    call    0x4bd0

```

```

0x18b08:    mov     dword ptr [rip + 0x12fb2], 0
0x18b12:    mov     dword ptr [rip + 0x12fa4], 0
0x18b1c:    mov     dword ptr [rip + 0x12f96], 0
0x18b26:    add     rsp, 8
0x18b2a:    ret
0x18b2b:    nop     dword ptr [rax + rax]
0x18b30:    xor     eax, eax
0x18b32:    call    0xa6c0
0x18b37:    xor     eax, eax
0x18b39:    call    0x9d80
0x18b3e:    mov     edi, 0x1c
0x18b43:    call    0x7bd0
0x18b48:    mov     dword ptr [rip + 0x12f6e], 0
0x18b52:    mov     dword ptr [rip + 0x12f60], 0
0x18b5c:    add     rsp, 8
0x18b60:    ret
0x18b61:    nop     word ptr cs:[rax + rax]
0x18b6c:    nop     dword ptr [rax]
0x18b70:    endbr64
0x18b74:    mov     dword ptr [rip + 0x12f3a], edi
0x18b7a:    test    edi, edi
0x18b7c:    jne     0x18b80
0x18b7e:    ret
0x18b7f:    nop
0x18b80:    xor     eax, eax
0x18b82:    jmp     0x18ab0
0x18ab0:
0x18b87:    nop     word ptr [rax + rax]
0x18b90:    endbr64
0x18b94:    sub     rsp, 8
0x18b98:    mov     eax, dword ptr [rip + 0x11bfa]
0x18b9e:    cmp     edi, 1
0x18ba1:    je      0x18bf0
0x18ba3:    test    eax, eax
0x18ba5:    jne     0x18bd0
0x18ba7:    mov     esi, 1
0x18bac:    mov     edi, 0x1c
0x18bb1:    call    0x4b50
0x18bb6:    mov     esi, 1
0x18bbb:    mov     edi, 0xe
0x18bc0:    add     rsp, 8
0x18bc4:    jmp     0x4b50
0x4b50:
0x18bc9:    nop     dword ptr [rax]
0x18bd0:    mov     esi, 1
0x18bd5:    mov     edi, 0x14
0x18bda:    call    0x4b50
0x18bdf:    mov     esi, 1

```

```

0x18be4:    mov     edi, 0x12
0x18be9:    call   0x4b50
0x18bee:    jmp     0x18ba7
0x18ba7:
0x18bf0:    test    eax, eax
0x18bf2:    jne     0x18c20
0x18bf4:    lea     rsi, [rip - 0x20b]
0x18bfb:    mov     edi, 0x1c
0x18c00:    call   0x4b50
0x18c05:    lea     rsi, [rip - 0x39c]
0x18c0c:    mov     edi, 0xe
0x18c11:    add     rsp, 8
0x18c15:    jmp     0x4b50
0x4b50:
0x18c1a:    nop     word ptr [rax + rax]
0x18c20:    lea     rsi, [rip - 0x347]
0x18c27:    mov     edi, 0x14
0x18c2c:    call   0x4b50
0x18c31:    lea     rsi, [rip - 0x2e8]
0x18c38:    mov     edi, 0x12
0x18c3d:    call   0x4b50
0x18c42:    jmp     0x18bf4
0x18bf4:
0x18c44:    nop     word ptr cs:[rax + rax]
0x18c4f:    nop
0x18c50:    endbr64
0x18c54:    push    rbp
0x18c55:    mov     esi, 1
0x18c5a:    mov     edi, 0x14
0x18c5f:    call   0x4b50
0x18c64:    mov     esi, 1
0x18c69:    mov     edi, 0x12
0x18c6e:    cmp     rax, 1
0x18c72:    setne   al
0x18c75:    movzx   eax, al
0x18c78:    mov     dword ptr [rip + 0x11b1a], eax
0x18c7e:    call   0x4b50
0x18c83:    mov     esi, 1
0x18c88:    mov     edi, 0x1c
0x18c8d:    call   0x4b50
0x18c92:    lea     rbp, [rip - 0x1519]
0x18c99:    mov     edi, 0xb
0x18c9e:    mov     rsi, rbp
0x18ca1:    call   0x4b50
0x18ca6:    mov     rsi, rbp
0x18ca9:    mov     edi, 1
0x18cae:    call   0x4b50
0x18cb3:    mov     rsi, rbp

```

```

0x18cb6:      mov     edi, 0xf
0x18cbb:      call    0x4b50
0x18cc0:      mov     esi, 1
0x18cc5:      mov     edi, 3
0x18cca:      call    0x4b50
0x18ccf:      lea     rsi, [rip - 0x516]
0x18cd6:      mov     edi, 2
0x18cdb:      pop     rbp
0x18cdc:      jmp     0x4b50
0x4b50
0x18ce1:      nop     word ptr cs:[rax + rax]
0x18ceb:      nop     dword ptr [rax + rax]
0x18cf0:      endbr64
0x18cf4:      push    rax
0x18cf5:      pop     rax
0x18cf6:      lea     rdi, [rip + 0x96a8]
0x18cfd:      sub     rsp, 8
0x18d01:      call    0x8b00
0x18d06:      call    0x4850
0x18d0b:      nop     dword ptr [rax + rax]
0x18d10:      endbr64
0x18d14:      test    rdi, rdi
0x18d17:      je      0x18d20
0x18d19:      jmp     0x4800
0x4800
0x18d1e:      nop
0x18d20:      lea     rdi, [rip + 0x9697]
0x18d27:      jmp     0x8b00
0x8b00
0x18d2c:      nop     dword ptr [rax]
0x18d30:      endbr64
0x18d34:      xor     r8d, r8d
0x18d37:      test    rdx, rdx
0x18d3a:      je      0x18d90
0x18d3c:      push    r14
0x18d3e:      push    r13
0x18d40:      push    r12
0x18d42:      mov     r12, rsi
0x18d45:      push    rbp
0x18d46:      mov     ebp, edi
0x18d48:      push    rbx
0x18d49:      mov     rbx, rdx
0x18d4c:      call    0x4860
0x18d51:      mov     r13d, dword ptr [rax]
0x18d54:      mov     r14, rax
0x18d57:      jmp     0x18d66
0x18d66
0x18d59:      nop     dword ptr [rax]

```

```

0x18d60:    cmp     dword ptr [r14], 4
0x18d64:    jne     0x18d7d
0x18d66:    mov     dword ptr [r14], r13d
0x18d69:    mov     rdx, rbx
0x18d6c:    mov     rsi, r12
0x18d6f:    mov     edi, ebp
0x18d71:    call    0x47a0
0x18d76:    mov     r8d, eax
0x18d79:    test    eax, eax
0x18d7b:    js      0x18d60
0x18d7d:    pop     rbx
0x18d7e:    mov     eax, r8d
0x18d81:    pop     rbp
0x18d82:    pop     r12
0x18d84:    pop     r13
0x18d86:    pop     r14
0x18d88:    ret
0x18d89:    nop     dword ptr [rax]
0x18d90:    mov     eax, r8d
0x18d93:    ret
0x18d94:    nop     word ptr cs:[rax + rax]
0x18d9f:    nop
0x18da0:    endbr64
0x18da4:    xor     r8d, r8d
0x18da7:    test    rdx, rdx
0x18daa:    je      0x18e00
0x18dac:    push    r14
0x18dae:    push    r13
0x18db0:    push    r12
0x18db2:    mov     r12, rsi
0x18db5:    push    rbp
0x18db6:    mov     ebp, edi
0x18db8:    push    rbx
0x18db9:    mov     rbx, rdx
0x18dbc:    call    0x4860
0x18dc1:    mov     r13d, dword ptr [rax]
0x18dc4:    mov     r14, rax
0x18dc7:    jmp     0x18dd6
0x18dd6:
0x18dc9:    nop     dword ptr [rax]
0x18dd0:    cmp     dword ptr [r14], 4
0x18dd4:    jne     0x18ded
0x18dd6:    mov     dword ptr [r14], r13d
0x18dd9:    mov     rdx, rbx
0x18ddc:    mov     rsi, r12
0x18ddf:    mov     edi, ebp
0x18de1:    call    0x4790
0x18de6:    mov     r8d, eax

```



```

0x18de9:      test    eax, eax
0x18deb:      js      0x18dd0
0x18ded:      pop     rbx
0x18dee:      mov     eax, r8d
0x18df1:      pop     rbp
0x18df2:      pop     r12
0x18df4:      pop     r13
0x18df6:      pop     r14
0x18df8:      ret
0x18df9:      nop     dword ptr [rax]
0x18e00:      mov     eax, r8d
0x18e03:      ret
0x18e04:      nop     word ptr cs:[rax + rax]
0x18e0f:      nop
0x18e10:      endbr64
0x18e14:      xor     r8d, r8d
0x18e17:      test    rdx, rdx
0x18e1a:      je      0x18e70
0x18e1c:      push    r14
0x18e1e:      push    r13
0x18e20:      push    r12
0x18e22:      mov     r12, rsi
0x18e25:      push    rbp
0x18e26:      mov     rbp, rdi
0x18e29:      push    rbx
0x18e2a:      mov     rbx, rdx
0x18e2d:      call    0x4860
0x18e32:      mov     r13d, dword ptr [rax]
0x18e35:      mov     r14, rax
0x18e38:      jmp     0x18e46
0x18e46:
0x18e3a:      nop     word ptr [rax + rax]
0x18e40:      cmp     dword ptr [r14], 4
0x18e44:      jne     0x18e5e
0x18e46:      mov     dword ptr [r14], r13d
0x18e49:      mov     rdx, rbx
0x18e4c:      mov     rsi, r12
0x18e4f:      mov     rdi, rbp
0x18e52:      call    0x4910
0x18e57:      mov     r8d, eax
0x18e5a:      test    eax, eax
0x18e5c:      js      0x18e40
0x18e5e:      pop     rbx
0x18e5f:      mov     eax, r8d
0x18e62:      pop     rbp
0x18e63:      pop     r12
0x18e65:      pop     r13
0x18e67:      pop     r14

```

```

0x18e69:    ret
0x18e6a:    nop      word ptr [rax + rax]
0x18e70:    mov      eax, r8d
0x18e73:    ret
0x18e74:    nop      word ptr cs:[rax + rax]
0x18e7f:    nop
0x18e80:    endbr64
0x18e84:    push     r13
0x18e86:    mov      r13, rdi
0x18e89:    push     r12
0x18e8b:    push     rbp
0x18e8c:    push     rbx
0x18e8d:    mov      ebx, 0x64
0x18e92:    sub      rsp, 0x18
0x18e96:    jmp      0x18eaa
0x18eaa:
0x18e98:    nop      dword ptr [rax + rax]
0x18ea0:    mov      rdi, rbp
0x18ea3:    add      ebx, ebx
0x18ea5:    call     0x18d10
0x18eaa:    movsxd   r12, ebx
0x18ead:    mov      rdi, r12
0x18eb0:    call     0x1d670
0x18eb5:    mov      rdx, r12
0x18eb8:    mov      rdi, r13
0x18ebb:    mov      rbp, rax
0x18ebe:    mov      rsi, rax
0x18ec1:    xor      eax, eax
0x18ec3:    call     0x18e10
0x18ec8:    cmp      ebx, eax
0x18eca:    jle      0x18ea0
0x18ecc:    mov      rdi, rbp
0x18ecf:    mov      dword ptr [rsp + 0xc], eax
0x18ed3:    call     0x18d10
0x18ed8:    mov      eax, dword ptr [rsp + 0xc]
0x18edc:    add      rsp, 0x18
0x18ee0:    pop      rbx
0x18ee1:    pop      rbp
0x18ee2:    pop      r12
0x18ee4:    pop      r13
0x18ee6:    ret
0x18ee7:    nop      word ptr [rax + rax]
0x18ef0:    endbr64
0x18ef4:    push     r13
0x18ef6:    push     r12
0x18ef8:    mov      r12d, edi
0x18efb:    push     rbp
0x18efc:    mov      rbp, rsi

```

```

0x18eff:    push    rbx
0x18f00:    sub     rsp, 8
0x18f04:    call    0x4860
0x18f09:    mov     r13d, dword ptr [rax]
0x18f0c:    mov     rbx, rax
0x18f0f:    jmp     0x18f1d
0x18f1d:
0x18f11:    nop     dword ptr [rax]
0x18f18:    cmp     dword ptr [rbx], 4
0x18f1b:    jne     0x18f2f
0x18f1d:    mov     dword ptr [rbx], r13d
0x18f20:    mov     rsi, rbp
0x18f23:    mov     edi, r12d
0x18f26:    call    0x4eb0
0x18f2b:    test    eax, eax
0x18f2d:    js      0x18f18
0x18f2f:    add     rsp, 8
0x18f33:    pop     rbx
0x18f34:    pop     rbp
0x18f35:    pop     r12
0x18f37:    pop     r13
0x18f39:    ret
0x18f3a:    nop     word ptr [rax + rax]
0x18f40:    endbr64
0x18f44:    push    r13
0x18f46:    push    r12
0x18f48:    mov     r12, rdi
0x18f4b:    push    rbp
0x18f4c:    mov     rbp, rsi
0x18f4f:    push    rbx
0x18f50:    sub     rsp, 8
0x18f54:    call    0x4860
0x18f59:    mov     r13d, dword ptr [rax]
0x18f5c:    mov     rbx, rax
0x18f5f:    jmp     0x18f6d
0x18f6d:
0x18f61:    nop     dword ptr [rax]
0x18f68:    cmp     dword ptr [rbx], 4
0x18f6b:    jne     0x18f7f
0x18f6d:    mov     dword ptr [rbx], r13d
0x18f70:    mov     rsi, rbp
0x18f73:    mov     rdi, r12
0x18f76:    call    0x4b90
0x18f7b:    test    eax, eax
0x18f7d:    js      0x18f68
0x18f7f:    add     rsp, 8
0x18f83:    pop     rbx
0x18f84:    pop     rbp

```

```

0x18f85:    pop     r12
0x18f87:    pop     r13
0x18f89:    ret
0x18f8a:    nop     word ptr [rax + rax]
0x18f90:    endbr64
0x18f94:    push    r13
0x18f96:    push    r12
0x18f98:    mov     r12, rdi
0x18f9b:    push    rbp
0x18f9c:    mov     rbp, rsi
0x18f9f:    push    rbx
0x18fa0:    sub     rsp, 8
0x18fa4:    call    0x4860
0x18fa9:    mov     r13d, dword ptr [rax]
0x18fac:    mov     rbx, rax
0x18faf:    jmp     0x18fbd
0x18fbd
0x18fb1:    nop     dword ptr [rax]
0x18fb8:    cmp     dword ptr [rbx], 4
0x18fbb:    jne     0x18fcf
0x18fbd:    mov     dword ptr [rbx], r13d
0x18fc0:    mov     rsi, rbp
0x18fc3:    mov     rdi, r12
0x18fc6:    call    0x4ad0
0x18fcb:    test    eax, eax
0x18fcd:    js      0x18fb8
0x18fcf:    add     rsp, 8
0x18fd3:    pop     rbx
0x18fd4:    pop     rbp
0x18fd5:    pop     r12
0x18fd7:    pop     r13
0x18fd9:    ret
0x18fda:    nop     word ptr [rax + rax]
0x18fe0:    endbr64
0x18fe4:    push    r12
0x18fe6:    push    rbp
0x18fe7:    push    rbx
0x18fe8:    mov     ebx, 0x40
0x18fed:    call    0x4860
0x18ff2:    mov     edi, 0x40
0x18ff7:    mov     dword ptr [rax], 0
0x18ffd:    mov     r12, rax
0x19000:    call    0x1d670
0x19005:    mov     rbp, rax
0x19008:    jmp     0x19031
0x19031
0x1900a:    nop     word ptr [rax + rax]
0x19010:    cmp     dword ptr [r12], 0x22

```

```

0x19015:      jne      0x19050
0x19017:      add      rbx, 0x40
0x1901b:      mov      rdi, rbp
0x1901e:      mov      rsi, rbx
0x19021:      call     0x1d690
0x19026:      mov      dword ptr [r12], 0
0x1902e:      mov      rbp, rax
0x19031:      mov      rsi, rbx
0x19034:      mov      rdi, rbp
0x19037:      call     0x4a80
0x1903c:      test     rax, rax
0x1903f:      je       0x19010
0x19041:      mov      rax, rbp
0x19044:      pop      rbx
0x19045:      pop      rbp
0x19046:      pop      r12
0x19048:      ret
0x19049:      nop      dword ptr [rax]
0x19050:      mov      rdi, rbp
0x19053:      xor      ebp, ebp
0x19055:      call     0x18d10
0x1905a:      mov      rax, rbp
0x1905d:      pop      rbx
0x1905e:      pop      rbp
0x1905f:      pop      r12
0x19061:      ret
0x19062:      nop      word ptr cs:[rax + rax]
0x1906d:      nop      dword ptr [rax]
0x19070:      endbr64
0x19074:      push     rbp
0x19075:      mov      esi, 0x2f
0x1907a:      mov      rbp, rdi
0x1907d:      push     rbx
0x1907e:      sub      rsp, 8
0x19082:      call     0x4a20
0x19087:      test     rax, rax
0x1908a:      je       0x190a8
0x1908c:      mov      rbx, rax
0x1908f:      mov      rdi, rbp
0x19092:      call     0x1d7a0
0x19097:      sub      rbx, rbp
0x1909a:      mov      byte ptr [rax + rbx], 0x30
0x1909e:      add      rsp, 8
0x190a2:      pop      rbx
0x190a3:      pop      rbp
0x190a4:      ret
0x190a5:      nop      dword ptr [rax]
0x190a8:      add      rsp, 8

```

```

0x190ac:    lea     rdi, [rip + 0x862b]
0x190b3:    pop     rbx
0x190b4:    pop     rbp
0x190b5:    jmp     0x1d7a0
0x1d7a0
0x190ba:    nop     word ptr [rax + rax]
0x190c0:    endbr64
0x190c4:    push    rbx
0x190c5:    mov     rbx, rdi
0x190c8:    call    0x49b0
0x190cd:    lea     rax, [rbx + rax - 1]
0x190d2:    cmp     byte ptr [rax], 0x2f
0x190d5:    jne     0x190da
0x190d7:    mov     byte ptr [rax], 0
0x190da:    mov     rdi, rbx
0x190dd:    mov     esi, 0x2f
0x190e2:    call    0x4a20
0x190e7:    lea     rdx, [rax + 1]
0x190eb:    test    rax, rax
0x190ee:    mov     rax, rbx
0x190f1:    pop     rbx
0x190f2:    cmovne  rax, rdx
0x190f6:    ret
0x190f7:    nop     word ptr [rax + rax]
0x19100:    endbr64
0x19104:    push    rbx
0x19105:    mov     esi, 0x10
0x1910a:    mov     ebx, edi
0x1910c:    mov     edi, 1
0x19111:    call    0x1d740
0x19116:    mov     dword ptr [rax + 8], ebx
0x19119:    pop     rbx
0x1911a:    ret
0x1911b:    nop     dword ptr [rax + rax]
0x19120:    endbr64
0x19124:    test    rdi, rdi
0x19127:    je      0x19148
0x19129:    push    rbp
0x1912a:    mov     rbp, rdi
0x1912d:    mov     rdi, qword ptr [rdi]
0x19130:    test    rdi, rdi
0x19133:    je      0x1913a
0x19135:    call    0x18d10
0x1913a:    mov     rdi, rbp
0x1913d:    pop     rbp
0x1913e:    jmp     0x18d10
0x18d10
0x19143:    nop     dword ptr [rax + rax]

```

```

0x19148:      ret
0x19149:      nop        dword ptr [rax]
0x19150:      endbr64
0x19154:      push     rbp
0x19155:      mov      rbp, rsi
0x19158:      push     rbx
0x19159:      mov      rbx, rdi
0x1915c:      sub      rsp, 8
0x19160:      mov      eax, dword ptr [rdi + 0xc]
0x19163:      lea      esi, [rax + 1]
0x19166:      mov      dword ptr [rdi + 0xc], esi
0x19169:      imul     esi, dword ptr [rdi + 8]
0x1916d:      mov      rdi, qword ptr [rdi]
0x19170:      movsxd   rsi, esi
0x19173:      call     0x1d690
0x19178:      mov      ecx, dword ptr [rbx + 0xc]
0x1917b:      movsxd   rdx, dword ptr [rbx + 8]
0x1917f:      mov      rsi, rbp
0x19182:      mov      qword ptr [rbx], rax
0x19185:      add      rsp, 8
0x19189:      sub      ecx, 1
0x1918c:      pop      rbx
0x1918d:      pop      rbp
0x1918e:      imul     ecx, edx
0x19191:      movsxd   rcx, ecx
0x19194:      lea      rdi, [rax + rcx]
0x19198:      jmp      0x4bb0
0x4bb0
0x1919d:      nop        dword ptr [rax]
0x191a0:      endbr64
0x191a4:      mov      eax, dword ptr [rdi + 0xc]
0x191a7:      test     eax, eax
0x191a9:      je       0x191f8
0x191ab:      push     rbp
0x191ac:      sub      eax, 1
0x191af:      mov      rbp, rsi
0x191b2:      push     rbx
0x191b3:      mov      rbx, rdi
0x191b6:      sub      rsp, 8
0x191ba:      movsxd   rdx, dword ptr [rdi + 8]
0x191be:      mov      dword ptr [rdi + 0xc], eax
0x191c1:      mov      rsi, rdx
0x191c4:      imul     esi, eax
0x191c7:      movsxd   rsi, esi
0x191ca:      add      rsi, qword ptr [rdi]
0x191cd:      mov      rdi, rbp
0x191d0:      call     0x4bb0
0x191d5:      mov      esi, dword ptr [rbx + 0xc]

```

```

0x191d8:    imul    esi, dword ptr [rbx + 8]
0x191dc:    mov     rdi, qword ptr [rbx]
0x191df:    movsxd  rsi, esi
0x191e2:    call    0x1d690
0x191e7:    mov     qword ptr [rbx], rax
0x191ea:    add     rsp, 8
0x191ee:    mov     rax, rbp
0x191f1:    pop     rbx
0x191f2:    pop     rbp
0x191f3:    ret
0x191f4:    nop     dword ptr [rax]
0x191f8:    xor     eax, eax
0x191fa:    ret
0x191fb:    nop     dword ptr [rax + rax]
0x19200:    endbr64
0x19204:    mov     ecx, dword ptr [rdi + 0xc]
0x19207:    cmp     ecx, edx
0x19209:    jl      0x19240
0x1920b:    test    ecx, ecx
0x1920d:    je      0x19240
0x1920f:    sub     rsp, 8
0x19213:    movsxd  r9, dword ptr [rdi + 8]
0x19217:    sub     ecx, edx
0x19219:    mov     r8, rsi
0x1921c:    imul    ecx, r9d
0x19220:    mov     rdx, r9
0x19223:    movsxd  rsi, ecx
0x19226:    add     rsi, qword ptr [rdi]
0x19229:    mov     rdi, r8
0x1922c:    call    0x4bb0
0x19231:    add     rsp, 8
0x19235:    ret
0x19236:    nop     word ptr cs:[rax + rax]
0x19240:    xor     eax, eax
0x19242:    ret
0x19243:    nop     word ptr cs:[rax + rax]
0x1924e:    nop
0x19250:    endbr64
0x19254:    push    rbx
0x19255:    mov     rbx, rdi
0x19258:    mov     dword ptr [rdi + 0xc], esi
0x1925b:    imul    esi, dword ptr [rdi + 8]
0x1925f:    mov     rdi, qword ptr [rdi]
0x19262:    movsxd  rsi, esi
0x19265:    call    0x1d690
0x1926a:    mov     qword ptr [rbx], rax
0x1926d:    pop     rbx
0x1926e:    ret

```



```

0x1926f:      nop
0x19270:      endbr64
0x19274:      mov     eax, dword ptr [rdi + 0xc]
0x19277:      ret
0x19278:      nop     dword ptr [rax + rax]
0x19280:      push    r12
0x19282:      mov     r12, rdi
0x19285:      mov     edi, 2
0x1928a:      push    rbp
0x1928b:      push    rbx
0x1928c:      mov     ebx, esi
0x1928e:      xor     esi, esi
0x19290:      call    0x4b50
0x19295:      xor     esi, esi
0x19297:      mov     edi, 3
0x1929c:      call    0x4b50
0x192a1:      test    ebx, ebx
0x192a3:      jne     0x192e0
0x192a5:      mov     r9, qword ptr [rip + 0x115b4]
0x192ac:      xor     r8d, r8d
0x192af:      mov     rcx, r12
0x192b2:      xor     eax, eax
0x192b4:      lea     rdi, [rip + 0x7363]
0x192bb:      lea     rdx, [rip + 0x911e]
0x192c2:      lea     rsi, [rip + 0x735a]
0x192c9:      call    0x4e30
0x192ce:      mov     edi, 0x7f
0x192d3:      call    0x48a0
0x192d8:      nop     dword ptr [rax + rax]
0x192e0:      mov     edi, 1
0x192e5:      lea     rbp, [rip + 0x90ed]
0x192ec:      call    0x4ab0
0x192f1:      mov     edi, 2
0x192f6:      call    0x4ab0
0x192fb:      mov     rdi, qword ptr [rip + 0x127ce]
0x19302:      mov     rsi, rbp
0x19305:      call    0x4da0
0x1930a:      mov     rdi, qword ptr [rip + 0x127b7]
0x19311:      mov     rsi, rbp
0x19314:      call    0x4da0
0x19319:      mov     edx, 4
0x1931e:      lea     rsi, [rip + 0x90b6]
0x19325:      mov     rdi, r12
0x19328:      call    0x4890
0x1932d:      test    eax, eax
0x1932f:      je      0x192a5
0x19335:      xor     edi, edi
0x19337:      call    0x4ab0

```

```

0x1933c:      jmp      0x192a5
0x192a5
0x19341:      nop      word ptr cs:[rax + rax]
0x1934c:      nop      dword ptr [rax]
0x19350:      endbr64
0x19354:      push     r12
0x19356:      push     rbp
0x19357:      push     rbx
0x19358:      sub      rsp, 0x10
0x1935c:      mov      rax, qword ptr fs:[0x28]
0x19365:      mov      qword ptr [rsp + 8], rax
0x1936a:      xor      eax, eax
0x1936c:      test     rdi, rdi
0x1936f:      je        0x193d0
0x19371:      mov      rbp, rdi
0x19374:      mov      r12d, esi
0x19377:      call     0x4ed0
0x1937c:      mov      ebx, eax
0x1937e:      test     eax, eax
0x19380:      js        0x193c0
0x19382:      je        0x193dc
0x19384:      lea      rbp, [rsp + 4]
0x19389:      nop      dword ptr [rax]
0x19390:      mov      rdi, rbp
0x19393:      call     0x4ec0
0x19398:      cmp      eax, ebx
0x1939a:      jne      0x19390
0x1939c:      mov      eax, dword ptr [rsp + 4]
0x193a0:      mov      rdx, qword ptr [rsp + 8]
0x193a5:      sub      rdx, qword ptr fs:[0x28]
0x193ae:      jne      0x193d7
0x193b0:      add      rsp, 0x10
0x193b4:      pop      rbx
0x193b5:      pop      rbp
0x193b6:      pop      r12
0x193b8:      ret
0x193b9:      nop      dword ptr [rax]
0x193c0:      mov      eax, 0xffffffff
0x193c5:      jmp      0x193a0
0x193a0
0x193c7:      nop      word ptr [rax + rax]
0x193d0:      mov      eax, 1
0x193d5:      jmp      0x193a0
0x193a0
0x193d7:      call     0x49d0
0x193dc:      mov      esi, r12d
0x193df:      mov      rdi, rbp
0x193e2:      xor      eax, eax

```

```

0x193e4:      call    0x19280
0x193e9:      nop     dword ptr [rax]
0x193f0:      endbr64
0x193f4:      push    r13
0x193f6:      mov     r13, rdi
0x193f9:      push    r12
0x193fb:      push    rbp
0x193fc:      push    rbx
0x193fd:      sub     rsp, 0x18
0x19401:      mov     rax, qword ptr fs:[0x28]
0x1940a:      mov     qword ptr [rsp + 8], rax
0x1940f:      xor     eax, eax
0x19411:      test    esi, esi
0x19413:      je      0x19488
0x19415:      mov     edi, 1
0x1941a:      mov     ebp, esi
0x1941c:      call    0x18b70
0x19421:      test    r13, r13
0x19424:      je      0x19520
0x1942a:      call    0x4ed0
0x1942f:      mov     r12d, eax
0x19432:      test    eax, eax
0x19434:      js      0x19550
0x1943a:      je      0x19560
0x19440:      lea     rbx, [rsp + 4]
0x19445:      nop     dword ptr [rax]
0x19448:      mov     rdi, rbx
0x1944b:      call    0x4ec0
0x19450:      cmp     r12d, eax
0x19453:      jne     0x19448
0x19455:      mov     r12d, dword ptr [rsp + 4]
0x1945a:      xor     edi, edi
0x1945c:      call    0x18b70
0x19461:      mov     rax, qword ptr [rsp + 8]
0x19466:      sub     rax, qword ptr fs:[0x28]
0x1946f:      jne     0x1955b
0x19475:      add     rsp, 0x18
0x19479:      mov     eax, r12d
0x1947c:      pop     rbx
0x1947d:      pop     rbp
0x1947e:      pop     r12
0x19480:      pop     r13
0x19482:      ret
0x19483:      nop     dword ptr [rax + rax]
0x19488:      xor     edi, edi
0x1948a:      call    0x9b00
0x1948f:      xor     eax, eax
0x19491:      call    0xa6c0

```

```

0x19496:      mov     rdi, qword ptr [rip + 0x1142b]
0x1949d:      call   0xbb40
0x194a2:      xor     edi, edi
0x194a4:      call   0x18b90
0x194a9:      test    r13, r13
0x194ac:      je      0x19530
0x194b2:      call   0x4ed0
0x194b7:      mov     ebp, eax
0x194b9:      test    eax, eax
0x194bb:      js      0x19540
0x194c1:      je      0x1956c
0x194c7:      lea     rbx, [rsp + 4]
0x194cc:      nop     dword ptr [rax]
0x194d0:      mov     rdi, rbx
0x194d3:      call   0x4ec0
0x194d8:      cmp     ebp, eax
0x194da:      jne     0x194d0
0x194dc:      mov     r12d, dword ptr [rsp + 4]
0x194e1:      mov     edi, 1
0x194e6:      call   0x18b90
0x194eb:      mov     edx, 2
0x194f0:      mov     edi, 1
0x194f5:      lea     rsi, [rip + 0x8ee7]
0x194fc:      call   0x18da0
0x19501:      mov     edi, 1
0x19506:      call   0x9b00
0x1950b:      xor     eax, eax
0x1950d:      call   0xa6c0
0x19512:      xor     edi, edi
0x19514:      call   0x7890
0x19519:      jmp     0x19461
0x19461
0x1951e:      nop
0x19520:      mov     r12d, 1
0x19526:      jmp     0x1945a
0x1945a
0x1952b:      nop     dword ptr [rax + rax]
0x19530:      mov     r12d, 1
0x19536:      jmp     0x194e1
0x194e1
0x19538:      nop     dword ptr [rax + rax]
0x19540:      mov     r12d, 0xffffffff
0x19546:      jmp     0x194e1
0x194e1
0x19548:      nop     dword ptr [rax + rax]
0x19550:      mov     r12d, 0xffffffff
0x19556:      jmp     0x1945a
0x1945a

```

```

0x1955b:    call    0x49d0
0x19560:    mov     esi, ebp
0x19562:    mov     rdi, r13
0x19565:    xor     eax, eax
0x19567:    call    0x19280
0x1956c:    xor     esi, esi
0x1956e:    mov     rdi, r13
0x19571:    xor     eax, eax
0x19573:    call    0x19280
0x19578:    nop     dword ptr [rax + rax]
0x19580:    endbr64
0x19584:    mov     rdi, qword ptr [rip + 0x12545]
0x1958b:    test    rdi, rdi
0x1958e:    je      0x195b8
0x19590:    sub     rsp, 8
0x19594:    call    0x4870
0x19599:    mov     rdi, qword ptr [rip + 0x12528]
0x195a0:    test    rdi, rdi
0x195a3:    je      0x195b0
0x195a5:    add     rsp, 8
0x195a9:    jmp     0x4870
0x4870
0x195ae:    nop
0x195b0:    add     rsp, 8
0x195b4:    ret
0x195b5:    nop     dword ptr [rax]
0x195b8:    mov     rdi, qword ptr [rip + 0x12509]
0x195bf:    test    rdi, rdi
0x195c2:    je      0x195d0
0x195c4:    jmp     0x4870
0x4870
0x195c9:    nop     dword ptr [rax]
0x195d0:    ret
0x195d1:    nop     word ptr cs:[rax + rax]
0x195dc:    nop     dword ptr [rax]
0x195e0:    endbr64
0x195e4:    push    r12
0x195e6:    lea     rsi, [rip + 0x89dc]
0x195ed:    push    rbp
0x195ee:    mov     rbp, rdi
0x195f1:    sub     rsp, 8
0x195f5:    mov     rdi, qword ptr [rip + 0x124cc]
0x195fc:    call    0x4da0
0x19601:    test    rax, rax
0x19604:    je      0x19676
0x19606:    mov     edi, 0x801
0x1960b:    mov     r12, rax
0x1960e:    call    0x1d670

```

```

0x19613:      mov     rbp, rax
0x19616:      jmp     0x1964a
0x1964a:
0x19618:      nop     dword ptr [rax + rax]
0x19620:      mov     rdi, rbp
0x19623:      call    0x49b0
0x19628:      cdqe
0x1962a:      lea     rax, [rbp + rax - 1]
0x1962f:      cmp     byte ptr [rax], 0xa
0x19632:      jne     0x19637
0x19634:      mov     byte ptr [rax], 0
0x19637:      xor     esi, esi
0x19639:      mov     edx, 0xa
0x1963e:      mov     rdi, rbp
0x19641:      call    0x8070
0x19646:      test    al, al
0x19648:      je      0x1965f
0x1964a:      mov     rdx, r12
0x1964d:      mov     esi, 0x801
0x19652:      mov     rdi, rbp
0x19655:      call    0x4b00
0x1965a:      test    rax, rax
0x1965d:      jne     0x19620
0x1965f:      mov     rdi, rbp
0x19662:      call    0x18d10
0x19667:      add     rsp, 8
0x1966b:      mov     rdi, r12
0x1966e:      pop     rbp
0x1966f:      pop     r12
0x19671:      jmp     0x4940
0x4940:
0x19676:      mov     rdi, rbp
0x19679:      call    0x49b0
0x1967e:      mov     rdi, qword ptr [rip + 0x12443]
0x19685:      mov     r12, rax
0x19688:      call    0x49b0
0x1968d:      lea     rdi, [r12 + rax + 0x20]
0x19692:      call    0x1d670
0x19697:      mov     r9, qword ptr [rip + 0x1242a]
0x1969e:      mov     r8, rbp
0x196a1:      or      rdx, 0xffffffffffffffff
0x196a5:      mov     r12, rax
0x196a8:      mov     rdi, rax
0x196ab:      mov     esi, 1
0x196b0:      xor     eax, eax
0x196b2:      lea     rcx, [rip + 0x8d2d]
0x196b9:      call    0x4f10
0x196be:      mov     rdi, r12

```

```

0x196c1:    mov     edx, 0x1e
0x196c6:    xor     esi, esi
0x196c8:    call    0x8070
0x196cd:    pop     rax
0x196ce:    mov     rdi, r12
0x196d1:    pop     rbp
0x196d2:    pop     r12
0x196d4:    jmp     0x18d10
0x18d10
0x196d9:    nop     dword ptr [rax]
0x196e0:    endbr64
0x196e4:    push    r15
0x196e6:    push    r14
0x196e8:    push    r13
0x196ea:    push    r12
0x196ec:    push    rbp
0x196ed:    push    rbx
0x196ee:    sub     rsp, 8
0x196f2:    test    rdi, rdi
0x196f5:    je      0x19878
0x196fb:    mov     rbp, rdi
0x196fe:    call    0x49b0
0x19703:    lea     rdi, [rax + 1]
0x19707:    call    0x1d670
0x1970c:    mov     rsi, rbp
0x1970f:    mov     rdi, rax
0x19712:    call    0x48b0
0x19717:    cmp     byte ptr [rax], 0x7e
0x1971a:    mov     rbp, rax
0x1971d:    mov     r12, rax
0x19720:    jne     0x19800
0x19726:    movzx   eax, byte ptr [rax + 1]
0x1972a:    lea     r13, [rbp + 1]
0x1972e:    test    al, al
0x19730:    je      0x19818
0x19736:    cmp     al, 0x2f
0x19738:    je      0x19818
0x1973e:    mov     rdi, rbp
0x19741:    call    0x49b0
0x19746:    mov     rdi, rax
0x19749:    call    0x1d670
0x1974e:    movzx   ecx, byte ptr [rbp + 1]
0x19752:    mov     r14, rax
0x19755:    cmp     cl, 0x2f
0x19758:    je      0x19938
0x1975e:    test    cl, cl
0x19760:    je      0x19938
0x19766:    mov     edx, 2

```

```

0x1976b:      nop      dword ptr [rax + rax]
0x19770:      mov      byte ptr [r14 + rdx - 2], cl
0x19775:      movzx    ecx, byte ptr [rbp + rdx]
0x1977a:      lea      r13, [rbp + rdx]
0x1977f:      mov      rax, rdx
0x19782:      add      rdx, 1
0x19786:      test     cl, cl
0x19788:      je       0x1978f
0x1978a:      cmp      cl, 0x2f
0x1978d:      jne      0x19770
0x1978f:      lea      rax, [r14 + rax - 1]
0x19794:      mov      byte ptr [rax], 0
0x19797:      mov      rdi, r14
0x1979a:      call     0x4b60
0x1979f:      mov      rbx, rax
0x197a2:      test     rax, rax
0x197a5:      je       0x19880
0x197ab:      mov      rdi, qword ptr [rax + 0x20]
0x197af:      call     0x49b0
0x197b4:      mov      rdi, r13
0x197b7:      mov      r15, rax
0x197ba:      call     0x49b0
0x197bf:      lea      edi, [r15 + 1]
0x197c3:      movsxd   rdi, edi
0x197c6:      add      rdi, rax
0x197c9:      call     0x1d670
0x197ce:      mov      rsi, qword ptr [rbx + 0x20]
0x197d2:      mov      rdi, rax
0x197d5:      mov      r12, rax
0x197d8:      call     0x48b0
0x197dd:      movsxd   rdi, r15d
0x197e0:      mov      rsi, r13
0x197e3:      add      rdi, r12
0x197e6:      call     0x48b0
0x197eb:      mov      rdi, rbp
0x197ee:      call     0x18d10
0x197f3:      call     0x4c70
0x197f8:      mov      rdi, r14
0x197fb:      call     0x18d10
0x19800:      add      rsp, 8
0x19804:      mov      rax, r12
0x19807:      pop      rbx
0x19808:      pop      rbp
0x19809:      pop      r12
0x1980b:      pop      r13
0x1980d:      pop      r14
0x1980f:      pop      r15
0x19811:      ret

```



```

0x19812:      nop      word ptr [rax + rax]
0x19818:      lea      rdi, [rip + 0x6d41]
0x1981f:      call     0x47e0
0x19824:      mov      r14, rax
0x19827:      test     rax, rax
0x1982a:      je       0x198f8
0x19830:      mov      rdi, r14
0x19833:      call     0x49b0
0x19838:      mov      rdi, r13
0x1983b:      mov      rbx, rax
0x1983e:      call     0x49b0
0x19843:      movsxd   rbx, ebx
0x19846:      lea      rdi, [rbx + rax + 1]
0x1984b:      call     0x1d670
0x19850:      mov      rsi, r14
0x19853:      mov      r12, rax
0x19856:      mov      rdi, rax
0x19859:      call     0x48b0
0x1985e:      lea      rdi, [r12 + rbx]
0x19862:      mov      rsi, r13
0x19865:      call     0x48b0
0x1986a:      mov      rdi, rbp
0x1986d:      call     0x18d10
0x19872:      jmp      0x19800
0x19800
0x19874:      nop      dword ptr [rax]
0x19878:      xor      r12d, r12d
0x1987b:      jmp      0x19800
0x19800
0x1987d:      nop      dword ptr [rax]
0x19880:      mov      rdx, qword ptr [rip + 0x12251]
0x19887:      test     rdx, rdx
0x1988a:      je       0x197f3
0x19890:      mov      rdi, r14
0x19893:      xor      eax, eax
0x19895:      call     rdx
0x19897:      mov      r15, rax
0x1989a:      test     rax, rax
0x1989d:      je       0x197f3
0x198a3:      mov      rdi, rax
0x198a6:      call     0x49b0
0x198ab:      mov      rdi, r13
0x198ae:      mov      rbx, rax
0x198b1:      call     0x49b0
0x198b6:      lea      edi, [rbx + 1]
0x198b9:      movsxd   rdi, edi
0x198bc:      add      rdi, rax
0x198bf:      call     0x1d670

```

```

0x198c4:      mov     rsi, r15
0x198c7:      mov     rdi, rax
0x198ca:      mov     r12, rax
0x198cd:      call    0x48b0
0x198d2:      movsxd  rdi, ebx
0x198d5:      mov     rsi, r13
0x198d8:      add     rdi, r12
0x198db:      call    0x48b0
0x198e0:      mov     rdi, r15
0x198e3:      call    0x18d10
0x198e8:      mov     rdi, rbp
0x198eb:      call    0x18d10
0x198f0:      jmp     0x197f3
0x197f3:
0x198f5:      nop     dword ptr [rax]
0x198f8:      call    0x49e0
0x198fd:      mov     edi, eax
0x198ff:      call    0x4970
0x19904:      test    rax, rax
0x19907:      je      0x19916
0x19909:      mov     r14, qword ptr [rax + 0x20]
0x1990d:      test    r14, r14
0x19910:      jne     0x19830
0x19916:      mov     rdi, r13
0x19919:      call    0x49b0
0x1991e:      lea     rdi, [rax + 1]
0x19922:      call    0x1d670
0x19927:      mov     r12, rax
0x1992a:      mov     rdi, rax
0x1992d:      jmp     0x19862
0x19862:
0x19932:      nop     word ptr [rax + rax]
0x19938:      mov     rax, r14
0x1993b:      jmp     0x19794
0x19794:
0x19940:      endbr64
0x19944:      push    r15
0x19946:      push    r14
0x19948:      mov     r14, rdi
0x1994b:      push    r13
0x1994d:      push    r12
0x1994f:      push    rbp
0x19950:      push    rbx
0x19951:      xor     ebx, ebx
0x19953:      sub     rsp, 0x48
0x19957:      mov     dword ptr [rsp + 0x1c], 0
0x1995f:      mov     qword ptr [rsp + 0x10], 0
0x19968:      nop     dword ptr [rax + rax]

```

```

0x19970:      mov     r15, qword ptr [rip + 0x10e31]
0x19977:      mov     rdi, r14
0x1997a:      mov     qword ptr [rsp + 8], r15
0x1997f:      call    0x49b0
0x19984:      mov     qword ptr [rsp + 0x20], rax
0x19989:      mov     rcx, rax
0x1998c:      movzx   eax, byte ptr [r14]
0x19990:      test    al, al
0x19992:      je      0x19bd0
0x19998:      cmp     al, 0x7e
0x1999a:      je      0x19bd0
0x199a0:      mov     dword ptr [rsp + 0x18], ecx
0x199a4:      mov     rax, rcx
0x199a7:      test    r15, r15
0x199aa:      je      0x19c50
0x199b0:      mov     edx, dword ptr [rsp + 0x20]
0x199b4:      test    edx, edx
0x199b6:      jle     0x19c4a
0x199bc:      mov     rax, qword ptr [rsp + 8]
0x199c1:      mov     dword ptr [rsp + 0x3c], ebx
0x199c5:      xor     r12d, r12d
0x199c8:      mov     rax, qword ptr [rax]
0x199cb:      mov     qword ptr [rsp + 0x28], rax
0x199d0:      mov     rax, qword ptr [rsp + 0x20]
0x199d5:      sub     eax, 1
0x199d8:      mov     qword ptr [rsp + 0x30], rax
0x199dd:      mov     rax, qword ptr [rsp + 8]
0x199e2:      mov     dword ptr [rsp + 0x38], r12d
0x199e7:      lea     rbp, [r14 + r12]
0x199eb:      lea     rbx, [rax + 8]
0x199ef:      mov     rax, qword ptr [rsp + 0x28]
0x199f4:      mov     r13, rax
0x199f7:      test    rax, rax
0x199fa:      jne     0x19a18
0x199fc:      jmp     0x19bb0
0x19bb0:
0x19a01:      nop     dword ptr [rax]
0x19a08:      mov     r13, qword ptr [rbx]
0x19a0b:      add     rbx, 8
0x19a0f:      test    r13, r13
0x19a12:      je      0x19bb0
0x19a18:      mov     rdi, r13
0x19a1b:      call    0x49b0
0x19a20:      mov     rsi, r13
0x19a23:      mov     rdi, rbp
0x19a26:      mov     rdx, rax
0x19a29:      mov     r15, rax
0x19a2c:      call    0x4890

```

```

0x19a31:    test    eax, eax
0x19a33:    jne     0x19a08
0x19a35:    mov     eax, dword ptr [rsp + 0x38]
0x19a39:    mov     ebx, dword ptr [rsp + 0x3c]
0x19a3d:    lea     eax, [rax + r15 - 1]
0x19a42:    lea     ecx, [rbx + rax]
0x19a45:    mov     dword ptr [rsp + 0x18], eax
0x19a49:    movsxd  r13, eax
0x19a4c:    mov     dword ptr [rsp + 8], ecx
0x19a50:    lea     r15, [r14 + r13]
0x19a54:    mov     ecx, dword ptr [rsp + 8]
0x19a58:    cmp     dword ptr [rsp + 0x1c], ecx
0x19a5c:    jle     0x19bf0
0x19a62:    movsxd  rbx, ebx
0x19a65:    add     rbx, qword ptr [rsp + 0x10]
0x19a6a:    mov     rdx, r13
0x19a6d:    mov     rsi, r14
0x19a70:    mov     rdi, rbx
0x19a73:    mov     qword ptr [rsp + 0x28], rbx
0x19a78:    call    0x4880
0x19a7d:    mov     rdi, r15
0x19a80:    mov     r13, qword ptr [rip + 0x10d19]
0x19a87:    call    0x49b0
0x19a8c:    mov     dword ptr [rsp + 0x20], eax
0x19a90:    test    eax, eax
0x19a92:    jle     0x19c64
0x19a98:    mov     rbx, r15
0x19a9b:    xor     ebp, ebp
0x19a9d:    movzx   eax, byte ptr [rbx]
0x19aa0:    cmp     al, 0x2f
0x19aa2:    je      0x19b80
0x19aa8:    test    al, al
0x19aaa:    je      0x19b80
0x19ab0:    test    r13, r13
0x19ab3:    je      0x19aea
0x19ab5:    mov     r14, r13
0x19ab8:    jmp     0x19ae2
0x19ae2:
0x19aba:    nop     word ptr [rax + rax]
0x19ac0:    mov     rdi, r12
0x19ac3:    add     r14, 8
0x19ac7:    call    0x49b0
0x19acc:    mov     rsi, r12
0x19acf:    mov     rdi, rbx
0x19ad2:    mov     rdx, rax
0x19ad5:    call    0x4890
0x19ada:    test    eax, eax
0x19adc:    je      0x19b80

```

```

0x19ae2:    mov     r12, qword ptr [r14]
0x19ae5:    test    r12, r12
0x19ae8:    jne     0x19ac0
0x19aea:    add     ebp, 1
0x19aed:    add     rbx, 1
0x19af1:    cmp     dword ptr [rsp + 0x20], ebp
0x19af5:    jne     0x19a9d
0x19af7:    nop     word ptr [rax + rax]
0x19b00:    lea     edi, [rbp + 1]
0x19b03:    movsxd  rbp, ebp
0x19b06:    movsxd  rdi, edi
0x19b09:    lea     r14, [r15 + rbp]
0x19b0d:    call    0x1d670
0x19b12:    mov     rdx, rbp
0x19b15:    mov     rsi, r15
0x19b18:    mov     r13, rax
0x19b1b:    mov     rdi, rax
0x19b1e:    call    0x4880
0x19b23:    mov     byte ptr [r13 + rbp], 0
0x19b29:    mov     rdi, r13
0x19b2c:    xor     eax, eax
0x19b2e:    call    0x196e0
0x19b33:    mov     rdi, r13
0x19b36:    mov     rbp, rax
0x19b39:    call    0x18d10
0x19b3e:    mov     rdi, rbp
0x19b41:    call    0x49b0
0x19b46:    mov     ecx, dword ptr [rsp + 8]
0x19b4a:    lea     ebx, [rcx + rax]
0x19b4d:    cmp     ebx, dword ptr [rsp + 0x1c]
0x19b51:    jge     0x19c20
0x19b57:    movsxd  rdi, dword ptr [rsp + 8]
0x19b5c:    mov     rsi, rbp
0x19b5f:    add     rdi, qword ptr [rsp + 0x10]
0x19b64:    call    0x48b0
0x19b69:    mov     rdi, rbp
0x19b6c:    call    0x18d10
0x19b71:    jmp     0x19970
0x19970:
0x19b76:    nop     word ptr cs:[rax + rax]
0x19b80:    or      dword ptr [rsp + 0x18], ebp
0x19b84:    mov     eax, dword ptr [rsp + 0x18]
0x19b88:    test    eax, eax
0x19b8a:    jne     0x19b00
0x19b90:    mov     rax, qword ptr [rsp + 0x28]
0x19b95:    mov     byte ptr [rax], 0
0x19b98:    mov     rax, qword ptr [rsp + 0x10]
0x19b9d:    add     rsp, 0x48

```

```

0x19ba1:    pop     rbx
0x19ba2:    pop     rbp
0x19ba3:    pop     r12
0x19ba5:    pop     r13
0x19ba7:    pop     r14
0x19ba9:    pop     r15
0x19bab:    ret
0x19bac:    nop     dword ptr [rax]
0x19bb0:    lea     rax, [r12 + 1]
0x19bb5:    cmp     qword ptr [rsp + 0x30], r12
0x19bba:    je      0x19c46
0x19bc0:    mov     r12, rax
0x19bc3:    jmp     0x199dd
0x199dd
0x19bc8:    nop     dword ptr [rax + rax]
0x19bd0:    mov     dword ptr [rsp + 8], ebx
0x19bd4:    mov     r15, r14
0x19bd7:    mov     ecx, dword ptr [rsp + 8]
0x19bdb:    xor     r13d, r13d
0x19bde:    mov     dword ptr [rsp + 0x18], 0
0x19be6:    cmp     dword ptr [rsp + 0x1c], ecx
0x19bea:    jg      0x19a62
0x19bf0:    mov     eax, dword ptr [rsp + 0x1c]
0x19bf4:    mov     ecx, dword ptr [rsp + 0x18]
0x19bf8:    mov     rdi, qword ptr [rsp + 0x10]
0x19bfd:    lea     eax, [rax + rcx + 0x14]
0x19c01:    lea     esi, [rax + 1]
0x19c04:    mov     dword ptr [rsp + 0x1c], eax
0x19c08:    movsxd  rsi, esi
0x19c0b:    call    0x1d690
0x19c10:    mov     qword ptr [rsp + 0x10], rax
0x19c15:    jmp     0x19a62
0x19a62
0x19c1a:    nop     word ptr [rax + rax]
0x19c20:    mov     ecx, dword ptr [rsp + 0x1c]
0x19c24:    mov     rdi, qword ptr [rsp + 0x10]
0x19c29:    lea     eax, [rcx + rax + 0x14]
0x19c2d:    lea     esi, [rax + 1]
0x19c30:    mov     dword ptr [rsp + 0x1c], eax
0x19c34:    movsxd  rsi, esi
0x19c37:    call    0x1d690
0x19c3c:    mov     qword ptr [rsp + 0x10], rax
0x19c41:    jmp     0x19b57
0x19b57
0x19c46:    mov     ebx, dword ptr [rsp + 0x3c]
0x19c4a:    mov     rax, qword ptr [rsp + 0x20]
0x19c4f:    nop
0x19c50:    mov     ecx, eax

```

```

0x19c52:      movsxd  r13, eax
0x19c55:      add     ecx, ebx
0x19c57:      lea     r15, [r14 + r13]
0x19c5b:      mov     dword ptr [rsp + 8], ecx
0x19c5f:      jmp     0x19a54
0x19a54:
0x19c64:      xor     ebp, ebp
0x19c66:      jmp     0x19b84
0x19b84:
0x19c6b:      nop     dword ptr [rax + rax]
0x19c70:      push    r15
0x19c72:      mov     r15, rsi
0x19c75:      push    r14
0x19c77:      mov     r14d, edx
0x19c7a:      push    r13
0x19c7c:      push    r12
0x19c7e:      mov     r12, rdi
0x19c81:      push    rbp
0x19c82:      push    rbx
0x19c83:      sub     rsp, 0x18
0x19c87:      movsxd  rsi, dword ptr [rsi]
0x19c8a:      lea     r13, [rdi + rsi]
0x19c8e:      mov     edi, ecx
0x19c90:      mov     ebx, esi
0x19c92:      mov     rbp, rsi
0x19c95:      movzx   eax, byte ptr [r13]
0x19c9a:      and     edi, 1
0x19c9d:      mov     rdx, rsi
0x19ca0:      test    al, al
0x19ca2:      jne     0x19cdc
0x19ca4:      jmp     0x19db9
0x19db9:
0x19ca9:      nop     dword ptr [rax]
0x19cb0:      cmp     al, 0x5c
0x19cb2:      jne     0x19cc5
0x19cb4:      movsx   eax, byte ptr [r12 + rdx + 1]
0x19cba:      cmp     eax, r14d
0x19cbd:      sete    al
0x19cc0:      movzx   eax, al
0x19cc3:      add     ebx, eax
0x19cc5:      add     ebx, 1
0x19cc8:      movsxd  rdx, ebx
0x19ccb:      lea     r13, [r12 + rdx]
0x19ccf:      movzx   eax, byte ptr [r13]
0x19cd4:      test    al, al
0x19cd6:      je      0x19da0
0x19cdc:      movsx   ecx, al
0x19cdf:      cmp     ecx, r14d

```

```

0x19ce2:    jne     0x19cb0
0x19ce4:    cmp     ebp, ebx
0x19ce6:    jl      0x19d10
0x19ce8:    test    dil, dil
0x19ceb:    jne     0x19d10
0x19ced:    movzx   eax, byte ptr [r13]
0x19cf2:    xor     r9d, r9d
0x19cf5:    cmp     al, 1
0x19cf7:    sbb     ebx, -1
0x19cfa:    mov     dword ptr [r15], ebx
0x19cfd:    add     rsp, 0x18
0x19d01:    mov     rax, r9
0x19d04:    pop     rbx
0x19d05:    pop     rbp
0x19d06:    pop     r12
0x19d08:    pop     r13
0x19d0a:    pop     r14
0x19d0c:    pop     r15
0x19d0e:    ret
0x19d0f:    nop
0x19d10:    mov     edi, ebx
0x19d12:    mov     qword ptr [rsp], r8
0x19d16:    sub     edi, ebp
0x19d18:    mov     qword ptr [rsp + 8], rsi
0x19d1d:    add     edi, 1
0x19d20:    movsxd  rdi, edi
0x19d23:    call    0x1d670
0x19d28:    cmp     ebp, ebx
0x19d2a:    mov     r8, qword ptr [rsp]
0x19d2e:    mov     r9, rax
0x19d31:    jge     0x19dd6
0x19d37:    mov     rsi, qword ptr [rsp + 8]
0x19d3c:    mov     eax, 1
0x19d41:    jmp     0x19d5d
0x19d5d:
0x19d43:    nop     dword ptr [rax + rax]
0x19d48:    add     ebp, 1
0x19d4b:    mov     byte ptr [r9 + rax - 1], dl
0x19d50:    mov     edx, eax
0x19d52:    add     rax, 1
0x19d56:    cmp     ebp, ebx
0x19d58:    jge     0x19d80
0x19d5a:    movsxd  rsi, ebp
0x19d5d:    movzx   edx, byte ptr [r12 + rsi]
0x19d62:    cmp     dl, 0x5c
0x19d65:    jne     0x19d48
0x19d67:    movsx   esi, byte ptr [r12 + rsi + 1]
0x19d6d:    cmp     esi, r14d

```



```

0x19d70:    jne     0x19d48
0x19d72:    add     ebp, 1
0x19d75:    mov     edx, esi
0x19d77:    jmp     0x19d48
0x19d48
0x19d79:    nop     dword ptr [rax]
0x19d80:    movsxd  rax, edx
0x19d83:    add     rax, r9
0x19d86:    mov     byte ptr [rax], 0
0x19d89:    mov     dword ptr [r8], edx
0x19d8c:    movzx   eax, byte ptr [r13]
0x19d91:    jmp     0x19cf5
0x19cf5
0x19d96:    nop     word ptr cs:[rax + rax]
0x19da0:    cmp     ebp, ebx
0x19da2:    jl      0x19d10
0x19da8:    test    dil, dil
0x19dab:    jne     0x19d10
0x19db1:    xor     r9d, r9d
0x19db4:    jmp     0x19cfa
0x19cfa
0x19db9:    test    ecx, ecx
0x19dbb:    je      0x19db1
0x19dbd:    mov     edi, 1
0x19dc2:    mov     qword ptr [rsp], r8
0x19dc6:    call    0x1d670
0x19dcb:    mov     r8, qword ptr [rsp]
0x19dcf:    xor     edx, edx
0x19dd1:    mov     r9, rax
0x19dd4:    jmp     0x19d86
0x19d86
0x19dd6:    xor     edx, edx
0x19dd8:    jmp     0x19d86
0x19d86
0x19dda:    nop     word ptr [rax + rax]
0x19de0:    push    r15
0x19de2:    push    r14
0x19de4:    push    r13
0x19de6:    push    r12
0x19de8:    push    rbp
0x19de9:    push    rbx
0x19dea:    sub     rsp, 0x28
0x19dee:    mov     eax, dword ptr [rip + 0x11cf0]
0x19df4:    mov     ebx, dword ptr [rip + 0x11d02]
0x19dfa:    mov     dword ptr [rsp + 8], esi
0x19dfe:    mov     dword ptr [rsp + 0x18], edx
0x19e02:    mov     dword ptr [rsp + 0x1c], eax
0x19e06:    test    eax, eax

```

```

0x19e08:    je      0x19f9d
0x19e0e:    mov     rbp, rdi
0x19e11:    cmp     eax, ebx
0x19e13:    je      0x19fb0
0x19e19:    call    0x49b0
0x19e1e:    mov     ecx, dword ptr [rsp + 8]
0x19e22:    mov     dword ptr [rsp + 0xc], eax
0x19e26:    test    ecx, ecx
0x19e28:    js      0x19fbb
0x19e2e:    mov     rax, qword ptr [rip + 0x11ccb]
0x19e35:    movsxd  r14, dword ptr [rsp + 0xc]
0x19e3a:    mov     qword ptr [rsp + 0x10], rax
0x19e3f:    mov     edx, dword ptr [rsp + 8]
0x19e43:    test    edx, edx
0x19e45:    js      0x19f16
0x19e4b:    cmp     dword ptr [rsp + 0x1c], ebx
0x19e4f:    je      0x19f9d
0x19e55:    mov     rcx, qword ptr [rsp + 0x10]
0x19e5a:    movsxd  rax, ebx
0x19e5d:    mov     rax, qword ptr [rcx + rax*8]
0x19e61:    mov     r13, qword ptr [rax]
0x19e64:    mov     rdi, r13
0x19e67:    call    0x49b0
0x19e6c:    mov     r10, rax
0x19e6f:    cmp     dword ptr [rsp + 0xc], eax
0x19e73:    jg      0x19f07
0x19e79:    cmp     dword ptr [rsp + 0x18], 1
0x19e7e:    movzx   r12d, byte ptr [rbp]
0x19e83:    je      0x19ee0
0x19e85:    mov     r15d, dword ptr [rsp + 0x18]
0x19e8a:    sub     r10d, dword ptr [rsp + 0xc]
0x19e8f:    jmp     0x19ea5
0x19ea5:
0x19e91:    nop     dword ptr [rax]
0x19e98:    add     r15d, 1
0x19e9c:    add     r13, 1
0x19ea0:    cmp     r15d, r10d
0x19ea3:    jg      0x19f07
0x19ea5:    cmp     byte ptr [r13], r12b
0x19ea9:    jne     0x19e98
0x19eab:    mov     rdx, r14
0x19eae:    mov     rsi, r13
0x19eb1:    mov     rdi, rbp
0x19eb4:    mov     dword ptr [rsp + 4], r10d
0x19eb9:    call    0x4890
0x19ebe:    mov     r10d, dword ptr [rsp + 4]
0x19ec3:    test    eax, eax
0x19ec5:    jne     0x19e98

```

```

0x19ec7:    mov     dword ptr [rip + 0x11c2f], ebx
0x19ecd:    add     rsp, 0x28
0x19ed1:    mov     eax, r15d
0x19ed4:    pop     rbx
0x19ed5:    pop     rbp
0x19ed6:    pop     r12
0x19ed8:    pop     r13
0x19eda:    pop     r14
0x19edc:    pop     r15
0x19ede:    ret
0x19edf:    nop
0x19ee0:    cmp     byte ptr [r13], r12b
0x19ee4:    jne     0x19efb
0x19ee6:    mov     rdx, r14
0x19ee9:    mov     rsi, r13
0x19eec:    mov     rdi, rbp
0x19eef:    call    0x4890
0x19ef4:    mov     r15d, eax
0x19ef7:    test    eax, eax
0x19ef9:    je      0x19ec7
0x19efb:    mov     eax, dword ptr [rsp + 8]
0x19eff:    test    eax, eax
0x19f01:    js      0x19f90
0x19f07:    mov     edx, dword ptr [rsp + 8]
0x19f0b:    add     ebx, 1
0x19f0e:    test    edx, edx
0x19f10:    jns     0x19e4b
0x19f16:    mov     rcx, qword ptr [rsp + 0x10]
0x19f1b:    movsxd  rax, ebx
0x19f1e:    mov     rax, qword ptr [rcx + rax*8]
0x19f22:    mov     r13, qword ptr [rax]
0x19f25:    mov     rdi, r13
0x19f28:    call    0x49b0
0x19f2d:    mov     r8, rax
0x19f30:    cmp     dword ptr [rsp + 0xc], eax
0x19f34:    jg      0x19f90
0x19f36:    cmp     dword ptr [rsp + 0x18], 1
0x19f3b:    movzx   r12d, byte ptr [rbp]
0x19f40:    je      0x19ee0
0x19f42:    mov     r15d, r8d
0x19f45:    sub     r15d, dword ptr [rsp + 0xc]
0x19f4a:    movsxd  rcx, r15d
0x19f4d:    add     r13, rcx
0x19f50:    jmp     0x19f66
0x19f66:
0x19f52:    nop     word ptr [rax + rax]
0x19f58:    sub     r15d, 1
0x19f5c:    sub     r13, 1

```

```

0x19f60:    cmp     r15d, -1
0x19f64:    je      0x19f90
0x19f66:    cmp     byte ptr [r13], r12b
0x19f6a:    jne     0x19f58
0x19f6c:    mov     rdx, r14
0x19f6f:    mov     rsi, r13
0x19f72:    mov     rdi, rbp
0x19f75:    call    0x4890
0x19f7a:    test    eax, eax
0x19f7c:    je      0x19ec7
0x19f82:    sub     r15d, 1
0x19f86:    sub     r13, 1
0x19f8a:    cmp     r15d, -1
0x19f8e:    jne     0x19f66
0x19f90:    sub     ebx, 1
0x19f93:    test    dword ptr [rsp + 8], ebx
0x19f97:    jns     0x19e3f
0x19f9d:    mov     r15d, 0xffffffff
0x19fa3:    jmp     0x19ecd
0x19ecd:
0x19fa8:    nop     dword ptr [rax + rax]
0x19fb0:    test    esi, esi
0x19fb2:    jns     0x19f9d
0x19fb4:    mov     eax, dword ptr [rsp + 0x1c]
0x19fb8:    lea     ebx, [rax - 1]
0x19fbb:    mov     rdi, rbp
0x19fbe:    call    0x49b0
0x19fc3:    mov     dword ptr [rsp + 0xc], eax
0x19fc7:    test    dword ptr [rsp + 8], ebx
0x19fcb:    jns     0x19e2e
0x19fd1:    jmp     0x19f9d
0x19f9d:
0x19fd3:    nop     word ptr cs:[rax + rax]
0x19fde:    nop
0x19fe0:    push    r14
0x19fe2:    push    r13
0x19fe4:    movsxd  r13, esi
0x19fe7:    sub     edx, r13d
0x19fea:    push    r12
0x19fec:    push    rbp
0x19fed:    mov     rbp, rdi
0x19ff0:    push    rbx
0x19ff1:    movsxd  rbx, edx
0x19ff4:    cmp     ecx, 2
0x19ff7:    je      0x1a070
0x19ff9:    mov     eax, 0x1d
0x19ffe:    lea     r14, [rip + 0x843a]
0x1a005:    cmp     ecx, 3

```

```

0x1a008:      je      0x1a028
0x1a00a:      xor     eax, eax
0x1a00c:      lea     r14, [rip + 0x83f5]
0x1a013:      lea     rdx, [rip + 0x83fe]
0x1a01a:      cmp     ecx, 1
0x1a01d:      sete    al
0x1a020:      cmov     r14, rdx
0x1a024:      lea     eax, [rax + rax*2 + 0xf]
0x1a028:      lea     edi, [rax + rbx + 3]
0x1a02c:      movsxd   rdi, edi
0x1a02f:      call    0x1d670
0x1a034:      lea     rsi, [rbp + r13]
0x1a039:      mov     rdx, rbx
0x1a03c:      mov     r12, rax
0x1a03f:      mov     rdi, rax
0x1a042:      call    0x4880
0x1a047:      mov     byte ptr [r12 + rbx], 0x3a
0x1a04c:      lea     rdi, [r12 + rbx + 2]
0x1a051:      mov     rsi, r14
0x1a054:      mov     byte ptr [r12 + rbx + 1], 0x20
0x1a05a:      call    0x48b0
0x1a05f:      pop     rbx
0x1a060:      mov     rax, r12
0x1a063:      pop     rbp
0x1a064:      pop     r12
0x1a066:      pop     r13
0x1a068:      pop     r14
0x1a06a:      ret
0x1a06b:      nop     dword ptr [rax + rax]
0x1a070:      mov     eax, 0x13
0x1a075:      lea     r14, [rip + 0x83af]
0x1a07c:      jmp     0x1a028
0x1a028
0x1a07e:      nop
0x1a080:      push    r12
0x1a082:      push    rbp
0x1a083:      push    rbx
0x1a084:      test    rdi, rdi
0x1a087:      je      0x1a0b0
0x1a089:      mov     rbp, rdi
0x1a08c:      call    0x49b0
0x1a091:      lea     rdi, [rax + 1]
0x1a095:      call    0x1d670
0x1a09a:      pop     rbx
0x1a09b:      mov     rsi, rbp
0x1a09e:      pop     rbp
0x1a09f:      mov     rdi, rax
0x1a0a2:      pop     r12

```

```

0x1a0a4:      jmp      0x48b0
0x48b0
0x1a0a9:      nop      dword ptr [rax]
0x1a0b0:      lea      rdi, [rip + 0x64a9]
0x1a0b7:      call     0x47e0
0x1a0bc:      mov      r12, rax
0x1a0bf:      test     rax, rax
0x1a0c2:      je       0x1a110
0x1a0c4:      mov      rdi, rax
0x1a0c7:      call     0x49b0
0x1a0cc:      lea      edi, [rax + 0xa]
0x1a0cf:      movsxd   rbp, eax
0x1a0d2:      movsxd   rdi, edi
0x1a0d5:      lea      rbx, [rbp + 1]
0x1a0d9:      call     0x1d670
0x1a0de:      mov      rsi, r12
0x1a0e1:      mov      rdi, rax
0x1a0e4:      call     0x48b0
0x1a0e9:      movabs   rdx, 0x79726f747369682e
0x1a0f3:      mov      byte ptr [rax + rbp], 0x2f
0x1a0f7:      mov      r8, rax
0x1a0fa:      lea      rax, [rax + rbx]
0x1a0fe:      mov      qword ptr [rax], rdx
0x1a101:      mov      byte ptr [rax + 8], 0
0x1a105:      mov      rax, r8
0x1a108:      pop      rbx
0x1a109:      pop      rbp
0x1a10a:      pop      r12
0x1a10c:      ret
0x1a10d:      nop      dword ptr [rax]
0x1a110:      mov      ebx, 2
0x1a115:      mov      ebp, 1
0x1a11a:      mov      edi, 0xb
0x1a11f:      lea      r12, [rip + 0x75b8]
0x1a126:      jmp      0x1a0d9
0x1a0d9
0x1a128:      nop      dword ptr [rax + rax]
0x1a130:      push     r15
0x1a132:      xor      eax, eax
0x1a134:      push     r14
0x1a136:      mov      r14d, esi
0x1a139:      push     r13
0x1a13b:      push     r12
0x1a13d:      push     rbp
0x1a13e:      push     rbx
0x1a13f:      mov      ebx, edx
0x1a141:      sub      rsp, 0x18
0x1a145:      call     0x1a080

```

```

0x1a14a:    cmp     ebx, 1
0x1a14d:    mov     edx, 0x1b6
0x1a152:    sbb     esi, esi
0x1a154:    mov     r13, rax
0x1a157:    mov     rdi, rax
0x1a15a:    xor     eax, eax
0x1a15c:    and     esi, 0x1c0
0x1a162:    add     esi, 0x241
0x1a168:    call    0x4d80
0x1a16d:    cmp     eax, -1
0x1a170:    je      0x1a270
0x1a176:    mov     ebp, eax
0x1a178:    mov     eax, dword ptr [rip + 0x11966]
0x1a17e:    cmp     eax, r14d
0x1a181:    mov     edx, eax
0x1a183:    cmovle  r14d, eax
0x1a187:    sub     edx, r14d
0x1a18a:    cmp     eax, edx
0x1a18c:    jle     0x1a290
0x1a192:    sub     eax, 1
0x1a195:    mov     rcx, qword ptr [rip + 0x11964]
0x1a19c:    movsxd  rsi, edx
0x1a19f:    xor     r12d, r12d
0x1a1a2:    sub     eax, edx
0x1a1a4:    add     rax, rsi
0x1a1a7:    lea     rbx, [rcx + rsi*8]
0x1a1ab:    lea     r15, [rcx + rax*8 + 8]
0x1a1b0:    mov     rax, qword ptr [rbx]
0x1a1b3:    add     rbx, 8
0x1a1b7:    mov     rdi, qword ptr [rax]
0x1a1ba:    call    0x49b0
0x1a1bf:    lea     r12d, [r12 + rax + 1]
0x1a1c4:    cmp     r15, rbx
0x1a1c7:    jne     0x1a1b0
0x1a1c9:    movsxd  rax, r12d
0x1a1cc:    mov     qword ptr [rsp + 8], rax
0x1a1d1:    mov     rdi, qword ptr [rsp + 8]
0x1a1d6:    call    0xd670
0x1a1db:    mov     r15, rax
0x1a1de:    mov     eax, dword ptr [rip + 0x11900]
0x1a1e4:    mov     ebx, eax
0x1a1e6:    sub     ebx, r14d
0x1a1e9:    cmp     eax, ebx
0x1a1eb:    jle     0x1a238
0x1a1ed:    mov     r12, qword ptr [rip + 0x1190c]
0x1a1f4:    movsxd  rbx, ebx
0x1a1f7:    xor     r14d, r14d
0x1a1fa:    nop     word ptr [rax + rax]

```

```

0x1a200:    mov     rax, qword ptr [r12 + rbx*8]
0x1a204:    movsxd  rdi, r14d
0x1a207:    add     rdi, r15
0x1a20a:    mov     rsi, qword ptr [rax]
0x1a20d:    call    0x48b0
0x1a212:    mov     rax, qword ptr [r12 + rbx*8]
0x1a216:    add     rbx, 1
0x1a21a:    mov     rdi, qword ptr [rax]
0x1a21d:    call    0x49b0
0x1a222:    add     eax, r14d
0x1a225:    lea     r14d, [rax + 1]
0x1a229:    cdqe
0x1a22b:    mov     byte ptr [r15 + rax], 0xa
0x1a230:    cmp     dword ptr [rip + 0x118ae], ebx
0x1a236:    jg      0x1a200
0x1a238:    mov     rdx, qword ptr [rsp + 8]
0x1a23d:    mov     rsi, r15
0x1a240:    mov     edi, ebp
0x1a242:    call    0x4790
0x1a247:    mov     rdi, r15
0x1a24a:    call    0x18d10
0x1a24f:    mov     edi, ebp
0x1a251:    call    0x4ab0
0x1a256:    mov     rdi, r13
0x1a259:    call    0x18d10
0x1a25e:    add     rsp, 0x18
0x1a262:    xor     eax, eax
0x1a264:    pop     rbx
0x1a265:    pop     rbp
0x1a266:    pop     r12
0x1a268:    pop     r13
0x1a26a:    pop     r14
0x1a26c:    pop     r15
0x1a26e:    ret
0x1a26f:    nop
0x1a270:    mov     rdi, r13
0x1a273:    call    0x18d10
0x1a278:    call    0x4860
0x1a27d:    mov     eax, dword ptr [rax]
0x1a27f:    add     rsp, 0x18
0x1a283:    pop     rbx
0x1a284:    pop     rbp
0x1a285:    pop     r12
0x1a287:    pop     r13
0x1a289:    pop     r14
0x1a28b:    pop     r15
0x1a28d:    ret
0x1a28e:    nop

```



```

0x1a290:      mov     qword ptr [rsp + 8], 0
0x1a299:      jmp     0x1a1d1
0x1a1d1:
0x1a29e:      nop
0x1a2a0:      endbr64
0x1a2a4:      sub     rsp, 8
0x1a2a8:      mov     edi, 0x18
0x1a2ad:      call    0x1d670
0x1a2b2:      mov     rdx, qword ptr [rip + 0x11847]
0x1a2b9:      mov     qword ptr [rax], rdx
0x1a2bc:      mov     edx, dword ptr [rip + 0x1183a]
0x1a2c2:      mov     dword ptr [rax + 8], edx
0x1a2c5:      mov     edx, dword ptr [rip + 0x11819]
0x1a2cb:      mov     dword ptr [rax + 0xc], edx
0x1a2ce:      mov     edx, dword ptr [rip + 0x11824]
0x1a2d4:      mov     dword ptr [rax + 0x10], edx
0x1a2d7:      add     rsp, 8
0x1a2db:      ret
0x1a2dc:      nop     dword ptr [rax]
0x1a2e0:      endbr64
0x1a2e4:      mov     rax, qword ptr [rdi]
0x1a2e7:      mov     qword ptr [rip + 0x11812], rax
0x1a2ee:      mov     eax, dword ptr [rdi + 8]
0x1a2f1:      mov     dword ptr [rip + 0x11805], eax
0x1a2f7:      mov     eax, dword ptr [rdi + 0xc]
0x1a2fa:      mov     dword ptr [rip + 0x117e4], eax
0x1a300:      mov     eax, dword ptr [rdi + 0x10]
0x1a303:      mov     dword ptr [rip + 0x117ef], eax
0x1a309:      ret
0x1a30a:      nop     word ptr [rax + rax]
0x1a310:      endbr64
0x1a314:      mov     eax, dword ptr [rip + 0x117ca]
0x1a31a:      mov     dword ptr [rip + 0x117dc], eax
0x1a320:      ret
0x1a321:      nop     word ptr cs:[rax + rax]
0x1a32c:      nop     dword ptr [rax]
0x1a330:      endbr64
0x1a334:      push    r12
0x1a336:      xor     r12d, r12d
0x1a339:      push    rbx
0x1a33a:      sub     rsp, 8
0x1a33e:      mov     rbx, qword ptr [rip + 0x117bb]
0x1a345:      test    rbx, rbx
0x1a348:      jne     0x1a35b
0x1a34a:      jmp     0x1a367
0x1a367:
0x1a34c:      nop     dword ptr [rax]
0x1a350:      mov     rdi, qword ptr [rax]

```

```

0x1a353:    call    0x49b0
0x1a358:    add     r12d, eax
0x1a35b:    mov     rax, qword ptr [rbx]
0x1a35e:    add     rbx, 8
0x1a362:    test    rax, rax
0x1a365:    jne     0x1a350
0x1a367:    add     rsp, 8
0x1a36b:    mov     eax, r12d
0x1a36e:    pop     rbx
0x1a36f:    pop     r12
0x1a371:    ret
0x1a372:    nop     word ptr cs:[rax + rax]
0x1a37d:    nop     dword ptr [rax]
0x1a380:    endbr64
0x1a384:    push    rbp
0x1a385:    mov     rbp, rdi
0x1a388:    push    rbx
0x1a389:    sub     rsp, 8
0x1a38d:    mov     eax, dword ptr [rip + 0x11759]
0x1a393:    test    eax, eax
0x1a395:    je      0x1a3aa
0x1a397:    movsxd  rdx, dword ptr [rip + 0x11746]
0x1a39e:    cmp     edx, dword ptr [rip + 0x11744]
0x1a3a4:    je      0x1a458
0x1a3aa:    mov     eax, dword ptr [rip + 0x11748]
0x1a3b0:    test    eax, eax
0x1a3b2:    je      0x1a428
0x1a3b4:    mov     edx, dword ptr [rip + 0x1172a]
0x1a3ba:    lea     ecx, [rax - 1]
0x1a3bd:    cmp     ecx, edx
0x1a3bf:    je      0x1a4b0
0x1a3c5:    add     edx, 1
0x1a3c8:    mov     dword ptr [rip + 0x11716], edx
0x1a3ce:    mov     edi, 0x10
0x1a3d3:    call    0x1d670
0x1a3d8:    mov     rdi, rbp
0x1a3db:    mov     rbx, rax
0x1a3de:    call    0x49b0
0x1a3e3:    lea     rdi, [rax + 1]
0x1a3e7:    call    0x1d670
0x1a3ec:    mov     rsi, rbp
0x1a3ef:    mov     rdi, rax
0x1a3f2:    call    0x48b0
0x1a3f7:    movsxd  rdx, dword ptr [rip + 0x116e6]
0x1a3fe:    mov     qword ptr [rbx + 8], 0
0x1a406:    mov     qword ptr [rbx], rax
0x1a409:    mov     rax, qword ptr [rip + 0x116f0]
0x1a410:    mov     qword ptr [rax + rdx*8], 0

```

```

0x1a418:    mov     qword ptr [rax + rdx*8 - 8], rbx
0x1a41d:    add     rsp, 8
0x1a421:    pop     rbx
0x1a422:    pop     rbp
0x1a423:    ret
0x1a424:    nop     dword ptr [rax]
0x1a428:    mov     dword ptr [rip + 0x116c6], 0x32
0x1a432:    mov     edi, 0x190
0x1a437:    call    0x1d670
0x1a43c:    mov     dword ptr [rip + 0x1169e], 1
0x1a446:    mov     qword ptr [rip + 0x116b3], rax
0x1a44d:    jmp     0x1a3ce
0x1a3ce:
0x1a452:    nop     word ptr [rax + rax]
0x1a458:    test    edx, edx
0x1a45a:    je      0x1a41d
0x1a45c:    mov     rax, qword ptr [rip + 0x1169d]
0x1a463:    mov     rax, qword ptr [rax]
0x1a466:    test    rax, rax
0x1a469:    je      0x1a489
0x1a46b:    mov     rdi, qword ptr [rax]
0x1a46e:    call    0x18d10
0x1a473:    mov     rax, qword ptr [rip + 0x11686]
0x1a47a:    mov     rdi, qword ptr [rax]
0x1a47d:    call    0x18d10
0x1a482:    movsxd  rdx, dword ptr [rip + 0x1165b]
0x1a489:    test    edx, edx
0x1a48b:    jle     0x1a4a1
0x1a48d:    mov     rdi, qword ptr [rip + 0x1166c]
0x1a494:    shl     rdx, 3
0x1a498:    lea     rsi, [rdi + 8]
0x1a49c:    call    0x4d40
0x1a4a1:    add     dword ptr [rip + 0x10340], 1
0x1a4a8:    jmp     0x1a3ce
0x1a3ce:
0x1a4ad:    nop     dword ptr [rax]
0x1a4b0:    add     eax, 0x32
0x1a4b3:    mov     rdi, qword ptr [rip + 0x11646]
0x1a4ba:    mov     dword ptr [rip + 0x11638], eax
0x1a4c0:    cdqe
0x1a4c2:    lea     rsi, [rax*8]
0x1a4ca:    call    0x1d690
0x1a4cf:    mov     edx, dword ptr [rip + 0x1160f]
0x1a4d5:    mov     qword ptr [rip + 0x11624], rax
0x1a4dc:    jmp     0x1a3c5
0x1a3c5:
0x1a4e1:    nop     word ptr cs:[rax + rax]
0x1a4ec:    nop     dword ptr [rax]

```

```

0x1a4f0:      endbr64
0x1a4f4:      push     r14
0x1a4f6:      xor      r14d, r14d
0x1a4f9:      push     r13
0x1a4fb:      mov      r13, rdx
0x1a4fe:      push     r12
0x1a500:      mov      r12, rsi
0x1a503:      push     rbp
0x1a504:      push     rbx
0x1a505:      movsxd   rbx, edi
0x1a508:      mov      edi, 0x10
0x1a50d:      call     0x1d670
0x1a512:      cmp      dword ptr [rip + 0x115cc], ebx
0x1a518:      jle      0x1a557
0x1a51a:      mov      rbp, rax
0x1a51d:      mov      rax, qword ptr [rip + 0x115dc]
0x1a524:      mov      rdi, r12
0x1a527:      mov      r14, qword ptr [rax + rbx*8]
0x1a52b:      call     0x49b0
0x1a530:      lea      rdi, [rax + 1]
0x1a534:      call     0x1d670
0x1a539:      mov      rsi, r12
0x1a53c:      mov      rdi, rax
0x1a53f:      call     0x48b0
0x1a544:      mov      qword ptr [rbp + 8], r13
0x1a548:      mov      qword ptr [rbp], rax
0x1a54c:      mov      rax, qword ptr [rip + 0x115ad]
0x1a553:      mov      qword ptr [rax + rbx*8], rbp
0x1a557:      pop      rbx
0x1a558:      mov      rax, r14
0x1a55b:      pop      rbp
0x1a55c:      pop      r12
0x1a55e:      pop      r13
0x1a560:      pop      r14
0x1a562:      ret
0x1a563:      nop      word ptr cs:[rax + rax]
0x1a56e:      nop
0x1a570:      endbr64
0x1a574:      mov      eax, dword ptr [rip + 0x11582]
0x1a57a:      ret
0x1a57b:      nop      dword ptr [rax + rax]
0x1a580:      endbr64
0x1a584:      xor      edx, edx
0x1a586:      xor      eax, eax
0x1a588:      jmp      0x19de0
0x19de0:
0x1a58d:      nop      dword ptr [rax]
0x1a590:      endbr64

```

```

0x1a594:      mov     edx, 1
0x1a599:      xor     eax, eax
0x1a59b:      jmp     0x19de0
0x19de0:
0x1a5a0:      endbr64
0x1a5a4:      push    r12
0x1a5a6:      push    rbx
0x1a5a7:      sub     rsp, 8
0x1a5ab:      mov     eax, dword ptr [rip + 0x11533]
0x1a5b1:      cmp     eax, edi
0x1a5b3:      jle     0x1a610
0x1a5b5:      test    eax, eax
0x1a5b7:      je      0x1a610
0x1a5b9:      lea     ebx, [rax - 1]
0x1a5bc:      movsxd   rdx, edi
0x1a5bf:      mov     rcx, qword ptr [rip + 0x1153a]
0x1a5c6:      mov     esi, ebx
0x1a5c8:      shl     rdx, 3
0x1a5cc:      sub     esi, edi
0x1a5ce:      cmp     eax, edi
0x1a5d0:      lea     r9, [rcx + rdx]
0x1a5d4:      mov     eax, 8
0x1a5d9:      lea     r8, [rsi*8 + 8]
0x1a5e1:      mov     r12, qword ptr [r9]
0x1a5e4:      mov     rdi, r9
0x1a5e7:      lea     rsi, [rcx + rdx + 8]
0x1a5ec:      cmovle  r8, rax
0x1a5f0:      mov     rdx, r8
0x1a5f3:      call    0x4d40
0x1a5f8:      mov     dword ptr [rip + 0x114e6], ebx
0x1a5fe:      add     rsp, 8
0x1a602:      mov     rax, r12
0x1a605:      pop     rbx
0x1a606:      pop     r12
0x1a608:      ret
0x1a609:      nop     dword ptr [rax]
0x1a610:      xor     r12d, r12d
0x1a613:      add     rsp, 8
0x1a617:      mov     rax, r12
0x1a61a:      pop     rbx
0x1a61b:      pop     r12
0x1a61d:      ret
0x1a61e:      nop
0x1a620:      endbr64
0x1a624:      push    r12
0x1a626:      mov     eax, dword ptr [rip + 0x114b8]
0x1a62c:      push    rbp
0x1a62d:      xor     ebp, ebp

```

```

0x1a62f:      test     edi, edi
0x1a631:      cmovns  ebp, edi
0x1a634:      push    rbx
0x1a635:      cmp     eax, ebp
0x1a637:      jle     0x1a6ce
0x1a63d:      sub     eax, ebp
0x1a63f:      mov     r12d, edi
0x1a642:      test    eax, eax
0x1a644:      jle     0x1a6e8
0x1a64a:      xor     ebx, ebx
0x1a64c:      nop     dword ptr [rax]
0x1a650:      mov     rax, qword ptr [rip + 0x114a9]
0x1a657:      mov     rax, qword ptr [rax + rbx*8]
0x1a65b:      mov     rdi, qword ptr [rax]
0x1a65e:      call    0x18d10
0x1a663:      mov     rax, qword ptr [rip + 0x11496]
0x1a66a:      mov     rdi, qword ptr [rax + rbx*8]
0x1a66e:      call    0x18d10
0x1a673:      mov     eax, dword ptr [rip + 0x1146b]
0x1a679:      lea     edx, [rbx + 1]
0x1a67c:      add     rbx, 1
0x1a680:      sub     eax, ebp
0x1a682:      cmp     eax, ebx
0x1a684:      jg      0x1a650
0x1a686:      mov     dword ptr [rip + 0x1015c], edx
0x1a68c:      mov     rcx, qword ptr [rip + 0x1146d]
0x1a693:      test    r12d, r12d
0x1a696:      jle     0x1a6f0
0x1a698:      cdqe
0x1a69a:      lea     rsi, [rcx + rax*8]
0x1a69e:      xor     eax, eax
0x1a6a0:      mov     rdx, qword ptr [rsi + rax*8]
0x1a6a4:      mov     qword ptr [rcx + rax*8], rdx
0x1a6a8:      add     rax, 1
0x1a6ac:      cmp     ebp, eax
0x1a6ae:      jg      0x1a6a0
0x1a6b0:      test    ebp, ebp
0x1a6b2:      mov     eax, 1
0x1a6b7:      cmovg   eax, ebp
0x1a6ba:      movsxd  rdx, eax
0x1a6bd:      lea     rcx, [rcx + rdx*8]
0x1a6c1:      mov     qword ptr [rcx], 0
0x1a6c8:      mov     dword ptr [rip + 0x11416], eax
0x1a6ce:      pop     rbx
0x1a6cf:      mov     dword ptr [rip + 0x11413], ebp
0x1a6d5:      mov     dword ptr [rip + 0x1140d], 1
0x1a6df:      pop     rbp
0x1a6e0:      pop     r12

```

```

0x1a6e2:      ret
0x1a6e3:      nop      dword ptr [rax + rax]
0x1a6e8:      xor      edx, edx
0x1a6ea:      jmp      0x1a686
0x1a686
0x1a6ec:      nop      dword ptr [rax]
0x1a6f0:      xor      eax, eax
0x1a6f2:      jmp      0x1a6c1
0x1a6c1
0x1a6f4:      nop      word ptr cs:[rax + rax]
0x1a6ff:      nop
0x1a700:      endbr64
0x1a704:      mov      edx, dword ptr [rip + 0x113e2]
0x1a70a:      mov      eax, dword ptr [rip + 0x113d8]
0x1a710:      test     edx, edx
0x1a712:      je       0x1a720
0x1a714:      mov      dword ptr [rip + 0x113ce], 0
0x1a71e:      neg      eax
0x1a720:      ret
0x1a721:      nop      word ptr cs:[rax + rax]
0x1a72c:      nop      dword ptr [rax]
0x1a730:      endbr64
0x1a734:      push     r15
0x1a736:      push     r14
0x1a738:      mov      r14d, esi
0x1a73b:      push     r13
0x1a73d:      push     r12
0x1a73f:      mov      r12d, edx
0x1a742:      push     rbp
0x1a743:      push     rbx
0x1a744:      sub      rsp, 0xa8
0x1a74b:      mov      rax, qword ptr fs:[0x28]
0x1a754:      mov      qword ptr [rsp + 0x98], rax
0x1a75c:      xor      eax, eax
0x1a75e:      call     0x1a080
0x1a763:      xor      esi, esi
0x1a765:      mov      edx, 0x1b6
0x1a76a:      mov      rbp, rax
0x1a76d:      mov      rdi, rax
0x1a770:      xor      eax, eax
0x1a772:      call     0x4d80
0x1a777:      test     eax, eax
0x1a779:      js       0x1a80f
0x1a77f:      mov      rsi, rsp
0x1a782:      mov      edi, eax
0x1a784:      mov      r15d, eax
0x1a787:      call     0x18ef0
0x1a78c:      cmp      eax, -1

```

```

0x1a78f:      je      0x1a807
0x1a791:      mov     eax, dword ptr [rsp + 0x30]
0x1a795:      lea     edi, [rax + 1]
0x1a798:      movsxd  rdi, edi
0x1a79b:      call    0x1d670
0x1a7a0:      mov     rdx, qword ptr [rsp + 0x30]
0x1a7a5:      mov     edi, r15d
0x1a7a8:      mov     rsi, rax
0x1a7ab:      mov     r13, rax
0x1a7ae:      call    0x47a0
0x1a7b3:      cmp     qword ptr [rsp + 0x30], rax
0x1a7b8:      je      0x1a819
0x1a7ba:      mov     edi, r15d
0x1a7bd:      call    0x4ab0
0x1a7c2:      mov     rdi, rbp
0x1a7c5:      call    0x18d10
0x1a7ca:      test    r13, r13
0x1a7cd:      je      0x1a7d7
0x1a7cf:      mov     rdi, r13
0x1a7d2:      call    0x18d10
0x1a7d7:      call    0x4860
0x1a7dc:      mov     eax, dword ptr [rax]
0x1a7de:      mov     rdx, qword ptr [rsp + 0x98]
0x1a7e6:      sub     rdx, qword ptr fs:[0x28]
0x1a7ef:      jne     0x1a8f0
0x1a7f5:      add     rsp, 0xa8
0x1a7fc:      pop     rbx
0x1a7fd:      pop     rbp
0x1a7fe:      pop     r12
0x1a800:      pop     r13
0x1a802:      pop     r14
0x1a804:      pop     r15
0x1a806:      ret
0x1a807:      mov     edi, r15d
0x1a80a:      call    0x4ab0
0x1a80f:      mov     rdi, rbp
0x1a812:      call    0x18d10
0x1a817:      jmp     0x1a7d7
0x1a7d7:
0x1a819:      mov     edi, r15d
0x1a81c:      call    0x4ab0
0x1a821:      mov     rcx, qword ptr [rsp + 0x30]
0x1a826:      test    r12d, r12d
0x1a829:      jns     0x1a82e
0x1a82b:      mov     r12d, ecx
0x1a82e:      xor     ebx, ebx
0x1a830:      xor     esi, esi
0x1a832:      movsxd  rdx, esi

```



```

0x1a835:      cmp     ebx, r14d
0x1a838:      jge     0x1a885
0x1a83a:      cmp     rdx, rcx
0x1a83d:      jge     0x1a885
0x1a83f:      lea     eax, [rsi + 1]
0x1a842:      cdqe
0x1a844:      cmp     rdx, rcx
0x1a847:      jl      0x1a85c
0x1a849:      jmp     0x1a87b
0x1a87b:
0x1a84b:      nop     dword ptr [rax + rax]
0x1a850:      lea     rdx, [rax + 1]
0x1a854:      cmp     rax, rcx
0x1a857:      je      0x1a878
0x1a859:      mov     rax, rdx
0x1a85c:      cmp     byte ptr [r13 + rax - 1], 0xa
0x1a862:      jne     0x1a850
0x1a864:      add     ebx, 1
0x1a867:      movsxd  rsi, eax
0x1a86a:      cmp     r14d, ebx
0x1a86d:      je      0x1a832
0x1a86f:      lea     rdx, [rax + 1]
0x1a873:      cmp     rax, rcx
0x1a876:      jne     0x1a859
0x1a878:      movsxd  rdx, esi
0x1a87b:      cmp     rcx, rdx
0x1a87e:      jle     0x1a885
0x1a880:      cmp     r14d, ebx
0x1a883:      jg      0x1a83f
0x1a885:      lea     r14, [r13 + rdx]
0x1a88a:      lea     r15d, [rsi + 1]
0x1a88e:      cmp     rcx, rdx
0x1a891:      jg      0x1a8aa
0x1a893:      jmp     0x1a8d4
0x1a8d4:
0x1a895:      nop     dword ptr [rax]
0x1a898:      movsxd  rax, r15d
0x1a89b:      add     r14, 1
0x1a89f:      add     r15d, 1
0x1a8a3:      cmp     rax, qword ptr [rsp + 0x30]
0x1a8a8:      jge     0x1a8d4
0x1a8aa:      cmp     byte ptr [r14], 0xa
0x1a8ae:      jne     0x1a898
0x1a8b0:      lea     rdi, [r13 + rsi]
0x1a8b5:      mov     byte ptr [r14], 0
0x1a8b9:      cmp     byte ptr [rdi], 0
0x1a8bc:      jne     0x1a8cb
0x1a8be:      add     ebx, 1

```

```

0x1a8c1:      cmp     r12d, ebx
0x1a8c4:      jle     0x1a8d4
0x1a8c6:      movsxd  rsi, r15d
0x1a8c9:      jmp     0x1a898
0x1a898:
0x1a8cb:      xor     eax, eax
0x1a8cd:      call    0x1a380
0x1a8d2:      jmp     0x1a8be
0x1a8be:
0x1a8d4:      mov     rdi, rbp
0x1a8d7:      call    0x18d10
0x1a8dc:      test    r13, r13
0x1a8df:      je      0x1a8e9
0x1a8e1:      mov     rdi, r13
0x1a8e4:      call    0x18d10
0x1a8e9:      xor     eax, eax
0x1a8eb:      jmp     0x1a7de
0x1a7de:
0x1a8f0:      call    0x49d0
0x1a8f5:      nop     word ptr cs:[rax + rax]
0x1a900:      endbr64
0x1a904:      mov     edx, 0xffffffff
0x1a909:      xor     esi, esi
0x1a90b:      xor     eax, eax
0x1a90d:      jmp     0x1a730
0x1a730:
0x1a912:      nop     word ptr cs:[rax + rax]
0x1a91d:      nop     dword ptr [rax]
0x1a920:      endbr64
0x1a924:      push    r14
0x1a926:      push    r13
0x1a928:      push    r12
0x1a92a:      mov     r12d, esi
0x1a92d:      push    rbp
0x1a92e:      push    rbx
0x1a92f:      sub     rsp, 0xa0
0x1a936:      mov     rax, qword ptr fs:[0x28]
0x1a93f:      mov     qword ptr [rsp + 0x98], rax
0x1a947:      xor     eax, eax
0x1a949:      call    0x1a080
0x1a94e:      xor     esi, esi
0x1a950:      mov     edx, 0x1b6
0x1a955:      mov     rbp, rax
0x1a958:      mov     rdi, rax
0x1a95b:      xor     eax, eax
0x1a95d:      call    0x4d80
0x1a962:      cmp     eax, -1
0x1a965:      je      0x1aa58

```

```

0x1a96b:    mov     rsi, rsp
0x1a96e:    mov     edi, eax
0x1a970:    mov     r14d, eax
0x1a973:    call    0x18ef0
0x1a978:    cmp     eax, -1
0x1a97b:    je      0x1aa58
0x1a981:    mov     eax, dword ptr [rsp + 0x30]
0x1a985:    lea     edi, [rax + 1]
0x1a988:    movsxd  rdi, edi
0x1a98b:    call    0x1d670
0x1a990:    mov     rdx, qword ptr [rsp + 0x30]
0x1a995:    mov     edi, r14d
0x1a998:    mov     rsi, rax
0x1a99b:    mov     r13, rax
0x1a99e:    call    0x47a0
0x1a9a3:    mov     edi, r14d
0x1a9a6:    mov     rbx, rax
0x1a9a9:    call    0x4ab0
0x1a9ae:    test    ebx, ebx
0x1a9b0:    jle     0x1aa88
0x1a9b6:    lea     eax, [rbx - 1]
0x1a9b9:    cmp     ebx, 1
0x1a9bc:    je      0x1a9f3
0x1a9be:    test    r12d, r12d
0x1a9c1:    je      0x1a9f3
0x1a9c3:    lea     edx, [rbx - 2]
0x1a9c6:    movsxd  rdx, edx
0x1a9c9:    nop     dword ptr [rax]
0x1a9d0:    cmp     byte ptr [r13 + rdx + 1], 0xa
0x1a9d6:    jne     0x1a9dc
0x1a9d8:    sub     r12d, 1
0x1a9dc:    test    r12d, r12d
0x1a9df:    mov     eax, edx
0x1a9e1:    setne   sil
0x1a9e5:    test    edx, edx
0x1a9e7:    setne   cl
0x1a9ea:    sub     rdx, 1
0x1a9ee:    test    sil, cl
0x1a9f1:    jne     0x1a9d0
0x1a9f3:    test    eax, eax
0x1a9f5:    je      0x1aa88
0x1a9fb:    cdqe
0x1a9fd:    jmp     0x1aa08
0x1aa08:
0x1a9ff:    nop
0x1aa00:    sub     rax, 1
0x1aa04:    test    eax, eax
0x1aa06:    je      0x1aa50

```

```

0x1aa08:    cmp     byte ptr [r13 + rax], 0xa
0x1aa0e:    mov     ebx, eax
0x1aa10:    jne     0x1aa00
0x1aa12:    mov     edx, 0x1b6
0x1aa17:    mov     esi, 0x201
0x1aa1c:    mov     rdi, rbp
0x1aa1f:    xor     eax, eax
0x1aa21:    call    0x4d80
0x1aa26:    mov     r12d, eax
0x1aa29:    cmp     eax, -1
0x1aa2c:    je      0x1aa50
0x1aa2e:    mov     rdx, qword ptr [rsp + 0x30]
0x1aa33:    lea     esi, [rbx + 1]
0x1aa36:    mov     edi, eax
0x1aa38:    movsxd  rsi, esi
0x1aa3b:    sub     rdx, rsi
0x1aa3e:    add     rsi, r13
0x1aa41:    call    0x4790
0x1aa46:    mov     edi, r12d
0x1aa49:    call    0x4ab0
0x1aa4e:    nop
0x1aa50:    mov     rdi, r13
0x1aa53:    call    0x18d10
0x1aa58:    mov     rdi, rbp
0x1aa5b:    call    0x18d10
0x1aa60:    mov     rax, qword ptr [rsp + 0x98]
0x1aa68:    sub     rax, qword ptr fs:[0x28]
0x1aa71:    jne     0x1aa8f
0x1aa73:    add     rsp, 0xa0
0x1aa7a:    xor     eax, eax
0x1aa7c:    pop     rbx
0x1aa7d:    pop     rbp
0x1aa7e:    pop     r12
0x1aa80:    pop     r13
0x1aa82:    pop     r14
0x1aa84:    ret
0x1aa85:    nop     dword ptr [rax]
0x1aa88:    test    r13, r13
0x1aa8b:    je      0x1aa58
0x1aa8d:    jmp     0x1aa50
0x1aa50:
0x1aa8f:    call    0x49d0
0x1aa94:    nop     word ptr cs:[rax + rax]
0x1aa9f:    nop
0x1aaa0:    endbr64
0x1aaa4:    mov     r8d, edi
0x1aaa7:    xor     edx, edx
0x1aaa9:    mov     rdi, rsi

```

```

0x1aaac:      xor     eax, eax
0x1aaaee:     mov     esi, r8d
0x1aab1:      jmp     0x1a130
0x1a130:
0x1aab6:      nop     word ptr cs:[rax + rax]
0x1aac0:      endbr64
0x1aac4:      mov     esi, dword ptr [rip + 0x1101a]
0x1aaca:      mov     edx, 1
0x1aacf:      xor     eax, eax
0x1aad1:      jmp     0x1a130
0x1a130:
0x1aad6:      nop     word ptr cs:[rax + rax]
0x1aae0:      endbr64
0x1aae4:      movsxd   rdx, dword ptr [rip + 0x11011]
0x1aaeb:      cmp     edx, dword ptr [rip + 0x10ff3]
0x1aaf1:      je      0x1ab08
0x1aaf3:      mov     rax, qword ptr [rip + 0x11006]
0x1aafa:      test    rax, rax
0x1aafd:      je      0x1ab0a
0x1aaff:      mov     rax, qword ptr [rax + rdx*8]
0x1ab03:      ret
0x1ab04:      nop     dword ptr [rax]
0x1ab08:      xor     eax, eax
0x1ab0a:      ret
0x1ab0b:      nop     dword ptr [rax + rax]
0x1ab10:      endbr64
0x1ab14:      mov     eax, dword ptr [rip + 0x10fe2]
0x1ab1a:      xor     r8d, r8d
0x1ab1d:      test    eax, eax
0x1ab1f:      je      0x1ab37
0x1ab21:      sub     eax, 1
0x1ab24:      mov     rdx, qword ptr [rip + 0x10fd5]
0x1ab2b:      mov     dword ptr [rip + 0x10fcb], eax
0x1ab31:      cdqe
0x1ab33:      mov     r8, qword ptr [rdx + rax*8]
0x1ab37:      mov     rax, r8
0x1ab3a:      ret
0x1ab3b:      nop     dword ptr [rax + rax]
0x1ab40:      endbr64
0x1ab44:      mov     eax, dword ptr [rip + 0x10fb2]
0x1ab4a:      cmp     eax, dword ptr [rip + 0x10f94]
0x1ab50:      je      0x1ab70
0x1ab52:      add     eax, 1
0x1ab55:      mov     rdx, qword ptr [rip + 0x10fa4]
0x1ab5c:      mov     dword ptr [rip + 0x10f9a], eax
0x1ab62:      cdqe
0x1ab64:      mov     rax, qword ptr [rdx + rax*8]
0x1ab68:      ret

```

```

0x1ab69:    nop     dword ptr [rax]
0x1ab70:    xor     eax, eax
0x1ab72:    ret
0x1ab73:    nop     word ptr cs:[rax + rax]
0x1ab7e:    nop
0x1ab80:    endbr64
0x1ab84:    mov     rax, qword ptr [rip + 0x10f75]
0x1ab8b:    ret
0x1ab8c:    nop     dword ptr [rax]
0x1ab90:    endbr64
0x1ab94:    sub     edi, dword ptr [rip + 0xfc4e]
0x1ab9a:    cmp     dword ptr [rip + 0x10f44], edi
0x1aba0:    jle     0x1abc0
0x1aba2:    test    edi, edi
0x1aba4:    js      0x1abc0
0x1aba6:    mov     rax, qword ptr [rip + 0x10f53]
0x1abad:    test    rax, rax
0x1abb0:    je      0x1abc2
0x1abb2:    movsxd  rdi, edi
0x1abb5:    mov     rax, qword ptr [rax + rdi*8]
0x1abb9:    ret
0x1abba:    nop     word ptr [rax + rax]
0x1abc0:    xor     eax, eax
0x1abc2:    ret
0x1abc3:    nop     word ptr cs:[rax + rax]
0x1abce:    nop
0x1abd0:    endbr64
0x1abd4:    cmp     edx, dword ptr [rip + 0x10f0a]
0x1abda:    push    rbx
0x1abdb:    mov     ebx, dword ptr [rip + 0x10f1b]
0x1abe1:    jg      0x1abf7
0x1abe3:    test    edx, edx
0x1abe5:    js      0x1abf7
0x1abe7:    cmp     qword ptr [rip + 0x10f11], 0
0x1abef:    je      0x1abf7
0x1abf1:    mov     dword ptr [rip + 0x10f05], edx
0x1abf7:    xor     edx, edx
0x1abf9:    xor     eax, eax
0x1abfb:    call    0x19de0
0x1ac00:    mov     edx, ebx
0x1ac02:    shr     edx, 0x1f
0x1ac05:    cmp     eax, -1
0x1ac08:    je      0x1ac10
0x1ac0a:    mov     eax, dword ptr [rip + 0x10eec]
0x1ac10:    cmp     ebx, dword ptr [rip + 0x10ece]
0x1ac16:    jg      0x1ac2c
0x1ac18:    test    dl, dl
0x1ac1a:    jne     0x1ac2c

```

```

0x1ac1c:    cmp     qword ptr [rip + 0x10edc], 0
0x1ac24:    je      0x1ac2c
0x1ac26:    mov     dword ptr [rip + 0x10ed0], ebx
0x1ac2c:    pop     rbx
0x1ac2d:    ret
0x1ac2e:    nop
0x1ac30:    endbr64
0x1ac34:    cmp     dword ptr [rip + 0x10eaa], edi
0x1ac3a:    jl      0x1ac60
0x1ac3c:    test    edi, edi
0x1ac3e:    js      0x1ac60
0x1ac40:    cmp     qword ptr [rip + 0x10eb8], 0
0x1ac48:    je      0x1ac60
0x1ac4a:    mov     dword ptr [rip + 0x10eac], edi
0x1ac50:    mov     eax, 1
0x1ac55:    ret
0x1ac56:    nop     word ptr cs:[rax + rax]
0x1ac60:    xor     eax, eax
0x1ac62:    ret
0x1ac63:    nop     word ptr cs:[rax + rax]
0x1ac6e:    nop
0x1ac70:    endbr64
0x1ac74:    movsxd  rax, dword ptr [rsi]
0x1ac77:    movzx   r8d, byte ptr [rip + 0xfb6e]
0x1ac7f:    cmp     byte ptr [rdi + rax], r8b
0x1ac83:    jne     0x1afd0
0x1ac89:    push    r15
0x1ac8b:    lea     ecx, [rax + 1]
0x1ac8e:    mov     r9d, edx
0x1ac91:    mov     rdx, rax
0x1ac94:    push    r14
0x1ac96:    movsxd  r10, ecx
0x1ac99:    push    r13
0x1ac9b:    mov     r13, rsi
0x1ac9e:    lea     rsi, [rdi + r10]
0x1aca2:    push    r12
0x1aca4:    push    rbp
0x1aca5:    push    rbx
0x1aca6:    sub     rsp, 0x18
0x1acaa:    movzx   eax, byte ptr [rsi]
0x1acad:    cmp     r8b, al
0x1acb0:    je      0x1af20
0x1acb6:    mov     r8d, 1
0x1acbc:    cmp     al, 0x2d
0x1acbe:    je      0x1aef8
0x1acc4:    lea     edx, [rax - 0x30]
0x1acc7:    cmp     dl, 9
0x1acca:    jbe     0x1ae68

```

```

0x1acd0:    mov     ebp, 1
0x1acd5:    xor     ebx, ebx
0x1acd7:    cmp     al, 0x3f
0x1acd9:    jne     0x1acef
0x1acdb:    add     ecx, 1
0x1acde:    xor     ebp, ebp
0x1ace0:    mov     ebx, 1
0x1ace5:    movsxd  r10, ecx
0x1ace8:    lea     rsi, [rdi + r10]
0x1acec:    movzx   eax, byte ptr [rsi]
0x1acef:    test    al, al
0x1acf1:    je      0x1b042
0x1acf7:    lea     rdx, [rdi + r10 + 1]
0x1acfc:    mov     r12d, ecx
0x1acff:    mov     r8, rsi
0x1ad02:    movabs  r10, 0x400000100000200
0x1ad0c:    jmp     0x1ad48
0x1ad48:
0x1ad0e:    nop
0x1ad10:    cmp     al, 0x3a
0x1ad12:    ja      0x1ad1e
0x1ad14:    bt      r10, rax
0x1ad18:    jb      0x1afd8
0x1ad1e:    movsx   edi, al
0x1ad21:    cmp     edi, r9d
0x1ad24:    je      0x1b010
0x1ad2a:    cmp     al, 0xa
0x1ad2c:    je      0x1ae10
0x1ad32:    mov     r8, rdx
0x1ad35:    movzx   eax, byte ptr [rdx]
0x1ad38:    lea     rdx, [rdx + 1]
0x1ad3c:    add     r12d, 1
0x1ad40:    test    al, al
0x1ad42:    je      0x1ae10
0x1ad48:    test    ebx, ebx
0x1ad4a:    je      0x1ad10
0x1ad4c:    cmp     al, 0xa
0x1ad4e:    je      0x1ae10
0x1ad54:    cmp     al, 0x3f
0x1ad56:    jne     0x1ad32
0x1ad58:    mov     r15d, r12d
0x1ad5b:    mov     qword ptr [rsp + 8], r8
0x1ad60:    sub     r15d, ecx
0x1ad63:    mov     qword ptr [rsp], rsi
0x1ad67:    lea     edi, [r15 + 1]
0x1ad6b:    movsxd  r15, r15d
0x1ad6e:    movsxd  rdi, edi
0x1ad71:    call    0x1d670

```



```

0x1ad76:      mov     rsi, qword ptr [rsp]
0x1ad7a:      mov     rdx, r15
0x1ad7d:      mov     rdi, rax
0x1ad80:      mov     r14, rax
0x1ad83:      call    0x4880
0x1ad88:      mov     r8, qword ptr [rsp + 8]
0x1ad8d:      xor     eax, eax
0x1ad8f:      mov     byte ptr [r14 + r15], 0
0x1ad94:      cmp     byte ptr [r8], 0x3f
0x1ad98:      sete    al
0x1ad9b:      add     r12d, eax
0x1ad9e:      nop
0x1ada0:      mov     dword ptr [r13], r12d
0x1ada4:      mov     r12d, ebx
0x1ada7:      and     r12d, 1
0x1adab:      jmp     0x1add3
0x1add3:
0x1adad:      nop     dword ptr [rax]
0x1adb0:      movsxd  rax, dword ptr [rip + 0x10d45]
0x1adb7:      je      0x1af78
0x1adb d:      test    r12b, r12b
0x1adc0:      jne     0x1af78
0x1adc6:      test    eax, eax
0x1adc8:      je      0x1ade8
0x1adca:      sub     eax, 1
0x1adcd:      mov     dword ptr [rip + 0x10d29], eax
0x1add3:      xor     eax, eax
0x1add5:      mov     edx, ebp
0x1add7:      mov     esi, 0xffffffff
0x1addc:      mov     rdi, r14
0x1addf:      call    0x19de0
0x1ade4:      test    eax, eax
0x1ade6:      jns     0x1adb0
0x1ade8:      mov     eax, dword ptr [rip + 0x10cf6]
0x1adee:      mov     rdi, r14
0x1adf1:      mov     dword ptr [rip + 0x10d05], eax
0x1adf7:      call    0x18d10
0x1adfc:      add     rsp, 0x18
0x1ae00:      xor     eax, eax
0x1ae02:      pop     rbx
0x1ae03:      pop     rbp
0x1ae04:      pop     r12
0x1ae06:      pop     r13
0x1ae08:      pop     r14
0x1ae0a:      pop     r15
0x1ae0c:      ret
0x1ae0d:      nop     dword ptr [rax]
0x1ae10:      mov     r15d, r12d

```

```

0x1ae13:    sub     r15d, ecx
0x1ae16:    lea     edi, [r15 + 1]
0x1ae1a:    movsxd  r15, r15d
0x1ae1d:    movsxd  rdi, edi
0x1ae20:    mov     qword ptr [rsp + 8], r8
0x1ae25:    mov     qword ptr [rsp], rsi
0x1ae29:    call    0x1d670
0x1ae2e:    mov     rsi, qword ptr [rsp]
0x1ae32:    mov     rdx, r15
0x1ae35:    mov     rdi, rax
0x1ae38:    mov     r14, rax
0x1ae3b:    call    0x4880
0x1ae40:    test    ebx, ebx
0x1ae42:    mov     byte ptr [r14 + r15], 0
0x1ae47:    mov     r8, qword ptr [rsp + 8]
0x1ae4c:    je      0x1ada0
0x1ae52:    xor     eax, eax
0x1ae54:    cmp     byte ptr [r8], 0x3f
0x1ae58:    sete    al
0x1ae5b:    add     r12d, eax
0x1ae5e:    jmp     0x1ada0
0x1ada0:
0x1ae63:    nop     dword ptr [rax + rax]
0x1ae68:    lea     edx, [rcx + 1]
0x1ae6b:    xor     ecx, ecx
0x1ae6d:    movsxd  rdx, edx
0x1ae70:    sub     eax, 0x30
0x1ae73:    lea     ecx, [rcx + rcx*4]
0x1ae76:    mov     r9d, edx
0x1ae79:    movsx   eax, al
0x1ae7c:    lea     ecx, [rax + rcx*2]
0x1ae7f:    movzx   eax, byte ptr [rdi + rdx]
0x1ae83:    add     rdx, 1
0x1ae87:    lea     esi, [rax - 0x30]
0x1ae8a:    cmp     sil, 9
0x1ae8e:    jbe     0x1ae70
0x1ae90:    mov     dword ptr [r13], r9d
0x1ae94:    mov     eax, dword ptr [rip + 0x10c4a]
0x1ae9a:    mov     edx, dword ptr [rip + 0xf948]
0x1aea0:    cmp     r8d, -1
0x1aea4:    jne     0x1aead
0x1aea6:    lea     esi, [rax + rdx]
0x1aea9:    sub     esi, ecx
0x1aeab:    mov     ecx, esi
0x1aead:    sub     ecx, edx
0x1aeaf:    cmp     ecx, eax
0x1aeb1:    jge     0x1adfc
0x1aeb7:    test    ecx, ecx

```

```

0x1aeb9:      js      0x1adfc
0x1aebf:      mov     rax, qword ptr [rip + 0x10c3a]
0x1aec6:      test    rax, rax
0x1aec9:      je       0x1adfc
0x1aecf:      movsxd   rcx, ecx
0x1aed2:      mov     rax, qword ptr [rax + rcx*8]
0x1aed6:      test    rax, rax
0x1aed9:      je       0x1adfc
0x1aedef:     mov     rax, qword ptr [rax]
0x1aee2:      add     rsp, 0x18
0x1aee6:      pop      rbx
0x1aee7:      pop      rbp
0x1aee8:      pop      r12
0x1aeea:      pop      r13
0x1aeec:      pop      r14
0x1aeef:      pop      r15
0x1aef0:      ret
0x1aef1:      nop      dword ptr [rax]
0x1aef8:      lea      ecx, [rdx + 2]
0x1aefb:      mov     r8d, 0xffffffff
0x1af01:      movsxd   r10, ecx
0x1af04:      lea      rsi, [rdi + r10]
0x1af08:      movzx    eax, byte ptr [rsi]
0x1af0b:      lea      edx, [rax - 0x30]
0x1af0e:      cmp     dl, 9
0x1af11:      ja       0x1acd0
0x1af17:      jmp      0x1ae68
0x1ae68
0x1af1c:      nop      dword ptr [rax]
0x1af20:      mov     ecx, dword ptr [rip + 0x10bbe]
0x1af26:      mov     eax, dword ptr [rip + 0xf8bc]
0x1af2c:      add     edx, 2
0x1af2f:      mov     dword ptr [r13], edx
0x1af33:      lea     eax, [rcx + rax - 1]
0x1af37:      sub     eax, dword ptr [rip + 0xf8ab]
0x1af3d:      cmp     eax, dword ptr [rip + 0x10ba1]
0x1af43:      jge      0x1adfc
0x1af49:      test    eax, eax
0x1af4b:      js       0x1adfc
0x1af51:      mov     rdx, qword ptr [rip + 0x10ba8]
0x1af58:      test    rdx, rdx
0x1af5b:      je       0x1adfc
0x1af61:      cdqe
0x1af63:      mov     rax, qword ptr [rdx + rax*8]
0x1af67:      test    rax, rax
0x1af6a:      jne      0x1aedef
0x1af70:      jmp      0x1adfc
0x1adfc

```

```

0x1af75:      nop      dword ptr [rax]
0x1af78:      mov      edx, dword ptr [rip + 0x10b66]
0x1af7e:      cmp      edx, eax
0x1af80:      je       0x1b03b
0x1af86:      mov      rbp, qword ptr [rip + 0x10b73]
0x1af8d:      test     rbp, rbp
0x1af90:      je       0x1af97
0x1af92:      mov      rbp, qword ptr [rbp + rax*8]
0x1af97:      mov      dword ptr [rip + 0x10b5f], edx
0x1af9d:      test     ebx, ebx
0x1af9f:      je       0x1b02e
0x1afa5:      mov      rdi, qword ptr [rip + 0x10b44]
0x1afac:      test     rdi, rdi
0x1afaf:      je       0x1afb6
0x1afb1:      call     0x18d10
0x1afb6:      mov      qword ptr [rip + 0x10b33], r14
0x1afbd:      mov      rax, qword ptr [rbp]
0x1afc1:      add      rsp, 0x18
0x1afc5:      pop      rbx
0x1afc6:      pop      rbp
0x1afc7:      pop      r12
0x1afc9:      pop      r13
0x1afcb:      pop      r14
0x1afcd:      pop      r15
0x1afcf:      ret
0x1afd0:      xor      eax, eax
0x1afd2:      ret
0x1afd3:      nop      dword ptr [rax + rax]
0x1afd8:      mov      eax, r12d
0x1afdb:      mov      qword ptr [rsp], rsi
0x1afdf:      sub      eax, ecx
0x1afe1:      movsxd   r15, eax
0x1afe4:      add      eax, 1
0x1afe7:      movsxd   rdi, eax
0x1afea:      call     0x1d670
0x1afef:      mov      r14, rax
0x1aff2:      mov      rsi, qword ptr [rsp]
0x1aff6:      mov      rdx, r15
0x1aff9:      mov      rdi, r14
0x1affc:      call     0x4880
0x1b001:      mov      byte ptr [r14 + r15], 0
0x1b006:      jmp      0x1ada0
0x1ada0:
0x1b00b:      nop      dword ptr [rax + rax]
0x1b010:      mov      r15d, r12d
0x1b013:      mov      qword ptr [rsp], rsi
0x1b017:      sub      r15d, ecx
0x1b01a:      lea      edi, [r15 + 1]

```

```

0x1b01e:      movsxd  r15, r15d
0x1b021:      movsxd  rdi, edi
0x1b024:      call    0x1d670
0x1b029:      mov     r14, rax
0x1b02c:      jmp     0x1aff2
0x1aff2
0x1b02e:      mov     rdi, r14
0x1b031:      call    0x18d10
0x1b036:      jmp     0x1afbd
0x1afbd
0x1b03b:      xor     ebp, ebp
0x1b03d:      jmp     0x1af97
0x1af97
0x1b042:      mov     r12d, ecx
0x1b045:      mov     r8, rsi
0x1b048:      xor     r15d, r15d
0x1b04b:      mov     edi, 1
0x1b050:      jmp     0x1ae20
0x1ae20
0x1b055:      nop     word ptr cs:[rax + rax]
0x1b060:      endbr64
0x1b064:      push    r15
0x1b066:      push    r14
0x1b068:      push    r13
0x1b06a:      push    r12
0x1b06c:      push    rbp
0x1b06d:      push    rbx
0x1b06e:      sub     rsp, 0x28
0x1b072:      movzx   r14d, byte ptr [rdi]
0x1b076:      test    r14b, r14b
0x1b079:      je      0x1b30d
0x1b07f:      mov     qword ptr [rsp], 0
0x1b087:      mov     r12, rdi
0x1b08a:      mov     r13, rdi
0x1b08d:      xor     ebx, ebx
0x1b08f:      mov     dword ptr [rsp + 8], 0
0x1b097:      xor     ebp, ebp
0x1b099:      mov     dword ptr [rsp + 0xc], 0
0x1b0a1:      jmp     0x1b0c0
0x1b0c0
0x1b0a3:      nop     dword ptr [rax + rax]
0x1b0a8:      add     ebx, 1
0x1b0ab:      movsxd  rbp, ebx
0x1b0ae:      lea     r13, [r12 + rbp]
0x1b0b2:      movzx   r14d, byte ptr [r13]
0x1b0b7:      test    r14b, r14b
0x1b0ba:      je      0x1b315
0x1b0c0:      cmp     r14b, 0x20

```

```

0x1b0c4:      je      0x1b0a8
0x1b0c6:      cmp     r14b, 9
0x1b0ca:      je      0x1b0a8
0x1b0cc:      cmp     byte ptr [rip + 0x10a0d], r14b
0x1b0d3:      je      0x1b315
0x1b0d9:      movsx   r15d, r14b
0x1b0dd:      lea      rdi, [rip + 0x7379]
0x1b0e4:      mov     esi, r15d
0x1b0e7:      call    0x4a00
0x1b0ec:      test    rax, rax
0x1b0ef:      je      0x1b190
0x1b0f5:      add     ebx, 1
0x1b0f8:      mov     r14d, 1
0x1b0fe:      mov     r9d, 2
0x1b104:      movsxd   rbp, ebx
0x1b107:      lea      rcx, [r12 + rbp]
0x1b10b:      mov     eax, dword ptr [rsp + 8]
0x1b10f:      add     eax, 2
0x1b112:      cmp     eax, dword ptr [rsp + 0xc]
0x1b116:      jge     0x1b2a0
0x1b11c:      movsxd   r8, dword ptr [rsp + 8]
0x1b121:      mov     rdx, qword ptr [rsp]
0x1b125:      mov     rdi, r9
0x1b128:      mov     qword ptr [rsp + 0x18], rcx
0x1b12d:      shl     r8, 3
0x1b131:      lea      r15, [rdx + r8]
0x1b135:      mov     qword ptr [rsp + 0x10], r8
0x1b13a:      call    0x1d670
0x1b13f:      mov     rdx, r14
0x1b142:      mov     rsi, r13
0x1b145:      mov     qword ptr [r15], rax
0x1b148:      mov     rdi, rax
0x1b14b:      call    0x4880
0x1b150:      mov     rax, qword ptr [r15]
0x1b153:      mov     r8, qword ptr [rsp + 0x10]
0x1b158:      mov     rcx, qword ptr [rsp + 0x18]
0x1b15d:      add     dword ptr [rsp + 8], 1
0x1b162:      mov     byte ptr [rax + r14], 0
0x1b167:      mov     rax, qword ptr [rsp]
0x1b16b:      mov     qword ptr [rax + r8 + 8], 0
0x1b174:      movzx   r14d, byte ptr [rcx]
0x1b178:      test    r14b, r14b
0x1b17b:      je      0x1b315
0x1b181:      mov     r13, rcx
0x1b184:      jmp     0x1b0c0
0x1b0c0:
0x1b189:      nop     dword ptr [rax]
0x1b190:      mov     esi, r15d

```

```

0x1b193:    lea     rdi, [rip + 0x72c7]
0x1b19a:    call    0x4a00
0x1b19f:    test    rax, rax
0x1b1a2:    je       0x1b369
0x1b1a8:    movsx    edx, byte ptr [r12 + rbp + 1]
0x1b1ae:    cmp      dl, r14b
0x1b1b1:    je       0x1b2d8
0x1b1b7:    cmp      edx, 0x26
0x1b1ba:    je       0x1b340
0x1b1c0:    cmp      edx, 0x3e
0x1b1c3:    jne      0x1b3ac
0x1b1c9:    cmp      r14b, 0x26
0x1b1cd:    je       0x1b34e
0x1b1d3:    cmp      r14b, 0x24
0x1b1d7:    jne      0x1b0f5
0x1b1dd:    mov      edx, ebx
0x1b1df:    xor      r15d, r15d
0x1b1e2:    movsxd   r8, edx
0x1b1e5:    lea      rcx, [r12 + r8]
0x1b1e9:    mov      rbp, r8
0x1b1ec:    movzx    eax, byte ptr [rcx]
0x1b1ef:    test     al, al
0x1b1f1:    je       0x1b404
0x1b1f7:    mov      qword ptr [rsp + 0x10], r13
0x1b1fc:    lea      r14, [rip + 0x7265]
0x1b203:    mov      r13, r8
0x1b206:    mov      dword ptr [rsp + 0x18], ebx
0x1b20a:    mov      rbx, r12
0x1b20d:    mov      r12d, edx
0x1b210:    jmp      0x1b265
0x1b265:
0x1b212:    nop      word ptr [rax + rax]
0x1b218:    movsx    ebp, al
0x1b21b:    test     r15d, r15d
0x1b21e:    jne      0x1b290
0x1b220:    mov      esi, ebp
0x1b222:    lea      rdi, [rip + 0x7243]
0x1b229:    call     0x4a00
0x1b22e:    test     rax, rax
0x1b231:    jne      0x1b3c0
0x1b237:    mov      esi, ebp
0x1b239:    mov      rdi, r14
0x1b23c:    call     0x4a00
0x1b241:    lea      esi, [r12 + 1]
0x1b246:    test     rax, rax
0x1b249:    cmovne   r15d, ebp
0x1b24d:    movsxd   rbp, esi
0x1b250:    mov      r12d, esi

```

```

0x1b253:    lea     rcx, [rbx + rbp]
0x1b257:    mov     r13, rbp
0x1b25a:    movzx   eax, byte ptr [rcx]
0x1b25d:    test    al, al
0x1b25f:    je      0x1b389
0x1b265:    cmp     al, 0x5c
0x1b267:    jne     0x1b218
0x1b269:    cmp     byte ptr [rbx + r13 + 1], 0xa
0x1b26f:    lea     esi, [r12 + 1]
0x1b274:    je      0x1b330
0x1b27a:    xor     eax, eax
0x1b27c:    cmp     r15d, 0x27
0x1b280:    setne   al
0x1b283:    add     esi, eax
0x1b285:    jmp     0x1b24d
0x1b24d:
0x1b287:    nop     word ptr [rax + rax]
0x1b290:    xor     eax, eax
0x1b292:    cmp     r15d, ebp
0x1b295:    lea     esi, [r12 + 1]
0x1b29a:    cmov     r15d, eax
0x1b29e:    jmp     0x1b24d
0x1b24d:
0x1b2a0:    add     dword ptr [rsp + 0xc], 0xa
0x1b2a5:    movsxd  rsi, dword ptr [rsp + 0xc]
0x1b2aa:    mov     rdi, qword ptr [rsp]
0x1b2ae:    mov     qword ptr [rsp + 0x18], rcx
0x1b2b3:    shl     rsi, 3
0x1b2b7:    mov     qword ptr [rsp + 0x10], r9
0x1b2bc:    call    0x1d690
0x1b2c1:    mov     rcx, qword ptr [rsp + 0x18]
0x1b2c6:    mov     r9, qword ptr [rsp + 0x10]
0x1b2cb:    mov     qword ptr [rsp], rax
0x1b2cf:    jmp     0x1b11c
0x1b11c:
0x1b2d4:    nop     dword ptr [rax]
0x1b2d8:    cmp     edx, 0x24
0x1b2db:    je      0x1b1d3
0x1b2e1:    mov     eax, ebx
0x1b2e3:    cmp     edx, 0x3c
0x1b2e6:    je      0x1b3f0
0x1b2ec:    add     eax, 2
0x1b2ef:    mov     r14d, eax
0x1b2f2:    movsxd  rbp, eax
0x1b2f5:    sub     r14d, ebx
0x1b2f8:    lea     rcx, [r12 + rbp]
0x1b2fc:    mov     ebx, eax
0x1b2fe:    lea     r9d, [r14 + 1]

```



```

0x1b302:      movsxd  r14, r14d
0x1b305:      movsxd  r9, r9d
0x1b308:      jmp      0x1b10b
0x1b10b
0x1b30d:      mov      qword ptr [rsp], 0
0x1b315:      mov      rax, qword ptr [rsp]
0x1b319:      add      rsp, 0x28
0x1b31d:      pop      rbx
0x1b31e:      pop      rbp
0x1b31f:      pop      r12
0x1b321:      pop      r13
0x1b323:      pop      r14
0x1b325:      pop      r15
0x1b327:      ret
0x1b328:      nop      dword ptr [rax + rax]
0x1b330:      lea      esi, [r12 + 2]
0x1b335:      jmp      0x1b24d
0x1b24d
0x1b33a:      nop      word ptr [rax + rax]
0x1b340:      mov      eax, r14d
0x1b343:      and      eax, 0xffffffff
0x1b346:      cmp      al, 0x3c
0x1b348:      jne      0x1b1d3
0x1b34e:      add      ebx, 2
0x1b351:      mov      r14d, 2
0x1b357:      mov      r9d, 3
0x1b35d:      movsxd  rbp, ebx
0x1b360:      lea      rcx, [r12 + rbp]
0x1b364:      jmp      0x1b10b
0x1b10b
0x1b369:      mov      esi, r15d
0x1b36c:      lea      rdi, [rip + 0x70f5]
0x1b373:      call     0x4a00
0x1b378:      test     rax, rax
0x1b37b:      je       0x1b1dd
0x1b381:      lea      edx, [rbx + 1]
0x1b384:      jmp      0x1b1e2
0x1b1e2
0x1b389:      mov      r12, rbx
0x1b38c:      mov      ebx, dword ptr [rsp + 0x18]
0x1b390:      mov      r14d, esi
0x1b393:      mov      r13, qword ptr [rsp + 0x10]
0x1b398:      sub      r14d, ebx
0x1b39b:      mov      ebx, esi
0x1b39d:      lea      r9d, [r14 + 1]
0x1b3a1:      movsxd  r14, r14d
0x1b3a4:      movsxd  r9, r9d
0x1b3a7:      jmp      0x1b10b

```

```

0x1b10b
0x1b3ac:      cmp     r14b, 0x24
0x1b3b0:      jne     0x1b0f5
0x1b3b6:      cmp     edx, 0x28
0x1b3b9:      je      0x1b34e
0x1b3bb:      jmp     0x1b1d3
0x1b1d3
0x1b3c0:      mov     edx, r12d
0x1b3c3:      mov     r12, rbx
0x1b3c6:      mov     ebx, dword ptr [rsp + 0x18]
0x1b3ca:      mov     r8, r13
0x1b3cd:      mov     r14d, edx
0x1b3d0:      mov     r13, qword ptr [rsp + 0x10]
0x1b3d5:      mov     rbp, r8
0x1b3d8:      lea     rcx, [r12 + r8]
0x1b3dc:      sub     r14d, ebx
0x1b3df:      mov     ebx, edx
0x1b3e1:      lea     r9d, [r14 + 1]
0x1b3e5:      movsxd  r14, r14d
0x1b3e8:      movsxd  r9, r9d
0x1b3eb:      jmp     0x1b10b
0x1b10b
0x1b3f0:      cmp     byte ptr [r12 + rbp + 2], 0x2d
0x1b3f6:      jne     0x1b2ec
0x1b3fc:      lea     eax, [rbx + 1]
0x1b3ff:      jmp     0x1b2ec
0x1b2ec
0x1b404:      mov     r14d, edx
0x1b407:      sub     r14d, ebx
0x1b40a:      mov     ebx, edx
0x1b40c:      lea     r9d, [r14 + 1]
0x1b410:      movsxd  r14, r14d
0x1b413:      movsxd  r9, r9d
0x1b416:      jmp     0x1b10b
0x1b10b
0x1b41b:      nop     dword ptr [rax + rax]
0x1b420:      endbr64
0x1b424:      push    r15
0x1b426:      xor     eax, eax
0x1b428:      push    r14
0x1b42a:      push    r13
0x1b42c:      push    r12
0x1b42e:      mov     r12d, esi
0x1b431:      push    rbp
0x1b432:      push    rbx
0x1b433:      mov     ebx, edi
0x1b435:      mov     rdi, rdx
0x1b438:      sub     rsp, 8

```

```

0x1b43c:      call    0x1b060
0x1b441:      test   rax, rax
0x1b444:      je     0x1b598
0x1b44a:      cmp     qword ptr [rax], 0
0x1b44e:      mov     rbp, rax
0x1b451:      je     0x1b590
0x1b457:      mov     eax, 1
0x1b45c:      nop     dword ptr [rax]
0x1b460:      mov     r14d, eax
0x1b463:      add     rax, 1
0x1b467:      cmp     qword ptr [rbp + rax*8 - 8], 0
0x1b46d:      jne     0x1b460
0x1b46f:      lea     eax, [r14 + r12 - 1]
0x1b474:      test    r12d, r12d
0x1b477:      cmovs   r12d, eax
0x1b47b:      lea     eax, [r14 + rbx - 1]
0x1b480:      test    ebx, ebx
0x1b482:      cmovs   ebx, eax
0x1b485:      lea     eax, [r12 + 1]
0x1b48a:      cmp     r12d, 0x24
0x1b48e:      jne     0x1b497
0x1b490:      lea     r12d, [r14 - 1]
0x1b494:      mov     eax, r14d
0x1b497:      lea     edx, [r14 - 1]
0x1b49b:      cmp     ebx, 0x24
0x1b49e:      cmov     ebx, edx
0x1b4a1:      mov     edx, eax
0x1b4a3:      cmp     ebx, eax
0x1b4a5:      cmovge   edx, ebx
0x1b4a8:      xor     r13d, r13d
0x1b4ab:      cmp     edx, r14d
0x1b4ae:      jg      0x1b547
0x1b4b4:      mov     ecx, ebx
0x1b4b6:      or      ecx, eax
0x1b4b8:      js      0x1b547
0x1b4be:      cmp     ebx, eax
0x1b4c0:      jge     0x1b59d
0x1b4c6:      movsxd  rbx, ebx
0x1b4c9:      xor     r15d, r15d
0x1b4cc:      mov     r13, rbx
0x1b4cf:      nop
0x1b4d0:      mov     rdi, qword ptr [rbp + r13*8]
0x1b4d5:      add     r13, 1
0x1b4d9:      call    0x49b0
0x1b4de:      lea     r15d, [r15 + rax + 1]
0x1b4e3:      lea     eax, [r13 - 1]
0x1b4e7:      cmp     r12d, eax
0x1b4ea:      jg      0x1b4d0

```

```

0x1b4ec:    lea     edi, [r15 + 1]
0x1b4f0:    xor     r15d, r15d
0x1b4f3:    movsxd  rdi, edi
0x1b4f6:    call    0x1d670
0x1b4fb:    mov     r13, rax
0x1b4fe:    nop
0x1b500:    mov     rsi, qword ptr [rbp + rbx*8]
0x1b505:    movsxd  rdi, r15d
0x1b508:    add     rdi, r13
0x1b50b:    call    0x48b0
0x1b510:    mov     rdi, qword ptr [rbp + rbx*8]
0x1b515:    call    0x49b0
0x1b51a:    add     r15d, eax
0x1b51d:    cmp     r12d, ebx
0x1b520:    jle     0x1b53b
0x1b522:    lea     eax, [r15 + 1]
0x1b526:    movsxd  r15, r15d
0x1b529:    movsxd  rdx, eax
0x1b52c:    mov     byte ptr [r13 + r15], 0x20
0x1b532:    mov     r15d, eax
0x1b535:    mov     byte ptr [r13 + rdx], 0
0x1b53b:    add     rbx, 1
0x1b53f:    lea     eax, [rbx - 1]
0x1b542:    cmp     r12d, eax
0x1b545:    jg      0x1b500
0x1b547:    test    r14d, r14d
0x1b54a:    je      0x1b571
0x1b54c:    lea     eax, [r14 - 1]
0x1b550:    mov     rbx, rbp
0x1b553:    lea     r12, [rbp + rax*8 + 8]
0x1b558:    nop     dword ptr [rax + rax]
0x1b560:    mov     rdi, qword ptr [rbx]
0x1b563:    add     rbx, 8
0x1b567:    call    0x18d10
0x1b56c:    cmp     r12, rbx
0x1b56f:    jne     0x1b560
0x1b571:    mov     rdi, rbp
0x1b574:    call    0x18d10
0x1b579:    add     rsp, 8
0x1b57d:    mov     rax, r13
0x1b580:    pop     rbx
0x1b581:    pop     rbp
0x1b582:    pop     r12
0x1b584:    pop     r13
0x1b586:    pop     r14
0x1b588:    pop     r15
0x1b58a:    ret
0x1b58b:    nop     dword ptr [rax + rax]

```

```

0x1b590:    xor     r14d, r14d
0x1b593:    jmp     0x1b46f
0x1b46f:
0x1b598:    xor     r13d, r13d
0x1b59b:    jmp     0x1b579
0x1b579:
0x1b59d:    mov     edi, 1
0x1b5a2:    call    0x1d670
0x1b5a7:    mov     r13, rax
0x1b5aa:    jmp     0x1b547
0x1b547:
0x1b5ac:    nop     dword ptr [rax]
0x1b5b0:    endbr64
0x1b5b4:    push    r15
0x1b5b6:    push    r14
0x1b5b8:    push    r13
0x1b5ba:    push    r12
0x1b5bc:    push    rbp
0x1b5bd:    push    rbx
0x1b5be:    mov     rbx, rdi
0x1b5c1:    sub     rsp, 0x88
0x1b5c8:    mov     qword ptr [rsp + 0x20], rdi
0x1b5cd:    mov     qword ptr [rsp + 0x30], rsi
0x1b5d2:    mov     rax, qword ptr fs:[0x28]
0x1b5db:    mov     qword ptr [rsp + 0x78], rax
0x1b5e0:    xor     eax, eax
0x1b5e2:    cmp     byte ptr [rip + 0xf204], 0
0x1b5e9:    je      0x1b656
0x1b5eb:    mov     edi, 0x100
0x1b5f0:    call    0x1d670
0x1b5f5:    mov     rbx, qword ptr [rsp + 0x20]
0x1b5fa:    mov     byte ptr [rax], 0
0x1b5fd:    mov     r15, rax
0x1b600:    mov     rdi, rbx
0x1b603:    call    0x49b0
0x1b608:    mov     dword ptr [rsp + 0xc], eax
0x1b60c:    mov     r12, rax
0x1b60f:    movzx   eax, byte ptr [rbx]
0x1b612:    cmp     al, byte ptr [rip + 0xf1d4]
0x1b618:    je      0x1c1e0
0x1b61e:    movzx   ebp, byte ptr [rip + 0xf1c8]
0x1b625:    mov     r13, qword ptr [rip + 0xf1c4]
0x1b62c:    lea     rbx, [rbx + 1]
0x1b630:    test    al, al
0x1b632:    jne     0x1b6c7
0x1b638:    nop     dword ptr [rax + rax]
0x1b640:    test    bpl, bpl
0x1b643:    je      0x1b6d8

```

```

0x1b649:    mov     rdi, r15
0x1b64c:    call   0x18d10
0x1b651:    mov     rbx, qword ptr [rsp + 0x20]
0x1b656:    mov     rdi, rbx
0x1b659:    call   0x49b0
0x1b65e:    lea     rdi, [rax + 1]
0x1b662:    call   0x1d670
0x1b667:    mov     rsi, rbx
0x1b66a:    mov     rdi, rax
0x1b66d:    call   0x48b0
0x1b672:    mov     rcx, qword ptr [rsp + 0x30]
0x1b677:    mov     qword ptr [rcx], rax
0x1b67a:    xor     eax, eax
0x1b67c:    mov     rdx, qword ptr [rsp + 0x78]
0x1b681:    sub     rdx, qword ptr fs:[0x28]
0x1b68a:    jne     0x1c487
0x1b690:    add     rsp, 0x88
0x1b697:    pop     rbx
0x1b698:    pop     rbp
0x1b699:    pop     r12
0x1b69b:    pop     r13
0x1b69d:    pop     r14
0x1b69f:    pop     r15
0x1b6a1:    ret
0x1b6a2:    nop     word ptr [rax + rax]
0x1b6a8:    mov     rdi, r13
0x1b6ab:    call   0x4a00
0x1b6b0:    test    rax, rax
0x1b6b3:    je      0x1b6d8
0x1b6b5:    nop     dword ptr [rax]
0x1b6b8:    movzx   eax, byte ptr [rbx]
0x1b6bb:    add     rbx, 1
0x1b6bf:    test    al, al
0x1b6c1:    je      0x1b640
0x1b6c7:    movsx   esi, byte ptr [rbx]
0x1b6ca:    cmp     al, bpl
0x1b6cd:    jne     0x1b6b8
0x1b6cf:    test    esi, esi
0x1b6d1:    je      0x1b6b8
0x1b6d3:    jmp     0x1b6a8
0x1b6a8:
0x1b6d5:    nop     dword ptr [rax]
0x1b6d8:    mov     rbp, qword ptr [rsp + 0x20]
0x1b6dd:    test    r12d, r12d
0x1b6e0:    jle     0x1c40a
0x1b6e6:    mov     dword ptr [rsp + 4], 0x100
0x1b6ee:    xor     r13d, r13d
0x1b6f1:    xor     ebx, ebx

```

```

0x1b6f3:    mov     dword ptr [rsp + 0x28], 0
0x1b6fb:    mov     dword ptr [rsp + 0x3c], 0
0x1b703:    mov     dword ptr [rsp + 8], 0
0x1b70b:    jmp     0x1b758
0x1b758:
0x1b70d:    nop     dword ptr [rax]
0x1b710:    cmp     r12b, byte ptr [rip + 0xf0d6]
0x1b717:    je      0x1b7d0
0x1b71d:    cmp     r12b, 0xfd
0x1b721:    je      0x1b7d0
0x1b727:    cmp     r12b, 0x5c
0x1b72b:    je      0x1bde0
0x1b731:    cmp     ebx, dword ptr [rsp + 4]
0x1b735:    jge     0x1bfe0
0x1b73b:    lea     eax, [rbx + 1]
0x1b73e:    mov     byte ptr [r15 + rbx], r12b
0x1b742:    movsxd  rdx, eax
0x1b745:    movsxd  rbx, eax
0x1b748:    mov     byte ptr [r15 + rdx], 0
0x1b74d:    add     r13d, 1
0x1b751:    cmp     r13d, dword ptr [rsp + 0xc]
0x1b756:    jge     0x1b79d
0x1b758:    movsxd  r14, r13d
0x1b75b:    mov     r10d, dword ptr [rsp + 8]
0x1b760:    lea     rdx, [rbp + r14]
0x1b765:    movzx   r12d, byte ptr [rdx]
0x1b769:    test    r10d, r10d
0x1b76c:    je      0x1b710
0x1b76e:    cmp     ebx, dword ptr [rsp + 4]
0x1b772:    jge     0x1be10
0x1b778:    lea     eax, [rbx + 1]
0x1b77b:    mov     byte ptr [r15 + rbx], r12b
0x1b77f:    add     r13d, 1
0x1b783:    movsxd  rdx, eax
0x1b786:    mov     dword ptr [rsp + 8], 0
0x1b78e:    movsxd  rbx, eax
0x1b791:    mov     byte ptr [r15 + rdx], 0
0x1b796:    cmp     r13d, dword ptr [rsp + 0xc]
0x1b79b:    jl      0x1b758
0x1b79d:    mov     rax, qword ptr [rsp + 0x30]
0x1b7a2:    mov     qword ptr [rax], r15
0x1b7a5:    cmp     qword ptr [rsp + 0x20], rbp
0x1b7aa:    je      0x1b7b4
0x1b7ac:    mov     rdi, rbp
0x1b7af:    call    0x18d10
0x1b7b4:    mov     ecx, dword ptr [rsp + 0x28]
0x1b7b8:    test    ecx, ecx
0x1b7ba:    jne     0x1c35b

```

```

0x1b7c0:    mov     edx, dword ptr [rsp + 0x3c]
0x1b7c4:    xor     eax, eax
0x1b7c6:    test    edx, edx
0x1b7c8:    setne   al
0x1b7cb:    jmp     0x1b67c
0x1b67c:
0x1b7d0:    lea     rcx, [rbp + r14 + 1]
0x1b7d5:    mov     qword ptr [rsp + 0x10], rdx
0x1b7da:    movsx   esi, byte ptr [rcx]
0x1b7dd:    test    esi, esi
0x1b7df:    je      0x1b830
0x1b7e1:    mov     rdi, qword ptr [rip + 0xf008]
0x1b7e8:    mov     qword ptr [rsp + 0x18], rcx
0x1b7ed:    call    0x4a00
0x1b7f2:    mov     rcx, qword ptr [rsp + 0x18]
0x1b7f7:    test    rax, rax
0x1b7fa:    je      0x1b830
0x1b7fc:    cmp     ebx, dword ptr [rsp + 4]
0x1b800:    jl      0x1b73b
0x1b806:    add     dword ptr [rsp + 4], 0x40
0x1b80b:    movsxd  rsi, dword ptr [rsp + 4]
0x1b810:    mov     rdi, r15
0x1b813:    call    0x1d690
0x1b818:    mov     rdx, qword ptr [rsp + 0x10]
0x1b81d:    mov     r15, rax
0x1b820:    movzx   r12d, byte ptr [rdx]
0x1b824:    jmp     0x1b73b
0x1b73b:
0x1b829:    nop     dword ptr [rax]
0x1b830:    mov     edi, 0x80
0x1b835:    mov     qword ptr [rsp + 0x18], rcx
0x1b83a:    call    0x1d670
0x1b83f:    mov     rcx, qword ptr [rsp + 0x18]
0x1b844:    mov     dword ptr [rsp + 0x6c], r13d
0x1b849:    mov     qword ptr [rsp + 0x10], rax
0x1b84e:    movzx   r12d, byte ptr [rcx]
0x1b852:    test    r12b, r12b
0x1b855:    je      0x1be30
0x1b85b:    movsx   esi, r12b
0x1b85f:    lea     rdi, [rip + 0x6c11]
0x1b866:    call    0x4a00
0x1b86b:    test    rax, rax
0x1b86e:    je      0x1bfc0
0x1b874:    lea     eax, [r13 + 1]
0x1b878:    lea     rsi, [rsp + 0x70]
0x1b87d:    xor     edx, edx
0x1b87f:    mov     byte ptr [rsp + 0x77], 0
0x1b884:    mov     dword ptr [rsp + 0x6c], eax

```



```

0x1b888:    movzx    eax, byte ptr [rip + 0xef5e]
0x1b88f:    lea      rdi, [rsp + 0x75]
0x1b894:    mov     dword ptr [rsp + 0x70], 0
0x1b89c:    mov     byte ptr [rsp + 0x76], al
0x1b8a0:    mov     byte ptr [rsp + 0x75], al
0x1b8a4:    xor     eax, eax
0x1b8a6:    call    0x1ac70
0x1b8ab:    mov     r12d, dword ptr [rsp + 0x6c]
0x1b8b0:    mov     r14, rax
0x1b8b3:    test    r14, r14
0x1b8b6:    je      0x1c435
0x1b8bc:    movsxd   rax, r12d
0x1b8bf:    xor     esi, esi
0x1b8c1:    mov     edx, r12d
0x1b8c4:    movzx    eax, byte ptr [rbp + rax]
0x1b8c9:    cmp     al, 0x3a
0x1b8cb:    jne     0x1b8df
0x1b8cd:    lea     edx, [r12 + 1]
0x1b8d2:    mov     esi, 1
0x1b8d7:    movsxd   rax, edx
0x1b8da:    movzx    eax, byte ptr [rbp + rax]
0x1b8df:    cmp     al, 0x25
0x1b8e1:    je      0x1c010
0x1b8e7:    cmp     al, 0x2d
0x1b8e9:    je      0x1c100
0x1b8ef:    jg      0x1bd48
0x1b8f5:    cmp     al, 0x24
0x1b8f7:    je      0x1bf80
0x1b8fd:    cmp     al, 0x2a
0x1b8ff:    jne     0x1bd54
0x1b905:    add     edx, 1
0x1b908:    mov     esi, 0x24
0x1b90d:    mov     edi, 1
0x1b912:    xor     eax, eax
0x1b914:    mov     dword ptr [rsp + 0x6c], edx
0x1b918:    mov     rdx, r14
0x1b91b:    call    0x1b420
0x1b920:    mov     r13, rax
0x1b923:    test    rax, rax
0x1b926:    je      0x1c080
0x1b92c:    lea     rax, [rip + 0x101ed]
0x1b933:    cmp     r13, rax
0x1b936:    je      0x1bdae
0x1b93c:    mov     rdi, r13
0x1b93f:    call    0x49b0
0x1b944:    lea     rdi, [rax + 1]
0x1b948:    call    0x1d670
0x1b94d:    mov     rsi, r13

```

```

0x1b950:      mov     rdi, rax
0x1b953:      call    0x48b0
0x1b958:      mov     rdi, r13
0x1b95b:      mov     r14, rax
0x1b95e:      call    0x18d10
0x1b963:      movsxd   rcx, dword ptr [rsp + 0x6c]
0x1b968:      cmp     byte ptr [rbp + rcx], 0x3a
0x1b96d:      mov     dword ptr [rsp + 0x44], ecx
0x1b971:      jne     0x1c3d4
0x1b977:      mov     dword ptr [rsp + 0x18], 0
0x1b97f:      lea     r12, [rip + 0x6afe]
0x1b986:      mov     dword ptr [rsp + 0x38], 0
0x1b98e:      mov     dword ptr [rsp + 0x40], ebx
0x1b992:      mov     qword ptr [rsp + 0x28], r15
0x1b997:      mov     r15, r14
0x1b99a:      mov     r14d, ecx
0x1b99d:      nop     dword ptr [rax]
0x1b9a0:      movzx   eax, byte ptr [rbp + rcx + 1]
0x1b9a5:      cmp     al, 0x67
0x1b9a7:      jne     0x1b9bf
0x1b9a9:      add     r14d, 1
0x1b9ad:      mov     dword ptr [rsp + 0x18], 1
0x1b9b5:      mov     dword ptr [rsp + 0x6c], r14d
0x1b9ba:      movzx   eax, byte ptr [rbp + rcx + 2]
0x1b9bf:      cmp     al, 0x26
0x1b9c1:      je      0x1be70
0x1b9c7:      sub     eax, 0x65
0x1b9ca:      cmp     al, 0xf
0x1b9cc:      ja      0x1b9e0
0x1b9ce:      movzx   eax, al
0x1b9d1:      movsxd   rax, dword ptr [r12 + rax*4]
0x1b9d5:      add     rax, r12
0x1b9d8:      jmp     rax
rax
-->-->-->                                indirect JMP

0x74d8: ret
0x74d9: nop     dword ptr [rax]
0x74e0: lea     rdi, [rip + 0x23349]
0x74e7: lea     rsi, [rip + 0x23342]
0x74ee: sub     rsi, rdi
0x74f1: mov     rax, rsi
0x74f4: shr     rsi, 0x3f
0x74f8: sar     rax, 3
0x74fc: add     rsi, rax
0x74ff: sar     rsi, 1
0x7502: je      0x7518
0x7504: mov     rax, qword ptr [rip + 0x21ae5]
0x750b: test    rax, rax

```

0x750e: je	0x7518	
0x7510: jmp	rax	
rax		
	->->->	indirect JMP
0x6b0b: call	0x18d10	
0x6b10: mov	rdi, qword ptr [rsp]	
0x6b14: call	0x4e70	
0x6b19: mov	rdi, rax	
0x6b1c: mov	qword ptr [rsp + 0x58], rax	
0x6b21: call	0xd610	
0x6b26: mov	rax, qword ptr [rsp + 0x58]	
0x6b2b: mov	rcx, rbp	
0x6b2e: mov	rdx, qword ptr [rip + 0x23d73]	
0x6b35: mov	rdi, qword ptr [rip + 0x23d74]	
0x6b3c: mov	r8d, 1	
0x6b42: mov	esi, 0x28	
0x6b47: mov	qword ptr [rsp + 0x68], rax	
0x6b4c: call	0x13e30	
0x6b51: cmp	dword ptr [rsp + 0x60], -1	
0x6b56: je	0x6d88	
0x6b5c: mov	dword ptr [rsp + 0xc], 0	
0x6b64: mov	rdi, qword ptr [rsp]	
0x6b68: mov	r13d, 1	
0x6b6e: call	0x49b0	
0x6b73: cmp	qword ptr [rsp + 0x70], rax	
0x6b78: jae	0x6216	
0x6b7e: xor	edi, edi	
0x6b80: xor	r13d, r13d	
0x6b83: call	0xd1a0	
0x6b88: mov	rdi, qword ptr [rsp + 0x58]	
0x6b8d: call	0x49b0	
0x6b92: cmp	rax, qword ptr [rsp + 0x70]	
0x6b97: jbe	0x6232	
0x6b9d: jmp	0x622b	
0x622b		
0x6ba2: mov	rax, qword ptr [rip + 0x23d17]	
0x6ba9: mov	qword ptr [rip + 0x23d00], rax	
0x6bb0: mov	rax, qword ptr [rip + 0x23d01]	
0x6bb7: jmp	0x56b0	
0x56b0		
0x6bbc: cmp	dword ptr [rsp + 0xc], 0	
0x6bc1: je	0x62d5	
0x6bc7: xor	eax, eax	
0x6bc9: call	0xa5a0	
0x6bce: mov	rdi, qword ptr [rip + 0x23cdb]	
0x6bd5: mov	esi, 1	
0x6bda: call	0xf660	
0x6bdf: mov	dword ptr [rsp + 0xc], 0	

```

0x6be7: mov     rdi, qword ptr [rsp + 0x58]
0x6bec: jmp     0x62d5
0x62d5:
0x6bf1: cmp     dword ptr [rsp + 0xc], 0
0x6bf6: je      0x61dc
0x6bfc: mov     rdi, qword ptr [rip + 0x23cad]
0x6c03: mov     esi, 1
0x6c08: call    0xf660
0x6c0d: mov     dword ptr [rsp + 0xc], 0
0x6c15: mov     rdi, qword ptr [rsp + 0x58]
0x6c1a: jmp     0x61dc
0x61dc:
0x6c1f: mov     rdi, qword ptr [r12]
0x6c23: call    0x49b0
0x6c28: lea     rdi, [rax + 0x51]
0x6c2c: call    0x1d670
0x6c31: mov     r9, qword ptr [r13 + 0x20]
0x6c35: mov     r8, qword ptr [r13]
0x6c39: lea     rcx, [rip + 0x19d80]
0x6c40: mov     rbp, rax
0x6c43: or      rdx, 0xffffffffffffffff
0x6c47: mov     rdi, rax
0x6c4a: mov     esi, 1
0x6c4f: xor     eax, eax
0x6c51: call    0x4f10
0x6c56: mov     rdi, rbp
0x6c59: mov     edx, 0x1c
0x6c5e: xor     esi, esi
0x6c60: call    0x8070
0x6c65: mov     rdi, rbp
0x6c68: call    0x18d10
0x6c6d: jmp     0x5730
0x5730:
0x6c72: mov     rax, qword ptr [rsp]
0x6c76: test    rax, rax
0x6c79: je      0x6c83
0x6c7b: mov     rdi, rax
0x6c7e: call    0x18d10
0x6c83: mov     rdi, qword ptr [rsp + 0x58]
0x6c88: test    rdi, rdi
0x6c8b: je      0x5730
0x6c91: call    0x4e70
0x6c96: mov     rdx, qword ptr [rip + 0x23c0b]
0x6c9d: xor     ecx, ecx
0x6c9f: mov     rdi, qword ptr [rip + 0x23c0a]
0x6ca6: mov     r8d, 1
0x6cac: mov     esi, 0x2a
0x6cb1: mov     qword ptr [rsp], rax

```

```

0x6cb5: call    0x13e30
0x6cba: xor     ecx, ecx
0x6cbc: mov     edx, 3
0x6cc1: xor     esi, esi
0x6cc3: xor     edi, edi
0x6cc5: call    0x86c0
0x6cca: jmp     0x5730
0x5730
0x6ccf: cmp     byte ptr [r13 + 0x18], 0
0x6cd4: jne     0x6f35
0x6cda: xor     eax, eax
0x6cdc: call    0xa480
0x6ce1: cmp     byte ptr [r13 + 0x19], 0
0x6ce6: je      0x6cf2
0x6ce8: mov     dword ptr [rip + 0x23bea], 1
0x6cf2: test    r12b, 0x7f
0x6cf6: jne     0x6955
0x6cfc: and     r12d, 0xff00
0x6d03: je      0x68f1
0x6d09: xor     ebp, ebp
0x6d0b: jmp     0x565a
0x565a
0x6d10: xor     edi, edi
0x6d12: call    0x4aa0
0x6d17: mov     edx, 0x21
0x6d1c: mov     rcx, qword ptr [rip + 0x23b1d]
0x6d23: mov     esi, 1
0x6d28: lea     rdi, [rip + 0x19c49]
0x6d2f: call    0x4e40
0x6d34: mov     eax, dword ptr [rip + 0x23c62]
0x6d3a: xor     esi, esi
0x6d3c: lea     edi, [rax - 1]
0x6d3f: call    0xa4b0
0x6d44: xor     edi, edi
0x6d46: call    0xb180
0x6d4b: mov     dword ptr [rip + 0x23b87], 0
0x6d55: jmp     0x5667
0x5667
0x6d5a: mov     rdi, qword ptr [rip + 0x23b4f]
0x6d61: call    0x11100
0x6d66: mov     rdi, rax
0x6d69: xor     eax, eax
0x6d6b: call    0x89d0
0x6d70: jmp     0x571c
0x571c
0x6d75: xor     eax, eax
0x6d77: call    0xa6c0
0x6d7c: xor     eax, eax

```

```

0x6d7e: call    0xa300
0x6d83: jmp     0x56c4
0x56c4
0x6d88: mov     r13d, 1
0x6d8e: xor     eax, eax
0x6d90: call    0xa5a0
0x6d95: mov     dword ptr [rsp + 0xc], 1
0x6d9d: jmp     0x619a
0x619a
0x6da2: mov     r13d, 1
0x6da8: xor     eax, eax
0x6daa: call    0xa5a0
0x6daf: mov     dword ptr [rsp + 0xc], 1
0x6db7: jmp     0x6293
0x6293
0x6dbc: mov     rdi, rbp
0x6dbf: call    0xc4b0
0x6dc4: jmp     0x68aa
0x68aa
0x6dc9: mov     rdi, qword ptr [r13]
0x6dcd: call    0x49b0
0x6dd2: lea     rdi, [rax + 0x41]
0x6dd6: call    0x1d670
0x6ddb: mov     r9, qword ptr [r13]
0x6ddf: lea     r8, [rip + 0x19921]
0x6de6: lea     rcx, [rip + 0x19bb3]
0x6ded: mov     r12, rax
0x6df0: or      rdx, 0xffffffffffffffff
0x6df4: mov     esi, 1
0x6df9: mov     rdi, rax
0x6dfc: xor     eax, eax
0x6dfe: call    0x4f10
0x6e03: mov     rdi, r12
0x6e06: call    0xdb60
0x6e0b: xor     eax, eax
0x6e0d: call    0xa820
0x6e12: mov     rdi, r12
0x6e15: call    0x18d10
0x6e1a: jmp     0x689a
0x689a
0x6e1f: cmp     r12d, -0x3d
0x6e23: jne     0x696f
0x6e29: mov     rax, qword ptr [rsp + 0x50]
0x6e2e: cmp     byte ptr [rax], 0
0x6e31: je      0x6611
0x6e37: mov     rdi, r14
0x6e3a: call    0xc7e0
0x6e3f: mov     rdi, qword ptr [rsp + 0x50]

```

```

0x6e44: xor     eax, eax
0x6e46: lea     rsi, [rsp + 0x48]
0x6e4b: call    0x1b5b0
0x6e50: test    eax, eax
0x6e52: js      0x7287
0x6e58: mov     rdi, qword ptr [rip + 0x23a69]
0x6e5f: call    0xbb40
0x6e64: xor     edi, edi
0x6e66: call    0xd1a0
0x6e6b: mov     rdi, qword ptr [rsp + 0x48]
0x6e70: call    0xd610
0x6e75: mov     rdi, qword ptr [rsp + 0x48]
0x6e7a: call    0xc4b0
0x6e7f: mov     rdi, qword ptr [rsp + 0x48]
0x6e84: xor     esi, esi
0x6e86: call    0x193f0
0x6e8b: mov     rdi, qword ptr [rip + 0x23a36]
0x6e92: call    0xbb10
0x6e97: xor     edi, edi
0x6e99: call    0xdbb0
0x6e9e: mov     rdi, qword ptr [rip + 0x22dc3]
0x6ea5: xor     edx, edx
0x6ea7: xor     esi, esi
0x6ea9: call    0xe480
0x6eae: xor     edi, edi
0x6eb0: call    0xd1a0
0x6eb5: xor     eax, eax
0x6eb7: call    0xc8c0
0x6ebc: mov     rdi, qword ptr [rip + 0x239ed]
0x6ec3: mov     r14, rax
0x6ec6: call    0x11100
0x6ecb: mov     rdi, rax
0x6ece: call    0xc4b0
0x6ed3: xor     eax, eax
0x6ed5: call    0xa820
0x6eda: jmp     0x6611
0x6611
0x6edf: cmp     al, 0x2d
0x6ee1: jne     0x60c3
0x6ee7: cmp     byte ptr [rbp + 1], 0
0x6eeb: mov     rdx, qword ptr [rip + 0x239b6]
0x6ef2: mov     rdi, qword ptr [rip + 0x239b7]
0x6ef9: jne     0x726e
0x6eff: mov     r8d, 1
0x6f05: xor     ecx, ecx
0x6f07: mov     esi, 6
0x6f0c: call    0x13e30
0x6f11: xor     edi, edi

```

```

0x6f13: call    0xd1a0
0x6f18: jmp      0x5723
0x5723
0x6f1d: mov      edx, 0xc
0x6f22: xor       esi, esi
0x6f24: lea       rdi, [rip + 0x197e9]
0x6f2b: call     0x8070
0x6f30: jmp      0x6cf2
0x6cf2
0x6f35: mov      rdi, qword ptr [rip + 0x2398c]
0x6f3c: call     0xbb10
0x6f41: jmp      0x6cda
0x6cda
0x6f46: mov      rdi, r12
0x6f49: call     0x49b0
0x6f4e: mov      rdi, r12
0x6f51: mov      rsi, rax
0x6f54: call     0x18530
0x6f59: test     eax, eax
0x6f5b: je       0x70a8
0x6f61: mov      r8, r12
0x6f64: lea       rcx, [rip + 0x197d6]
0x6f6b: jmp      0x5f42
0x5f42
0x6f70: mov      rdi, r14
0x6f73: call     0x1d7a0
0x6f78: mov      r12, rax
0x6f7b: jmp      0x5593
0x5593
0x6f80: xor       esi, esi
0x6f82: lea       rdi, [rip + 0x19717]
0x6f89: call     0x17e50
0x6f8e: mov      dword ptr [rsp], eax
0x6f91: jmp      0x53b4
0x53b4
0x6f96: mov      rdi, qword ptr [rbx + rax*8]
0x6f9a: lea       edx, [rax + 1]
0x6f9d: mov      dword ptr [rip + 0x238a5], edx
0x6fa3: call     0x1d7a0
0x6fa8: mov      r13, rax
0x6fab: movsxd   rax, dword ptr [rip + 0x23896]
0x6fb2: cmp      eax, ebp
0x6fb4: jge      0x5124
0x6fba: mov      rdi, qword ptr [rbx + rax*8]
0x6fbe: lea       edx, [rax + 1]
0x6fc1: mov      dword ptr [rip + 0x23881], edx
0x6fc7: call     0x1d7a0
0x6fcc: cmp      dword ptr [rip + 0x23876], ebp

```



```

0x6fd2: mov     r14, rax
0x6fd5: jge     0x5127
0x6fdb: mov     rcx, qword ptr [rip + 0x23906]
0x6fe2: mov     rdi, qword ptr [rip + 0x23897]
0x6fe9: lea     rdx, [rip + 0x198f8]
0x6ff0: xor     eax, eax
0x6ff2: mov     esi, 1
0x6ff7: call    0x4e60
0x6ffc: jmp     0x5127
0x5127:
0x7001: mov     rdi, r13
0x7004: call    0x1d7a0
0x7009: mov     r12, rax
0x700c: jmp     0x5532
0x5532:
0x7011: lea     rbx, [rip + 0x19543]
0x7018: jmp     0x529c
0x529c:
0x701d: mov     rdi, qword ptr [rip + 0x238ac]
0x7024: xor     edx, edx
0x7026: lea     rsi, [rip + 0x19505]
0x702d: call    0x8070
0x7032: cmp     al, 0x79
0x7034: jne     0x5730
0x703a: jmp     0x6442
0x6442:
0x703f: mov     edi, 1
0x7044: xor     eax, eax
0x7046: call    0x7bd0
0x704b: jmp     0x59ce
0x59ce:
0x7050: mov     esi, 1
0x7055: lea     rdi, [rip + 0x19664]
0x705c: call    0x17e50
0x7061: jmp     0x53df
0x53df:
0x7066: mov     rcx, qword ptr [rip + 0x2387b]
0x706d: mov     rdi, qword ptr [rip + 0x2380c]
0x7074: xor     eax, eax
0x7076: mov     r8d, 0x800
0x707c: lea     rdx, [rip + 0x19895]
0x7083: mov     esi, 1
0x7088: call    0x4e60
0x708d: jmp     0x54a3
0x54a3:
0x7092: xor     esi, esi
0x7094: lea     rdi, [rip + 0x195fa]
0x709b: call    0x17e50

```

```

0x70a0: mov     r12d, eax
0x70a3: jmp     0x53a3
0x53a3
0x70a8: mov     r8, r12
0x70ab: lea     rcx, [rip + 0x19696]
0x70b2: jmp     0x5f42
0x5f42
0x70b7: mov     rdi, qword ptr [r13 + 0x10]
0x70bb: test    rdi, rdi
0x70be: je      0x6955
0x70c4: xor     eax, eax
0x70c6: call    0x19940
0x70cb: mov     r8d, 1
0x70d1: mov     esi, 0x21
0x70d6: mov     rdx, qword ptr [rip + 0x237cb]
0x70dd: mov     rdi, qword ptr [rip + 0x237cc]
0x70e4: mov     rcx, rax
0x70e7: mov     rbp, rax
0x70ea: call    0x13e30
0x70ef: mov     rdi, qword ptr [rip + 0x237ba]
0x70f6: call    0x11100
0x70fb: mov     rdi, rax
0x70fe: xor     eax, eax
0x7100: call    0x89d0
0x7105: mov     rdi, rbp
0x7108: xor     ebp, ebp
0x710a: call    0x18d10
0x710f: jmp     0x565a
0x565a
0x7114: mov     qword ptr [rsp + 0x40], 0
0x711d: lea     r12, [rsp + 0x40]
0x7122: lea     rbp, [rip + 0x1966f]
0x7129: jmp     0x714f
0x714f
0x712b: lea     rdi, [rip + 0x22ade]
0x7132: call    0xdb60
0x7137: xor     eax, eax
0x7139: call    0xa5a0
0x713e: xor     eax, eax
0x7140: call    0xa820
0x7145: mov     edi, 2
0x714a: call    0x4ea0
0x714f: mov     rsi, r12
0x7152: xor     ecx, ecx
0x7154: xor     edx, edx
0x7156: mov     rdi, rbp
0x7159: call    0x83e0
0x715e: xor     eax, eax

```

```

0x7160: call    0xa820
0x7165: mov     rsi, qword ptr [rsp + 0x40]
0x716a: test    rsi, rsi
0x716d: je       0x712b
0x716f: mov     rdi, qword ptr [rsp + 0x38]
0x7174: call    0x4b30
0x7179: test    eax, eax
0x717b: jne     0x712b
0x717d: xor     eax, eax
0x717f: call    0xa820
0x7184: mov     rdi, qword ptr [rsp + 0x38]
0x7189: call    0x18d10
0x718e: mov     rdi, qword ptr [rsp + 0x40]
0x7193: call    0x18d10
0x7198: jmp     0x5a59
0x5a59
0x719d: xor     eax, eax
0x719f: xor     ebp, ebp
0x71a1: call    0xa5a0
0x71a6: mov     rdi, qword ptr [r13]
0x71aa: call    0x195e0
0x71af: jmp     0x565a
0x565a
0x71b4: mov     rdi, qword ptr [rip + 0x236f5]
0x71bb: xor     ebp, ebp
0x71bd: call    0x110b0
0x71c2: jmp     0x565a
0x565a
0x71c7: cmp     byte ptr [rbp + 1], 0
0x71cb: mov     rdx, qword ptr [rip + 0x236d6]
0x71d2: mov     rdi, qword ptr [rip + 0x236d7]
0x71d9: jne     0x72a7
0x71df: mov     r8d, 1
0x71e5: xor     ecx, ecx
0x71e7: mov     esi, 5
0x71ec: call    0x13e30
0x71f1: jmp     0x6f11
0x6f11
0x71f6: mov     rcx, qword ptr [rip + 0x236cb]
0x71fd: mov     rdx, qword ptr [rip + 0x236a4]
0x7204: mov     esi, 1
0x7209: mov     r8d, 1
0x720f: mov     rdi, qword ptr [rip + 0x2369a]
0x7216: call    0x13e30
0x721b: mov     rdi, qword ptr [rip + 0x2368e]
0x7222: mov     ebp, eax
0x7224: call    0x11100
0x7229: mov     rdi, rax

```

```

0x722c: call    0xc4b0
0x7231: xor     edi, edi
0x7233: call    0xd1a0
0x7238: xor     eax, eax
0x723a: call    0x9640
0x723f: xor     eax, eax
0x7241: call    0xc8c0
0x7246: mov     r14, rax
0x7249: test    ebp, ebp
0x724b: je      0x5730
0x7251: xor     ebp, ebp
0x7253: jmp     0x565a
0x565a
0x7258: mov     rdi, qword ptr [rsp + 0x48]
0x725d: mov     edx, 0x1c
0x7262: xor     esi, esi
0x7264: call    0x8070
0x7269: jmp     0x5723
0x5723
0x726e: lea     rcx, [rbp + 1]
0x7272: mov     r8d, 1
0x7278: mov     esi, 0x25
0x727d: call    0x13e30
0x7282: jmp     0x6f11
0x6f11
0x7287: mov     rdi, qword ptr [rsp + 0x48]
0x728c: mov     edx, 0x1c
0x7291: xor     esi, esi
0x7293: call    0x8070
0x7298: xor     eax, eax
0x729a: call    0xc8c0
0x729f: mov     r14, rax
0x72a2: jmp     0x6611
0x6611
0x72a7: lea     rcx, [rbp + 1]
0x72ab: mov     r8d, 1
0x72b1: mov     esi, 0x24
0x72b6: call    0x13e30
0x72bb: jmp     0x6f11
0x6f11
0x72c0: mov     rdi, qword ptr [rsp + 0x48]
0x72c5: call    0x185d0
0x72ca: mov     ebp, eax
0x72cc: test    eax, eax
0x72ce: je      0x73fa
0x72d4: xor     edi, edi
0x72d6: call    0xd1a0
0x72db: mov     rdi, qword ptr [rsp + 0x48]

```

```

0x72e0: call    0xd610
0x72e5: mov     rdi, qword ptr [rsp + 0x48]
0x72ea: mov     esi, ebp
0x72ec: call    0x193f0
0x72f1: xor     edi, edi
0x72f3: call    0xdbb0
0x72f8: jmp     0x6f11
0x6f11
0x72fd: mov     rdx, qword ptr [rip + 0x235a4]
0x7304: mov     rdi, qword ptr [rip + 0x235a5]
0x730b: xor     ecx, ecx
0x730d: mov     esi, 7
0x7312: mov     r8d, 1
0x7318: call    0x13e30
0x731d: xor     edi, edi
0x731f: call    0xd1a0
0x7324: jmp     0x5723
0x5723
0x7329: mov     rdi, qword ptr [rip + 0x23580]
0x7330: mov     r15d, dword ptr [rsp + 0x10]
0x7335: call    0xf480
0x733a: mov     rdi, qword ptr [rip + 0x23567]
0x7341: call    0xf480
0x7346: xor     eax, eax
0x7348: call    0xa480
0x734d: xor     eax, eax
0x734f: call    0xe4e0
0x7354: xor     eax, eax
0x7356: call    0xa820
0x735b: xor     eax, eax
0x735d: call    0x18330
0x7362: mov     r8, rax
0x7365: mov     eax, 0x3c
0x736a: sub     eax, dword ptr [r8]
0x736d: mov     edi, eax
0x736f: call    0x4aa0
0x7374: mov     dword ptr [rip + 0x23566], 0
0x737e: jmp     0x6447
0x6447
0x7383: mov     rdi, r14
0x7386: xor     ebp, ebp
0x7388: call    0xc7e0
0x738d: xor     eax, eax
0x738f: call    0xc8c0
0x7394: mov     rdi, qword ptr [rip + 0x23515]
0x739b: mov     r14, rax
0x739e: call    0xf480
0x73a3: mov     rdi, qword ptr [rip + 0x234fe]

```

```

0x73aa: call    0xf480
0x73af: xor     eax, eax
0x73b1: call    0xa480
0x73b6: xor     eax, eax
0x73b8: call    0xe4e0
0x73bd: xor     eax, eax
0x73bf: call    0xa820
0x73c4: xor     eax, eax
0x73c6: call    0x18330
0x73cb: mov     r8, rax
0x73ce: mov     eax, 0x3c
0x73d3: sub     eax, dword ptr [r8]
0x73d6: mov     edi, eax
0x73d8: call    0x4aa0
0x73dd: mov     dword ptr [rip + 0x234fd], 0
0x73e7: jmp     0x565a
0x565a
0x73ec: lea     rdi, [rip + 0x1955d]
0x73f3: xor     eax, eax
0x73f5: call    0x8b00
0x73fa: mov     rdi, qword ptr [rsp + 0x48]
0x73ff: xor     ebp, ebp
0x7401: call    0xc4b0
0x7406: xor     edi, edi
0x7408: call    0xd1a0
0x740d: mov     rdi, qword ptr [rsp + 0x48]
0x7412: call    0xd610
0x7417: mov     rdi, qword ptr [rsp + 0x48]
0x741c: xor     esi, esi
0x741e: call    0x193f0
0x7423: xor     edi, edi
0x7425: call    0xdbb0
0x742a: xor     edi, edi
0x742c: call    0xd1a0
0x7431: mov     rdi, qword ptr [rip + 0x23478]
0x7438: call    0xf480
0x743d: mov     rdi, qword ptr [rip + 0x23464]
0x7444: call    0xf480
0x7449: xor     eax, eax
0x744b: call    0xa480
0x7450: mov     rdi, qword ptr [rip + 0x23471]
0x7457: call    0xbb10
0x745c: xor     eax, eax
0x745e: mov     dword ptr [rip + 0x23474], 1
0x7468: call    0xc8c0
0x746d: mov     r14, rax
0x7470: jmp     0x565a
0x565a

```

```

0x7475: call    0x49d0
0x747a: nop      word ptr [rax + rax]
0x7480: endbr64
0x7484: xor       ebp, ebp
0x7486: mov       r9, rdx
0x7489: pop       rsi
0x748a: mov       rdx, rsp
0x748d: and       rsp, 0xfffffffffffffff0
0x7491: push      rax
0x7492: push      rsp
0x7493: xor       r8d, r8d
0x7496: xor       ecx, ecx
0x7498: lea       rdi, [rip - 0x256f]
0x749f: call      qword ptr [rip + 0x21b23]
0x74a5: hlt
0x74a6: nop      word ptr cs:[rax + rax]
0x74b0: lea       rdi, [rip + 0x23379]
0x74b7: lea       rax, [rip + 0x23372]
0x74be: cmp       rax, rdi
0x74c1: je        0x74d8
0x74c3: mov       rax, qword ptr [rip + 0x21b06]
0x74ca: test      rax, rax
0x74cd: je        0x74d8
0x74cf: jmp       rax
rax
->->->                                indirect JMP

0x6a74: call      0x18d10
0x6a79: mov       rdi, qword ptr [rsp]
0x6a7d: call      0x4e70
0x6a82: mov       rdi, rax
0x6a85: mov       qword ptr [rsp + 0x58], rax
0x6a8a: call      0xd610
0x6a8f: mov       rax, qword ptr [rsp + 0x58]
0x6a94: mov       rcx, rbp
0x6a97: mov       rdx, qword ptr [rip + 0x23e0a]
0x6a9e: mov       rdi, qword ptr [rip + 0x23e0b]
0x6aa5: mov       r8d, 1
0x6aab: mov       esi, 0x29
0x6ab0: mov       qword ptr [rsp + 0x68], rax
0x6ab5: call      0x13e30
0x6aba: cmp       dword ptr [rsp + 0x60], -1
0x6abf: je        0x6da2
0x6ac5: mov       dword ptr [rsp + 0xc], 0
0x6acd: mov       rdi, qword ptr [rsp]
0x6ad1: mov       r13d, 1
0x6ad7: call      0x49b0
0x6adc: cmp       qword ptr [rsp + 0x70], rax
0x6ae1: jae       0x630f

```

```

0x6ae7: xor     edi, edi
0x6ae9: xor     r13d, r13d
0x6aec: call    0xd1a0
0x6af1: mov     rdi, qword ptr [rsp + 0x58]
0x6af6: call    0x49b0
0x6afb: cmp     rax, qword ptr [rsp + 0x70]
0x6b00: jbe     0x632b
0x6b06: jmp     0x6324
0x6324:
0x6b0b: call    0x18d10
0x6b10: mov     rdi, qword ptr [rsp]
0x6b14: call    0x4e70
0x6b19: mov     rdi, rax
0x6b1c: mov     qword ptr [rsp + 0x58], rax
0x6b21: call    0xd610
0x6b26: mov     rax, qword ptr [rsp + 0x58]
0x6b2b: mov     rcx, rbp
0x6b2e: mov     rdx, qword ptr [rip + 0x23d73]
0x6b35: mov     rdi, qword ptr [rip + 0x23d74]
0x6b3c: mov     r8d, 1
0x6b42: mov     esi, 0x28
0x6b47: mov     qword ptr [rsp + 0x68], rax
0x6b4c: call    0x13e30
0x6b51: cmp     dword ptr [rsp + 0x60], -1
0x6b56: je      0x6d88
0x6b5c: mov     dword ptr [rsp + 0xc], 0
0x6b64: mov     rdi, qword ptr [rsp]
0x6b68: mov     r13d, 1
0x6b6e: call    0x49b0
0x6b73: cmp     qword ptr [rsp + 0x70], rax
0x6b78: jae     0x6216
0x6b7e: xor     edi, edi
0x6b80: xor     r13d, r13d
0x6b83: call    0xd1a0
0x6b88: mov     rdi, qword ptr [rsp + 0x58]
0x6b8d: call    0x49b0
0x6b92: cmp     rax, qword ptr [rsp + 0x70]
0x6b97: jbe     0x6232
0x6b9d: jmp     0x622b
0x622b:
0x6ba2: mov     rax, qword ptr [rip + 0x23d17]
0x6ba9: mov     qword ptr [rip + 0x23d00], rax
0x6bb0: mov     rax, qword ptr [rip + 0x23d01]
0x6bb7: jmp     0x56b0
0x56b0:
0x6bbc: cmp     dword ptr [rsp + 0xc], 0
0x6bc1: je      0x62d5
0x6bc7: xor     eax, eax

```



```

0x6bc9: call    0xa5a0
0x6bce: mov     rdi, qword ptr [rip + 0x23cdb]
0x6bd5: mov     esi, 1
0x6bda: call    0xf660
0x6bdf: mov     dword ptr [rsp + 0xc], 0
0x6be7: mov     rdi, qword ptr [rsp + 0x58]
0x6bec: jmp     0x62d5
0x62d5
0x6bf1: cmp     dword ptr [rsp + 0xc], 0
0x6bf6: je      0x61dc
0x6bfc: mov     rdi, qword ptr [rip + 0x23cad]
0x6c03: mov     esi, 1
0x6c08: call    0xf660
0x6c0d: mov     dword ptr [rsp + 0xc], 0
0x6c15: mov     rdi, qword ptr [rsp + 0x58]
0x6c1a: jmp     0x61dc
0x61dc
0x6c1f: mov     rdi, qword ptr [r12]
0x6c23: call    0x49b0
0x6c28: lea     rdi, [rax + 0x51]
0x6c2c: call    0x1d670
0x6c31: mov     r9, qword ptr [r13 + 0x20]
0x6c35: mov     r8, qword ptr [r13]
0x6c39: lea     rcx, [rip + 0x19d80]
0x6c40: mov     rbp, rax
0x6c43: or      rdx, 0xffffffffffffffff
0x6c47: mov     rdi, rax
0x6c4a: mov     esi, 1
0x6c4f: xor     eax, eax
0x6c51: call    0x4f10
0x6c56: mov     rdi, rbp
0x6c59: mov     edx, 0x1c
0x6c5e: xor     esi, esi
0x6c60: call    0x8070
0x6c65: mov     rdi, rbp
0x6c68: call    0x18d10
0x6c6d: jmp     0x5730
0x5730
0x6c72: mov     rax, qword ptr [rsp]
0x6c76: test    rax, rax
0x6c79: je      0x6c83
0x6c7b: mov     rdi, rax
0x6c7e: call    0x18d10
0x6c83: mov     rdi, qword ptr [rsp + 0x58]
0x6c88: test    rdi, rdi
0x6c8b: je      0x5730
0x6c91: call    0x4e70
0x6c96: mov     rdx, qword ptr [rip + 0x23c0b]

```

```

0x6c9d: xor      ecx, ecx
0x6c9f: mov      rdi, qword ptr [rip + 0x23c0a]
0x6ca6: mov      r8d, 1
0x6cac: mov      esi, 0x2a
0x6cb1: mov      qword ptr [rsp], rax
0x6cb5: call     0x13e30
0x6cba: xor      ecx, ecx
0x6cbc: mov      edx, 3
0x6cc1: xor      esi, esi
0x6cc3: xor      edi, edi
0x6cc5: call     0x86c0
0x6cca: jmp      0x5730
0x5730
0x6ccf: cmp      byte ptr [r13 + 0x18], 0
0x6cd4: jne      0x6f35
0x6cda: xor      eax, eax
0x6cdc: call     0xa480
0x6ce1: cmp      byte ptr [r13 + 0x19], 0
0x6ce6: je       0x6cf2
0x6ce8: mov      dword ptr [rip + 0x23bea], 1
0x6cf2: test     r12b, 0x7f
0x6cf6: jne      0x6955
0x6cfc: and      r12d, 0xff00
0x6d03: je       0x68f1
0x6d09: xor      ebp, ebp
0x6d0b: jmp      0x565a
0x565a
0x6d10: xor      edi, edi
0x6d12: call     0x4aa0
0x6d17: mov      edx, 0x21
0x6d1c: mov      rcx, qword ptr [rip + 0x23b1d]
0x6d23: mov      esi, 1
0x6d28: lea      rdi, [rip + 0x19c49]
0x6d2f: call     0x4e40
0x6d34: mov      eax, dword ptr [rip + 0x23c62]
0x6d3a: xor      esi, esi
0x6d3c: lea      edi, [rax - 1]
0x6d3f: call     0xa4b0
0x6d44: xor      edi, edi
0x6d46: call     0xb180
0x6d4b: mov      dword ptr [rip + 0x23b87], 0
0x6d55: jmp      0x5667
0x5667
0x6d5a: mov      rdi, qword ptr [rip + 0x23b4f]
0x6d61: call     0x11100
0x6d66: mov      rdi, rax
0x6d69: xor      eax, eax
0x6d6b: call     0x89d0

```

```

0x6d70: jmp      0x571c
0x571c
0x6d75: xor       eax, eax
0x6d77: call      0xa6c0
0x6d7c: xor       eax, eax
0x6d7e: call      0xa300
0x6d83: jmp      0x56c4
0x56c4
0x6d88: mov       r13d, 1
0x6d8e: xor       eax, eax
0x6d90: call      0xa5a0
0x6d95: mov       dword ptr [rsp + 0xc], 1
0x6d9d: jmp      0x619a
0x619a
0x6da2: mov       r13d, 1
0x6da8: xor       eax, eax
0x6daa: call      0xa5a0
0x6daf: mov       dword ptr [rsp + 0xc], 1
0x6db7: jmp      0x6293
0x6293
0x6dbc: mov       rdi, rbp
0x6dbf: call      0xc4b0
0x6dc4: jmp      0x68aa
0x68aa
0x6dc9: mov       rdi, qword ptr [r13]
0x6dcd: call      0x49b0
0x6dd2: lea       rdi, [rax + 0x41]
0x6dd6: call      0x1d670
0x6ddb: mov       r9, qword ptr [r13]
0x6ddf: lea       r8, [rip + 0x19921]
0x6de6: lea       rcx, [rip + 0x19bb3]
0x6ded: mov       r12, rax
0x6df0: or        rdx, 0xffffffffffffffff
0x6df4: mov       esi, 1
0x6df9: mov       rdi, rax
0x6dfc: xor       eax, eax
0x6dfe: call      0x4f10
0x6e03: mov       rdi, r12
0x6e06: call      0xdb60
0x6e0b: xor       eax, eax
0x6e0d: call      0xa820
0x6e12: mov       rdi, r12
0x6e15: call      0x18d10
0x6e1a: jmp      0x689a
0x689a
0x6e1f: cmp       r12d, -0x3d
0x6e23: jne       0x696f
0x6e29: mov       rax, qword ptr [rsp + 0x50]

```

```

0x6e2e: cmp     byte ptr [rax], 0
0x6e31: je        0x6611
0x6e37: mov     rdi, r14
0x6e3a: call    0xc7e0
0x6e3f: mov     rdi, qword ptr [rsp + 0x50]
0x6e44: xor     eax, eax
0x6e46: lea     rsi, [rsp + 0x48]
0x6e4b: call    0x1b5b0
0x6e50: test    eax, eax
0x6e52: js      0x7287
0x6e58: mov     rdi, qword ptr [rip + 0x23a69]
0x6e5f: call    0xbb40
0x6e64: xor     edi, edi
0x6e66: call    0xd1a0
0x6e6b: mov     rdi, qword ptr [rsp + 0x48]
0x6e70: call    0xd610
0x6e75: mov     rdi, qword ptr [rsp + 0x48]
0x6e7a: call    0xc4b0
0x6e7f: mov     rdi, qword ptr [rsp + 0x48]
0x6e84: xor     esi, esi
0x6e86: call    0x193f0
0x6e8b: mov     rdi, qword ptr [rip + 0x23a36]
0x6e92: call    0xbb10
0x6e97: xor     edi, edi
0x6e99: call    0xdbb0
0x6e9e: mov     rdi, qword ptr [rip + 0x22dc3]
0x6ea5: xor     edx, edx
0x6ea7: xor     esi, esi
0x6ea9: call    0xe480
0x6eae: xor     edi, edi
0x6eb0: call    0xd1a0
0x6eb5: xor     eax, eax
0x6eb7: call    0xc8c0
0x6ebc: mov     rdi, qword ptr [rip + 0x239ed]
0x6ec3: mov     r14, rax
0x6ec6: call    0x11100
0x6ecb: mov     rdi, rax
0x6ece: call    0xc4b0
0x6ed3: xor     eax, eax
0x6ed5: call    0xa820
0x6eda: jmp     0x6611
0x6611:
0x6edf: cmp     al, 0x2d
0x6ee1: jne     0x60c3
0x6ee7: cmp     byte ptr [rbp + 1], 0
0x6eeb: mov     rdx, qword ptr [rip + 0x239b6]
0x6ef2: mov     rdi, qword ptr [rip + 0x239b7]
0x6ef9: jne     0x726e

```

```

0x6eff: mov     r8d, 1
0x6f05: xor     ecx, ecx
0x6f07: mov     esi, 6
0x6f0c: call    0x13e30
0x6f11: xor     edi, edi
0x6f13: call    0xd1a0
0x6f18: jmp     0x5723
0x5723
0x6f1d: mov     edx, 0xc
0x6f22: xor     esi, esi
0x6f24: lea     rdi, [rip + 0x197e9]
0x6f2b: call    0x8070
0x6f30: jmp     0x6cf2
0x6cf2
0x6f35: mov     rdi, qword ptr [rip + 0x2398c]
0x6f3c: call    0xbb10
0x6f41: jmp     0x6cda
0x6cda
0x6f46: mov     rdi, r12
0x6f49: call    0x49b0
0x6f4e: mov     rdi, r12
0x6f51: mov     rsi, rax
0x6f54: call    0x18530
0x6f59: test    eax, eax
0x6f5b: je      0x70a8
0x6f61: mov     r8, r12
0x6f64: lea     rcx, [rip + 0x197d6]
0x6f6b: jmp     0x5f42
0x5f42
0x6f70: mov     rdi, r14
0x6f73: call    0x1d7a0
0x6f78: mov     r12, rax
0x6f7b: jmp     0x5593
0x5593
0x6f80: xor     esi, esi
0x6f82: lea     rdi, [rip + 0x19717]
0x6f89: call    0x17e50
0x6f8e: mov     dword ptr [rsp], eax
0x6f91: jmp     0x53b4
0x53b4
0x6f96: mov     rdi, qword ptr [rbx + rax*8]
0x6f9a: lea     edx, [rax + 1]
0x6f9d: mov     dword ptr [rip + 0x238a5], edx
0x6fa3: call    0x1d7a0
0x6fa8: mov     r13, rax
0x6fab: movsxd  rax, dword ptr [rip + 0x23896]
0x6fb2: cmp     eax, ebp
0x6fb4: jge     0x5124

```

```

0x6fba: mov     rdi, qword ptr [rbx + rax*8]
0x6fbe: lea     edx, [rax + 1]
0x6fc1: mov     dword ptr [rip + 0x23881], edx
0x6fc7: call    0x1d7a0
0x6fcc: cmp     dword ptr [rip + 0x23876], ebp
0x6fd2: mov     r14, rax
0x6fd5: jge     0x5127
0x6fdb: mov     rcx, qword ptr [rip + 0x23906]
0x6fe2: mov     rdi, qword ptr [rip + 0x23897]
0x6fe9: lea     rdx, [rip + 0x198f8]
0x6ff0: xor     eax, eax
0x6ff2: mov     esi, 1
0x6ff7: call    0x4e60
0x6ffc: jmp     0x5127
0x5127:
0x7001: mov     rdi, r13
0x7004: call    0x1d7a0
0x7009: mov     r12, rax
0x700c: jmp     0x5532
0x5532:
0x7011: lea     rbx, [rip + 0x19543]
0x7018: jmp     0x529c
0x529c:
0x701d: mov     rdi, qword ptr [rip + 0x238ac]
0x7024: xor     edx, edx
0x7026: lea     rsi, [rip + 0x19505]
0x702d: call    0x8070
0x7032: cmp     al, 0x79
0x7034: jne     0x5730
0x703a: jmp     0x6442
0x6442:
0x703f: mov     edi, 1
0x7044: xor     eax, eax
0x7046: call    0x7bd0
0x704b: jmp     0x59ce
0x59ce:
0x7050: mov     esi, 1
0x7055: lea     rdi, [rip + 0x19664]
0x705c: call    0x17e50
0x7061: jmp     0x53df
0x53df:
0x7066: mov     rcx, qword ptr [rip + 0x2387b]
0x706d: mov     rdi, qword ptr [rip + 0x2380c]
0x7074: xor     eax, eax
0x7076: mov     r8d, 0x800
0x707c: lea     rdx, [rip + 0x19895]
0x7083: mov     esi, 1
0x7088: call    0x4e60

```

```

0x708d: jmp      0x54a3
0x54a3
0x7092: xor      esi, esi
0x7094: lea      rdi, [rip + 0x195fa]
0x709b: call     0x17e50
0x70a0: mov      r12d, eax
0x70a3: jmp      0x53a3
0x53a3
0x70a8: mov      r8, r12
0x70ab: lea      rcx, [rip + 0x19696]
0x70b2: jmp      0x5f42
0x5f42
0x70b7: mov      rdi, qword ptr [r13 + 0x10]
0x70bb: test     rdi, rdi
0x70be: je       0x6955
0x70c4: xor      eax, eax
0x70c6: call     0x19940
0x70cb: mov      r8d, 1
0x70d1: mov      esi, 0x21
0x70d6: mov      rdx, qword ptr [rip + 0x237cb]
0x70dd: mov      rdi, qword ptr [rip + 0x237cc]
0x70e4: mov      rcx, rax
0x70e7: mov      rbp, rax
0x70ea: call     0x13e30
0x70ef: mov      rdi, qword ptr [rip + 0x237ba]
0x70f6: call     0x11100
0x70fb: mov      rdi, rax
0x70fe: xor      eax, eax
0x7100: call     0x89d0
0x7105: mov      rdi, rbp
0x7108: xor      ebp, ebp
0x710a: call     0x18d10
0x710f: jmp      0x565a
0x565a
0x7114: mov      qword ptr [rsp + 0x40], 0
0x711d: lea      r12, [rsp + 0x40]
0x7122: lea      rbp, [rip + 0x1966f]
0x7129: jmp      0x714f
0x714f
0x712b: lea      rdi, [rip + 0x22ade]
0x7132: call     0xdb60
0x7137: xor      eax, eax
0x7139: call     0xa5a0
0x713e: xor      eax, eax
0x7140: call     0xa820
0x7145: mov      edi, 2
0x714a: call     0x4ea0
0x714f: mov      rsi, r12

```

```

0x7152: xor     ecx, ecx
0x7154: xor     edx, edx
0x7156: mov     rdi, rbp
0x7159: call    0x83e0
0x715e: xor     eax, eax
0x7160: call    0xa820
0x7165: mov     rsi, qword ptr [rsp + 0x40]
0x716a: test    rsi, rsi
0x716d: je      0x712b
0x716f: mov     rdi, qword ptr [rsp + 0x38]
0x7174: call    0x4b30
0x7179: test    eax, eax
0x717b: jne     0x712b
0x717d: xor     eax, eax
0x717f: call    0xa820
0x7184: mov     rdi, qword ptr [rsp + 0x38]
0x7189: call    0x18d10
0x718e: mov     rdi, qword ptr [rsp + 0x40]
0x7193: call    0x18d10
0x7198: jmp     0x5a59
0x5a59
0x719d: xor     eax, eax
0x719f: xor     ebp, ebp
0x71a1: call    0xa5a0
0x71a6: mov     rdi, qword ptr [r13]
0x71aa: call    0x195e0
0x71af: jmp     0x565a
0x565a
0x71b4: mov     rdi, qword ptr [rip + 0x236f5]
0x71bb: xor     ebp, ebp
0x71bd: call    0x110b0
0x71c2: jmp     0x565a
0x565a
0x71c7: cmp     byte ptr [rbp + 1], 0
0x71cb: mov     rdx, qword ptr [rip + 0x236d6]
0x71d2: mov     rdi, qword ptr [rip + 0x236d7]
0x71d9: jne     0x72a7
0x71df: mov     r8d, 1
0x71e5: xor     ecx, ecx
0x71e7: mov     esi, 5
0x71ec: call    0x13e30
0x71f1: jmp     0x6f11
0x6f11
0x71f6: mov     rcx, qword ptr [rip + 0x236cb]
0x71fd: mov     rdx, qword ptr [rip + 0x236a4]
0x7204: mov     esi, 1
0x7209: mov     r8d, 1
0x720f: mov     rdi, qword ptr [rip + 0x2369a]

```



```

0x7216: call    0x13e30
0x721b: mov     rdi, qword ptr [rip + 0x2368e]
0x7222: mov     ebp, eax
0x7224: call    0x11100
0x7229: mov     rdi, rax
0x722c: call    0xc4b0
0x7231: xor     edi, edi
0x7233: call    0xd1a0
0x7238: xor     eax, eax
0x723a: call    0x9640
0x723f: xor     eax, eax
0x7241: call    0xc8c0
0x7246: mov     r14, rax
0x7249: test    ebp, ebp
0x724b: je      0x5730
0x7251: xor     ebp, ebp
0x7253: jmp     0x565a
0x565a
0x7258: mov     rdi, qword ptr [rsp + 0x48]
0x725d: mov     edx, 0x1c
0x7262: xor     esi, esi
0x7264: call    0x8070
0x7269: jmp     0x5723
0x5723
0x726e: lea     rcx, [rbp + 1]
0x7272: mov     r8d, 1
0x7278: mov     esi, 0x25
0x727d: call    0x13e30
0x7282: jmp     0x6f11
0x6f11
0x7287: mov     rdi, qword ptr [rsp + 0x48]
0x728c: mov     edx, 0x1c
0x7291: xor     esi, esi
0x7293: call    0x8070
0x7298: xor     eax, eax
0x729a: call    0xc8c0
0x729f: mov     r14, rax
0x72a2: jmp     0x6611
0x6611
0x72a7: lea     rcx, [rbp + 1]
0x72ab: mov     r8d, 1
0x72b1: mov     esi, 0x24
0x72b6: call    0x13e30
0x72bb: jmp     0x6f11
0x6f11
0x72c0: mov     rdi, qword ptr [rsp + 0x48]
0x72c5: call    0x185d0
0x72ca: mov     ebp, eax

```

```

0x72cc: test    eax, eax
0x72ce: je        0x73fa
0x72d4: xor      edi, edi
0x72d6: call     0xd1a0
0x72db: mov      rdi, qword ptr [rsp + 0x48]
0x72e0: call     0xd610
0x72e5: mov      rdi, qword ptr [rsp + 0x48]
0x72ea: mov      esi, ebp
0x72ec: call     0x193f0
0x72f1: xor      edi, edi
0x72f3: call     0xdbb0
0x72f8: jmp      0x6f11
0x6f11
0x72fd: mov      rdx, qword ptr [rip + 0x235a4]
0x7304: mov      rdi, qword ptr [rip + 0x235a5]
0x730b: xor      ecx, ecx
0x730d: mov      esi, 7
0x7312: mov      r8d, 1
0x7318: call     0x13e30
0x731d: xor      edi, edi
0x731f: call     0xd1a0
0x7324: jmp      0x5723
0x5723
0x7329: mov      rdi, qword ptr [rip + 0x23580]
0x7330: mov      r15d, dword ptr [rsp + 0x10]
0x7335: call     0xf480
0x733a: mov      rdi, qword ptr [rip + 0x23567]
0x7341: call     0xf480
0x7346: xor      eax, eax
0x7348: call     0xa480
0x734d: xor      eax, eax
0x734f: call     0xe4e0
0x7354: xor      eax, eax
0x7356: call     0xa820
0x735b: xor      eax, eax
0x735d: call     0x18330
0x7362: mov      r8, rax
0x7365: mov      eax, 0x3c
0x736a: sub      eax, dword ptr [r8]
0x736d: mov      edi, eax
0x736f: call     0x4aa0
0x7374: mov      dword ptr [rip + 0x23566], 0
0x737e: jmp      0x6447
0x6447
0x7383: mov      rdi, r14
0x7386: xor      ebp, ebp
0x7388: call     0xc7e0
0x738d: xor      eax, eax

```

```

0x738f: call    0xc8c0
0x7394: mov     rdi, qword ptr [rip + 0x23515]
0x739b: mov     r14, rax
0x739e: call    0xf480
0x73a3: mov     rdi, qword ptr [rip + 0x234fe]
0x73aa: call    0xf480
0x73af: xor     eax, eax
0x73b1: call    0xa480
0x73b6: xor     eax, eax
0x73b8: call    0xe4e0
0x73bd: xor     eax, eax
0x73bf: call    0xa820
0x73c4: xor     eax, eax
0x73c6: call    0x18330
0x73cb: mov     r8, rax
0x73ce: mov     eax, 0x3c
0x73d3: sub     eax, dword ptr [r8]
0x73d6: mov     edi, eax
0x73d8: call    0x4aa0
0x73dd: mov     dword ptr [rip + 0x234fd], 0
0x73e7: jmp     0x565a
0x565a
0x73ec: lea     rdi, [rip + 0x1955d]
0x73f3: xor     eax, eax
0x73f5: call    0x8b00
0x73fa: mov     rdi, qword ptr [rsp + 0x48]
0x73ff: xor     ebp, ebp
0x7401: call    0xc4b0
0x7406: xor     edi, edi
0x7408: call    0xd1a0
0x740d: mov     rdi, qword ptr [rsp + 0x48]
0x7412: call    0xd610
0x7417: mov     rdi, qword ptr [rsp + 0x48]
0x741c: xor     esi, esi
0x741e: call    0x193f0
0x7423: xor     edi, edi
0x7425: call    0xdbb0
0x742a: xor     edi, edi
0x742c: call    0xd1a0
0x7431: mov     rdi, qword ptr [rip + 0x23478]
0x7438: call    0xf480
0x743d: mov     rdi, qword ptr [rip + 0x23464]
0x7444: call    0xf480
0x7449: xor     eax, eax
0x744b: call    0xa480
0x7450: mov     rdi, qword ptr [rip + 0x23471]
0x7457: call    0xbb10
0x745c: xor     eax, eax

```

```

0x745e: mov     dword ptr [rip + 0x23474], 1
0x7468: call    0xc8c0
0x746d: mov     r14, rax
0x7470: jmp     0x565a
0x565a
0x7475: call    0x49d0
0x747a: nop     word ptr [rax + rax]
0x7480: endbr64
0x7484: xor     ebp, ebp
0x7486: mov     r9, rdx
0x7489: pop     rsi
0x748a: mov     rdx, rsp
0x748d: and     rsp, 0xffffffffffffffff
0x7491: push    rax
0x7492: push    rsp
0x7493: xor     r8d, r8d
0x7496: xor     ecx, ecx
0x7498: lea     rdi, [rip - 0x256f]
0x749f: call    qword ptr [rip + 0x21b23]
0x74a5: hlt
0x74a6: nop     word ptr cs:[rax + rax]
0x74b0: lea     rdi, [rip + 0x23379]
0x74b7: lea     rax, [rip + 0x23372]
0x74be: cmp     rax, rdi
0x74c1: je      0x74d8
0x74c3: mov     rax, qword ptr [rip + 0x21b06]
0x74ca: test    rax, rax
0x74cd: je      0x74d8
0x74cf: jmp     rax
rax
                                ->->->                                indirect JMP
0x8b00: endbr64
0x8b04: push    rbx
0x8b05: xor     eax, eax
0x8b07: mov     rbx, rdi
0x8b0a: call    0x9c90
0x8b0f: cmp     eax, 1
0x8b12: je      0x8b42
0x8b14: mov     rdi, qword ptr [rip + 0x21d65]
0x8b1b: mov     rcx, qword ptr [rip + 0x21dc6]
0x8b22: mov     r8, rbx
0x8b25: xor     eax, eax
0x8b27: lea     rdx, [rip + 0x179e8]
0x8b2e: mov     esi, 1
0x8b33: call    0x4e60
0x8b38: mov     edi, 1
0x8b3d: call    0x4e20
0x8b42: xor     eax, eax

```

```

0x8b44: call    0x8ad0
0x8b49: jmp      0x8b14
0x8b14:
0x8b4b: nop      dword ptr [rax + rax]
0x8b50: endbr64
0x8b54: push     r15
0x8b56: push     r14
0x8b58: push     r13
0x8b5a: push     r12
0x8b5c: mov      r12, rsi
0x8b5f: push     rbp
0x8b60: push     rbx
0x8b61: sub      rsp, 0x68
0x8b65: mov      r14, qword ptr [rdi + 8]
0x8b69: mov      qword ptr [rsp + 0x18], rdi
0x8b6e: mov      qword ptr [rsp + 8], rdx
0x8b73: mov      rdi, r14
0x8b76: mov      qword ptr [rsp + 0x10], rcx
0x8b7b: mov      rax, qword ptr fs:[0x28]
0x8b84: mov      qword ptr [rsp + 0x58], rax
0x8b89: xor      eax, eax
0x8b8b: mov      qword ptr [rsp + 0x48], 0
0x8b94: mov      qword ptr [rsp + 0x50], 0
0x8b9d: call     0x49b0
0x8ba2: lea      r13, [rax + 1]
0x8ba6: mov      rdi, r13
0x8ba9: call     0x1d670
0x8bae: mov      qword ptr [r12], rax
0x8bb2: mov      rbx, rax
0x8bb5: movzx    eax, byte ptr [r14]
0x8bb9: test     al, al
0x8bbb: je       0x925c
0x8bc1: mov      dword ptr [rsp + 0x24], 0
0x8bc9: jmp      0x8be5
0x8be5:
0x8bcb: nop      dword ptr [rax + rax]
0x8bd0: mov      byte ptr [rbx], al
0x8bd2: mov      r14, rbp
0x8bd5: add      rbx, 1
0x8bd9: movzx    eax, byte ptr [r14]
0x8bdd: test     al, al
0x8bdf: je       0x8d38
0x8be5: lea      rbp, [r14 + 1]
0x8be9: cmp      al, 0x25
0x8beb: jne      0x8bd0
0x8bed: call     0x4f00
0x8bf2: movsx    rcx, byte ptr [r14 + 1]
0x8bf7: mov      r15, qword ptr [rsp + 0x10]

```

```

0x8bfc: mov     rax, qword ptr [rax]
0x8bff: mov     rdx, rcx
0x8c02: test    byte ptr [rax + rcx*2 + 1], 2
0x8c07: lea     eax, [rcx - 0x3f]
0x8c0a: cmovne  r15, qword ptr [rsp + 8]
0x8c10: cmp     al, 0x3b
0x8c12: ja      0x8c38
0x8c14: lea     rcx, [rip + 0x174f1]
0x8c1b: movzx   eax, al
0x8c1e: movsxd  rax, dword ptr [rcx + rax*4]
0x8c22: add     rax, rcx
0x8c25: jmp     rax
rax
->->->                                indirect JMP
0x1d7a0: endbr64
0x1d7a4: push    r12
0x1d7a6: push    rbp
0x1d7a7: mov     rbp, rdi
0x1d7aa: sub     rsp, 8
0x1d7ae: call    0x49b0
0x1d7b3: lea     r12, [rax + 1]
0x1d7b7: mov     rdi, r12
0x1d7ba: call    0x4c20
0x1d7bf: test    rax, rax
0x1d7c2: je      0x1d7d9
0x1d7c4: add     rsp, 8
0x1d7c8: mov     rdx, r12
0x1d7cb: mov     rsi, rbp
0x1d7ce: mov     rdi, rax
0x1d7d1: pop     rbp
0x1d7d2: pop     r12
0x1d7d4: jmp     0x4bb0
0x4bb0
0x1d7d9: call    0x18cf0
0x1d7de: nop
0x1d7e0: endbr64
0x1d7e4: push    r15
0x1d7e6: mov     r15, rsi
0x1d7e9: push    r14
0x1d7eb: mov     r14, rcx
0x1d7ee: push    r13
0x1d7f0: mov     r13d, edx
0x1d7f3: push    r12
0x1d7f5: mov     r12, r8
0x1d7f8: push    rbp
0x1d7f9: mov     rbp, rdi
0x1d7fc: push    rbx
0x1d7fd: sub     rsp, 0x28

```

```

0x1d801:    movzx    ebx, byte ptr [rdi]
0x1d804:    mov      rax, qword ptr fs:[0x28]
0x1d80d:    mov      qword ptr [rsp + 0x18], rax
0x1d812:    xor      eax, eax
0x1d814:    lea      rax, [rsp + 0x10]
0x1d819:    test     rsi, rsi
0x1d81c:    cmov     r15, rax
0x1d820:    call     0x4f00
0x1d825:    mov      rcx, qword ptr [rax]
0x1d828:    mov      rax, rbp
0x1d82b:    jmp      0x1d838
0x1d838:
0x1d82d:    nop      dword ptr [rax]
0x1d830:    movzx    ebx, byte ptr [rax + 1]
0x1d834:    add      rax, 1
0x1d838:    movzx    edx, bl
0x1d83b:    test     byte ptr [rcx + rdx*2 + 1], 0x20
0x1d840:    jne      0x1d830
0x1d842:    cmp      bl, 0x2d
0x1d845:    je       0x1d83a
0x1d84b:    call     0x4860
0x1d850:    mov      edx, r13d
0x1d853:    mov      rsi, r15
0x1d856:    mov      rdi, rbp
0x1d859:    mov      dword ptr [rax], 0
0x1d85f:    mov      qword ptr [rsp + 8], rax
0x1d864:    call     0x4dd0
0x1d869:    mov      r13, qword ptr [r15]
0x1d86c:    mov      rcx, qword ptr [rsp + 8]
0x1d871:    mov      rbx, rax
0x1d874:    cmp      r13, rbp
0x1d877:    je       0x1d8d8
0x1d879:    mov      eax, dword ptr [rcx]
0x1d87b:    test     eax, eax
0x1d87d:    jne      0x1d8c0
0x1d87f:    xor      ebp, ebp
0x1d881:    test     r12, r12
0x1d884:    je       0x1d893
0x1d886:    movzx    edx, byte ptr [r13]
0x1d88b:    test     dl, dl
0x1d88d:    jne      0x1d958
0x1d893:    mov      qword ptr [r14], rbx
0x1d896:    mov      rax, qword ptr [rsp + 0x18]
0x1d89b:    sub      rax, qword ptr fs:[0x28]
0x1d8a4:    jne      0x1dbf9
0x1d8aa:    add      rsp, 0x28
0x1d8ae:    mov      eax, ebp
0x1d8b0:    pop      rbx

```

```

0x1d8b1:    pop     rbp
0x1d8b2:    pop     r12
0x1d8b4:    pop     r13
0x1d8b6:    pop     r14
0x1d8b8:    pop     r15
0x1d8ba:    ret
0x1d8bb:    nop     dword ptr [rax + rax]
0x1d8c0:    mov     ebp, 1
0x1d8c5:    cmp     eax, 0x22
0x1d8c8:    je      0x1d881
0x1d8ca:    mov     ebp, 4
0x1d8cf:    jmp     0x1d896
0x1d896:
0x1d8d1:    nop     dword ptr [rax]
0x1d8d8:    test    r12, r12
0x1d8db:    je      0x1d8ca
0x1d8dd:    movzx   edx, byte ptr [rbp]
0x1d8e1:    test    dl, dl
0x1d8e3:    je      0x1d8ca
0x1d8e5:    movsx   esi, dl
0x1d8e8:    mov     rdi, r12
0x1d8eb:    mov     byte ptr [rsp + 8], dl
0x1d8ef:    xor     ebp, ebp
0x1d8f1:    call    0x4a00
0x1d8f6:    mov     ebx, 1
0x1d8fb:    movzx   edx, byte ptr [rsp + 8]
0x1d900:    test    rax, rax
0x1d903:    je      0x1d8ca
0x1d905:    mov     esi, 0x30
0x1d90a:    mov     rdi, r12
0x1d90d:    mov     byte ptr [rsp + 8], dl
0x1d911:    call    0x4a00
0x1d916:    movzx   edx, byte ptr [rsp + 8]
0x1d91b:    test    rax, rax
0x1d91e:    je      0x1d97c
0x1d920:    movzx   eax, byte ptr [r13 + 1]
0x1d925:    cmp     al, 0x44
0x1d927:    je      0x1d9a2
0x1d929:    cmp     al, 0x69
0x1d92b:    je      0x1dbe0
0x1d931:    cmp     al, 0x42
0x1d933:    je      0x1d9a2
0x1d935:    sub     edx, 0x42
0x1d938:    cmp     dl, 0x35
0x1d93b:    ja      0x1d971
0x1d93d:    lea     rcx, [rip + 0x4c0c]
0x1d944:    movzx   edx, dl
0x1d947:    movsxd  rax, dword ptr [rcx + rdx*4]

```



```

0x1d94b:      add     rax, rcx
0x1d94e:      jmp     rax
rax
->->->                                indirect JMP
0x1c1e0:      lea     edi, [r12 + 5]
0x1c1e5:      movsxd  rdi, edi
0x1c1e8:      call    0x1d670
0x1c1ed:      mov     r11d, 0x733a
0x1c1f3:      mov     rsi, qword ptr [rsp + 0x20]
0x1c1f8:      mov     rbp, rax
0x1c1fb:      movzx   eax, byte ptr [rip + 0xe5eb]
0x1c202:      mov     word ptr [rbp + 2], r11w
0x1c207:      lea     rdi, [rbp + 4]
0x1c20b:      mov     byte ptr [rbp + 1], al
0x1c20e:      mov     byte ptr [rbp], al
0x1c211:      call    0x48b0
0x1c216:      lea     eax, [r12 + 4]
0x1c21b:      mov     dword ptr [rsp + 0xc], eax
0x1c21f:      test    eax, eax
0x1c221:      jg      0x1b6e6
0x1c227:      mov     rax, qword ptr [rsp + 0x30]
0x1c22c:      mov     qword ptr [rax], r15
0x1c22f:      cmp     qword ptr [rsp + 0x20], rbp
0x1c234:      je      0x1b67a
0x1c23a:      mov     rdi, rbp
0x1c23d:      call    0x18d10
0x1c242:      mov     dword ptr [rsp + 0x3c], 0
0x1c24a:      jmp     0x1b7c0
0x1b7c0
0x1c24f:      nop
0x1c250:      cmp     al, 0x5e
0x1c252:      je      0x1c478
0x1c258:      add     edx, 1
0x1c25b:      mov     esi, 0x24
0x1c260:      mov     dword ptr [rsp + 0x6c], edx
0x1c264:      jmp     0x1bd98
0x1bd98
0x1c269:      add     eax, 1
0x1c26c:      mov     rdi, qword ptr [rsp + 0x10]
0x1c271:      movsxd  rsi, eax
0x1c274:      call    0x1d690
0x1c279:      mov     qword ptr [rsp + 0x10], rax
0x1c27e:      jmp     0x1baeb
0x1baeb
0x1c283:      add     edx, 2
0x1c286:      xor     esi, esi
0x1c288:      movsxd  rdx, edx
0x1c28b:      nop     dword ptr [rax + rax]

```

```

0x1c290:    sub     eax, 0x30
0x1c293:    lea     ecx, [rsi + rsi*4]
0x1c296:    mov     r8d, edx
0x1c299:    movsx   eax, al
0x1c29c:    lea     esi, [rax + rcx*2]
0x1c29f:    movzx   eax, byte ptr [rbp + rdx]
0x1c2a4:    add     rdx, 1
0x1c2a8:    lea     ecx, [rax - 0x30]
0x1c2ab:    cmp     cl, 9
0x1c2ae:    jbe     0x1c290
0x1c2b0:    cmp     esi, 0x24
0x1c2b3:    sete    al
0x1c2b6:    cmp     esi, edi
0x1c2b8:    setge   dl
0x1c2bb:    or      eax, edx
0x1c2bd:    mov     dword ptr [rsp + 0x6c], r8d
0x1c2c2:    test    al, al
0x1c2c4:    je      0x1bdb3
0x1c2ca:    jmp     0x1bd98
0x1bd98:
0x1c2cf:    nop
0x1c2d0:    xor     ebx, ebx
0x1c2d2:    mov     byte ptr [rax], 0
0x1c2d5:    mov     rdi, r10
0x1c2d8:    mov     qword ptr [rsp + 0x48], r8
0x1c2dd:    call    0x18d10
0x1c2e2:    mov     r8, qword ptr [rsp + 0x48]
0x1c2e7:    mov     dword ptr [rip + 0xf81b], ebx
0x1c2ed:    mov     qword ptr [rip + 0xf81c], r8
0x1c2f4:    jmp     0x1be75
0x1be75:
0x1c2f9:    nop     dword ptr [rax]
0x1c300:    cmp     byte ptr [r10 + r14 + 1], 0x26
0x1c306:    lea     rax, [r14 + 1]
0x1c30a:    jne     0x1bcd8
0x1c310:    add     r13d, 1
0x1c314:    mov     r14, rax
0x1c317:    jmp     0x1bcd8
0x1bcd8:
0x1c31c:    nop     dword ptr [rax]
0x1c320:    add     r15d, r15d
0x1c323:    mov     rdi, rbp
0x1c326:    movsxd  rsi, r15d
0x1c329:    call    0x1d690
0x1c32e:    mov     r10, qword ptr [rip + 0xf7db]
0x1c335:    mov     rbp, rax
0x1c338:    jmp     0x1bce1
0x1bce1:

```

```

0x1c33d:    lea     r15d, [rax + r15*2]
0x1c341:    mov     rdi, rbp
0x1c344:    movsxd  rsi, r15d
0x1c347:    call    0x1d690
0x1c34c:    mov     r10, qword ptr [rip + 0xf7bd]
0x1c353:    mov     rbp, rax
0x1c356:    jmp     0x1bd1f
0x1bd1f:
0x1c35b:    xor     eax, eax
0x1c35d:    mov     rdi, r15
0x1c360:    call    0x1a380
0x1c365:    mov     eax, 2
0x1c36a:    jmp     0x1b67c
0x1b67c:
0x1c36f:    mov     r8, rbp
0x1c372:    movsxd  rax, ebx
0x1c375:    mov     r15, qword ptr [rsp + 0x58]
0x1c37a:    mov     rbp, qword ptr [rsp + 0x50]
0x1c37f:    add     rax, r8
0x1c382:    jmp     0x1c2d2
0x1c2d2:
0x1c387:    cmp     qword ptr [rip + 0xf789], 0
0x1c38f:    mov     eax, dword ptr [rip + 0xf777]
0x1c395:    jne     0x1bc29
0x1c39b:    mov     rbx, qword ptr [rip + 0xf74e]
0x1c3a2:    test    rbx, rbx
0x1c3a5:    je      0x1c3b0
0x1c3a7:    cmp     byte ptr [rbx], 0
0x1c3aa:    jne     0x1c442
0x1c3b0:    mov     dword ptr [rip + 0xf752], 0
0x1c3ba:    mov     r14, r15
0x1c3bd:    mov     r15, qword ptr [rsp + 0x28]
0x1c3c2:    mov     edx, dword ptr [rsp + 0x6c]
0x1c3c6:    mov     ecx, 2
0x1c3cb:    mov     esi, dword ptr [rsp + 0x44]
0x1c3cf:    jmp     0x1b9f7
0x1b9f7:
0x1c3d4:    mov     dword ptr [rsp + 0x28], 0
0x1c3dc:    mov     edx, dword ptr [rsp + 0x44]
0x1c3e0:    jmp     0x1bad1
0x1bad1:
0x1c3e5:    add     edx, 2
0x1c3e8:    mov     esi, 0x24
0x1c3ed:    mov     dword ptr [rsp + 0x6c], edx
0x1c3f1:    jmp     0x1bd98
0x1bd98:
0x1c3f6:    mov     esi, 0xffffffff
0x1c3fb:    jmp     0x1bd98

```

```

0x1bd98
0x1c400:    mov     r14, r15
0x1c403:    mov     r15, qword ptr [rsp + 0x28]
0x1c408:    jmp     0x1c3c2
0x1c3c2
0x1c40a:    mov     rax, qword ptr [rsp + 0x30]
0x1c40f:    mov     qword ptr [rax], r15
0x1c412:    jmp     0x1b67a
0x1b67a
0x1c417:    mov     esi, dword ptr [rsp + 0x50]
0x1c41b:    test    esi, esi
0x1c41d:    je      0x1baa5
0x1c423:    mov     edx, r14d
0x1c426:    mov     ecx, 2
0x1c42b:    mov     r14, r15
0x1c42e:    mov     r15, qword ptr [rsp + 0x28]
0x1c433:    jmp     0x1c3cb
0x1c3cb
0x1c435:    xor     ecx, ecx
0x1c437:    mov     edx, r12d
0x1c43a:    mov     esi, r13d
0x1c43d:    jmp     0x1bdbbe
0x1bdbbe
0x1c442:    mov     rdi, rbx
0x1c445:    call    0x49b0
0x1c44a:    lea     rdi, [rax + 1]
0x1c44e:    call    0x1d670
0x1c453:    mov     rsi, rbx
0x1c456:    mov     rdi, rax
0x1c459:    call    0x48b0
0x1c45e:    mov     rdi, rax
0x1c461:    mov     qword ptr [rip + 0xf6b0], rax
0x1c468:    call    0x49b0
0x1c46d:    mov     dword ptr [rip + 0xf699], eax
0x1c473:    jmp     0x1bc29
0x1bc29
0x1c478:    cmp     edi, 1
0x1c47b:    lea     r8d, [rdx + 1]
0x1c47f:    setle   al
0x1c482:    jmp     0x1c2bd
0x1c2bd
0x1c487:    call    0x49d0
0x1c48c:    nop     dword ptr [rax]
0x1c490:    endbr64
0x1c494:    push    rbx
0x1c495:    mov     rbx, rdx
0x1c498:    add     rsp, -0x80
0x1c49c:    mov     rax, qword ptr fs:[0x28]

```

```

0x1c4a5:    mov     qword ptr [rsp + 0x78], rax
0x1c4aa:    xor     eax, eax
0x1c4ac:    mov     rsi, rsp
0x1c4af:    call    0x4cb0
0x1c4b4:    test    eax, eax
0x1c4b6:    js      0x1c514
0x1c4b8:    mov     rax, qword ptr [rsp + 8]
0x1c4bd:    test    rax, rax
0x1c4c0:    cmov     rax, qword ptr [rsp]
0x1c4c5:    mov     qword ptr [rbx], rax
0x1c4c8:    mov     rax, qword ptr [rsp + 0x10]
0x1c4cd:    mov     qword ptr [rbx + 8], rax
0x1c4d1:    mov     rax, qword ptr [rsp + 0x18]
0x1c4d6:    mov     qword ptr [rbx + 0x10], rax
0x1c4da:    mov     rax, qword ptr [rsp + 0x20]
0x1c4df:    mov     qword ptr [rbx + 0x18], rax
0x1c4e3:    shr     rax, 0x3f
0x1c4e7:    mov     byte ptr [rbx + 0x20], al
0x1c4ea:    mov     rax, qword ptr [rsp + 0x28]
0x1c4ef:    mov     qword ptr [rbx + 0x28], rax
0x1c4f3:    mov     rax, qword ptr [rsp + 0x30]
0x1c4f8:    mov     qword ptr [rbx + 0x30], rax
0x1c4fc:    xor     eax, eax
0x1c4fe:    mov     rdx, qword ptr [rsp + 0x78]
0x1c503:    sub     rdx, qword ptr fs:[0x28]
0x1c50c:    jne     0x1c51b
0x1c50e:    sub     rsp, -0x80
0x1c512:    pop     rbx
0x1c513:    ret
0x1c514:    mov     eax, 0xffffffff
0x1c519:    jmp     0x1c4fe
0x1c4fe
0x1c51b:    call    0x49d0
0x1c520:    endbr64
0x1c524:    push    r15
0x1c526:    push    r14
0x1c528:    push    r13
0x1c52a:    mov     r13, r8
0x1c52d:    push    r12
0x1c52f:    mov     r12, rcx
0x1c532:    push    rbp
0x1c533:    push    rbx
0x1c534:    mov     rbx, rdi
0x1c537:    sub     rsp, 0x98
0x1c53e:    mov     qword ptr [rsp], rsi
0x1c542:    mov     qword ptr [rsp + 0x18], r8
0x1c547:    mov     dword ptr [rsp + 0x48], edx
0x1c54b:    mov     rax, qword ptr fs:[0x28]

```

```

0x1c554:    mov     qword ptr [rsp + 0x88], rax
0x1c55c:    mov     eax, edx
0x1c55e:    and     edx, 0x20
0x1c561:    mov     dword ptr [rsp + 0x30], edx
0x1c565:    and     eax, 3
0x1c568:    cmp     edx, 1
0x1c56b:    mov     dword ptr [rsp + 0x10], eax
0x1c56f:    sbb     eax, eax
0x1c571:    and     eax, 0xfffffffffe8
0x1c574:    add     eax, 0x400
0x1c579:    mov     dword ptr [rsp + 0x34], eax
0x1c57d:    call    0x4900
0x1c582:    mov     r15, qword ptr [rax]
0x1c585:    mov     r14, rax
0x1c588:    mov     rdi, r15
0x1c58b:    call    0x49b0
0x1c590:    mov     rcx, qword ptr [r14 + 0x10]
0x1c594:    mov     r14, qword ptr [r14 + 8]
0x1c598:    mov     rbp, rax
0x1c59b:    sub     rax, 1
0x1c59f:    cmp     rax, 0x10
0x1c5a3:    mov     eax, 1
0x1c5a8:    mov     rdi, r14
0x1c5ab:    mov     qword ptr [rsp + 0x38], rcx
0x1c5b0:    cmovae  rbp, rax
0x1c5b4:    lea     rax, [rip + 0x5123]
0x1c5bb:    cmovae  r15, rax
0x1c5bf:    call    0x49b0
0x1c5c4:    cmp     rax, 0x10
0x1c5c8:    lea     rax, [rip + 0x5e16]
0x1c5cf:    cmova   r14, rax
0x1c5d3:    mov     rax, qword ptr [rsp]
0x1c5d7:    add     rax, 0x288
0x1c5dd:    mov     qword ptr [rsp + 8], rax
0x1c5e2:    cmp     r13, r12
0x1c5e5:    ja      0x1c740
0x1c5eb:    mov     rax, r12
0x1c5ee:    xor     edx, edx
0x1c5f0:    div     r13
0x1c5f3:    test    rdx, rdx
0x1c5f6:    jne     0x1c604
0x1c5f8:    mul     rbx
0x1c5fb:    mov     rcx, rax
0x1c5fe:    jno     0x1cb98
0x1c604:    mov     qword ptr [rsp + 0x20], r12
0x1c609:    fild    qword ptr [rsp + 0x20]
0x1c60d:    test    r12, r12
0x1c610:    js      0x1c928

```

```

0x1c616:    mov     rax, qword ptr [rsp + 0x18]
0x1c61b:    mov     qword ptr [rsp + 0x20], rax
0x1c620:    fild    qword ptr [rsp + 0x20]
0x1c624:    test    rax, rax
0x1c627:    js      0x1c908
0x1c62d:    mov     qword ptr [rsp + 0x20], rbx
0x1c632:    fdivp    st(1)
0x1c634:    fild    qword ptr [rsp + 0x20]
0x1c638:    test    rbx, rbx
0x1c63b:    js      0x1c8f0
0x1c641:    fmulp    st(1)
0x1c643:    test    byte ptr [rsp + 0x48], 0x10
0x1c648:    je      0x1c938
0x1c64e:    fild    dword ptr [rsp + 0x34]
0x1c652:    xor     ebx, ebx
0x1c654:    fld     st(0)
0x1c656:    jmp     0x1c664
0x1c664:
0x1c658:    nop     dword ptr [rax + rax]
0x1c660:    fstp    st(1)
0x1c662:    fxch    st(2)
0x1c664:    fld     st(0)
0x1c666:    add     ebx, 1
0x1c669:    fmul    st(2)
0x1c66b:    fxch    st(3)
0x1c66d:    fcomi    st(3)
0x1c66f:    jb      0x1c680
0x1c671:    cmp     ebx, 8
0x1c674:    jne     0x1c660
0x1c676:    fstp    st(2)
0x1c678:    fstp    st(2)
0x1c67a:    jmp     0x1c684
0x1c684:
0x1c67c:    nop     dword ptr [rax]
0x1c680:    fstp    st(2)
0x1c682:    fstp    st(2)
0x1c684:    lea     r15, [rbp + 1]
0x1c688:    fdivrp    st(1)
0x1c68a:    cmp     dword ptr [rsp + 0x30], 1
0x1c68f:    adc     rbp, 2
0x1c693:    cmp     dword ptr [rsp + 0x10], 1
0x1c698:    je      0x1cc70
0x1c69e:    fld     xword ptr [rip + 0x5e9c]
0x1c6a4:    fcomip    st(1)
0x1c6a6:    jbe     0x1cb10
0x1c6ac:    fld     dword ptr [rip + 0x5e7e]
0x1c6b2:    fxch    st(1)
0x1c6b4:    fcomi    st(1)

```

```

0x1c6b6:    jae     0x1ce00
0x1c6bc:    fstp    st(1)
0x1c6be:    fnstcw  word ptr [rsp + 0x4e]
0x1c6c2:    movzx   eax, word ptr [rsp + 0x4e]
0x1c6c7:    or      ah, 0xc
0x1c6ca:    mov     word ptr [rsp + 0x4c], ax
0x1c6cf:    fld     st(0)
0x1c6d1:    fldcw   word ptr [rsp + 0x4c]
0x1c6d5:    fistp   qword ptr [rsp + 0x20]
0x1c6d9:    fldcw   word ptr [rsp + 0x4e]
0x1c6dd:    mov     rax, qword ptr [rsp + 0x20]
0x1c6e2:    mov     qword ptr [rsp + 0x20], rax
0x1c6e7:    fild    qword ptr [rsp + 0x20]
0x1c6eb:    test    rax, rax
0x1c6ee:    jns     0x1c6f6
0x1c6f0:    fadd    dword ptr [rip + 0x5e36]
0x1c6f6:    mov     ecx, dword ptr [rsp + 0x10]
0x1c6fa:    test    ecx, ecx
0x1c6fc:    jne     0x1cb18
0x1c702:    fxch    st(1)
0x1c704:    fucomi  st(1)
0x1c706:    jp      0x1c712
0x1c708:    je      0x1cb20
0x1c70e:    fstp    st(1)
0x1c710:    jmp     0x1c718
0x1c718:
0x1c712:    fstp    st(1)
0x1c714:    nop     dword ptr [rax]
0x1c718:    add     rax, 1
0x1c71c:    mov     qword ptr [rsp + 0x20], rax
0x1c721:    fild    qword ptr [rsp + 0x20]
0x1c725:    jns     0x1cb1c
0x1c72b:    fadd    dword ptr [rip + 0x5dfb]
0x1c731:    fxch    st(1)
0x1c733:    jmp     0x1cb20
0x1cb20:
0x1c738:    nop     dword ptr [rax + rax]
0x1c740:    test    r12, r12
0x1c743:    je      0x1c604
0x1c749:    mov     rax, qword ptr [rsp + 0x18]
0x1c74e:    xor     edx, edx
0x1c750:    div     r12
0x1c753:    mov     r8, rax
0x1c756:    test    rdx, rdx
0x1c759:    jne     0x1c604
0x1c75f:    mov     rax, rbx
0x1c762:    xor     edx, edx
0x1c764:    div     r8

```



```

0x1c767:    mov     rcx, rax
0x1c76a:    lea     rax, [rdx + rdx*4]
0x1c76e:    xor     edx, edx
0x1c770:    add     rax, rax
0x1c773:    div     r8
0x1c776:    add     rdx, rdx
0x1c779:    mov     edi, eax
0x1c77b:    cmp     r8, rdx
0x1c77e:    jbe     0x1cde8
0x1c784:    xor     esi, esi
0x1c786:    test    rdx, rdx
0x1c789:    setne   sil
0x1c78d:    mov     r10d, dword ptr [rsp + 0x48]
0x1c792:    and     r10d, 0x10
0x1c796:    je      0x1cbab
0x1c79c:    mov     r8d, dword ptr [rsp + 0x34]
0x1c7a1:    xor     ebx, ebx
0x1c7a3:    mov     r11, r8
0x1c7a6:    cmp     r8, rcx
0x1c7a9:    jbe     0x1cacf
0x1c7af:    cmp     dword ptr [rsp + 0x10], 1
0x1c7b4:    mov     r8, qword ptr [rsp + 8]
0x1c7b9:    je      0x1cdc0
0x1c7bf:    mov     r11d, dword ptr [rsp + 0x10]
0x1c7c4:    test    r11d, r11d
0x1c7c7:    jne     0x1c7e8
0x1c7c9:    add     esi, edi
0x1c7cb:    test    esi, esi
0x1c7cd:    jle     0x1c7e8
0x1c7cf:    add     rcx, 1
0x1c7d3:    test    r10d, r10d
0x1c7d6:    je      0x1c7e8
0x1c7d8:    mov     eax, dword ptr [rsp + 0x34]
0x1c7dc:    cmp     rax, rcx
0x1c7df:    je      0x1cec2
0x1c7e5:    nop     dword ptr [rax]
0x1c7e8:    movabs  rsi, 0xffffffffffffffff
0x1c7f2:    mov     rbp, r8
0x1c7f5:    nop     dword ptr [rax]
0x1c7f8:    mov     rax, rcx
0x1c7fb:    sub     rbp, 1
0x1c7ff:    mul     rsi
0x1c802:    mov     rax, rcx
0x1c805:    shr     rdx, 3
0x1c809:    lea     rdi, [rdx + rdx*4]
0x1c80d:    add     rdi, rdi
0x1c810:    sub     rax, rdi
0x1c813:    add     eax, 0x30

```

```

0x1c816:    mov     byte ptr [rbp], al
0x1c819:    mov     rax, rcx
0x1c81c:    mov     rcx, rdx
0x1c81f:    cmp     rax, 9
0x1c823:    ja      0x1c7f8
0x1c825:    test    byte ptr [rsp + 0x48], 4
0x1c82a:    je      0x1ca2c
0x1c830:    mov     rdi, r14
0x1c833:    mov     r12, r8
0x1c836:    mov     qword ptr [rsp + 0x40], r8
0x1c83b:    mov     r13, -1
0x1c842:    call    0x49b0
0x1c847:    sub     r12, rbp
0x1c84a:    mov     rsi, rbp
0x1c84d:    mov     ecx, 0x29
0x1c852:    mov     r15, rax
0x1c855:    lea     rax, [rsp + 0x50]
0x1c85a:    mov     rdx, r12
0x1c85d:    mov     rdi, rax
0x1c860:    mov     qword ptr [rsp + 0x10], rax
0x1c865:    call    0x4b70
0x1c86a:    mov     dword ptr [rsp + 0x20], ebx
0x1c86e:    mov     rbp, qword ptr [rsp + 0x40]
0x1c873:    mov     rbx, r12
0x1c876:    mov     r12, qword ptr [rsp + 0x38]
0x1c87b:    jmp     0x1c8bb
0x1c8bb:
0x1c87d:    nop     dword ptr [rax]
0x1c880:    cmp     r13, rbx
0x1c883:    mov     rax, qword ptr [rsp + 0x10]
0x1c888:    cmova   r13, rbx
0x1c88c:    sub     rbx, r13
0x1c88f:    lea     rsi, [rax + rbx]
0x1c893:    sub     rbp, r13
0x1c896:    mov     rdx, r13
0x1c899:    mov     rdi, rbp
0x1c89c:    call    0x4bb0
0x1c8a1:    test    rbx, rbx
0x1c8a4:    je      0x1caa0
0x1c8aa:    sub     rbp, r15
0x1c8ad:    mov     rdx, r15
0x1c8b0:    mov     rsi, r14
0x1c8b3:    mov     rdi, rbp
0x1c8b6:    call    0x4bb0
0x1c8bb:    movzx   eax, byte ptr [r12]
0x1c8c0:    test    al, al
0x1c8c2:    je      0x1c880
0x1c8c4:    cmp     al, 0x7e

```

```

0x1c8c6:    ja      0x1ca88
0x1c8cc:    cmp     rax, rbx
0x1c8cf:    cmova   rax, rbx
0x1c8d3:    sub     rbx, rax
0x1c8d6:    mov     r13, rax
0x1c8d9:    mov     rax, qword ptr [rsp + 0x10]
0x1c8de:    lea     rsi, [rax + rbx]
0x1c8e2:    add     r12, 1
0x1c8e6:    jmp     0x1c893
0x1c893:
0x1c8e8:    nop     dword ptr [rax + rax]
0x1c8f0:    fadd    dword ptr [rip + 0x5c36]
0x1c8f6:    fmulp   st(1)
0x1c8f8:    test    byte ptr [rsp + 0x48], 0x10
0x1c8fd:    jne     0x1c64e
0x1c903:    jmp     0x1c938
0x1c938:
0x1c905:    nop     dword ptr [rax]
0x1c908:    fadd    dword ptr [rip + 0x5c1e]
0x1c90e:    mov     qword ptr [rsp + 0x20], rbx
0x1c913:    fdivp   st(1)
0x1c915:    fild    qword ptr [rsp + 0x20]
0x1c919:    test    rbx, rbx
0x1c91c:    jns     0x1c641
0x1c922:    jmp     0x1c8f0
0x1c8f0:
0x1c924:    nop     dword ptr [rax]
0x1c928:    fadd    dword ptr [rip + 0x5bfe]
0x1c92e:    jmp     0x1c616
0x1c616:
0x1c933:    nop     dword ptr [rax + rax]
0x1c938:    cmp     dword ptr [rsp + 0x10], 1
0x1c93d:    je      0x1c9ca
0x1c943:    fld     xword ptr [rip + 0x5bf7]
0x1c949:    fcomip  st(1)
0x1c94b:    jbe     0x1c9ca
0x1c94d:    fld     dword ptr [rip + 0x5bdd]
0x1c953:    fxch    st(1)
0x1c955:    fcomi   st(1)
0x1c957:    jae     0x1ce60
0x1c95d:    fstp    st(1)
0x1c95f:    fnstcw  word ptr [rsp + 0x4e]
0x1c963:    movzx   eax, word ptr [rsp + 0x4e]
0x1c968:    or      ah, 0xc
0x1c96b:    mov     word ptr [rsp + 0x4c], ax
0x1c970:    fld     st(0)
0x1c972:    fldcw   word ptr [rsp + 0x4c]
0x1c976:    fistp   qword ptr [rsp + 0x20]

```

```

0x1c97a:      fldcw   word ptr [rsp + 0x4e]
0x1c97e:      mov     rax, qword ptr [rsp + 0x20]
0x1c983:      mov     qword ptr [rsp + 0x20], rax
0x1c988:      fild    qword ptr [rsp + 0x20]
0x1c98c:      test    rax, rax
0x1c98f:      jns     0x1c997
0x1c991:      fadd     dword ptr [rip + 0x5b95]
0x1c997:      mov     r8d, dword ptr [rsp + 0x10]
0x1c99c:      test    r8d, r8d
0x1c99f:      jne     0x1c9c8
0x1c9a1:      fxch     st(1)
0x1c9a3:      fucomip  st(1)
0x1c9a5:      jp      0x1c9ad
0x1c9a7:      je      0x1c9ca
0x1c9a9:      fstp     st(0)
0x1c9ab:      jmp     0x1c9af
0x1c9af:
0x1c9ad:      fstp     st(0)
0x1c9af:      add     rax, 1
0x1c9b3:      mov     qword ptr [rsp + 0x10], rax
0x1c9b8:      fild    qword ptr [rsp + 0x10]
0x1c9bc:      js      0x1cee7
0x1c9c2:      jmp     0x1c9ca
0x1c9ca:
0x1c9c4:      nop     dword ptr [rax]
0x1c9c8:      fstp     st(1)
0x1c9ca:      sub     rsp, 0x10
0x1c9ce:      mov     rdx, -1
0x1c9d5:      mov     esi, 1
0x1c9da:      xor     eax, eax
0x1c9dc:      mov     rbx, qword ptr [rsp + 0x10]
0x1c9e1:      lea     rcx, [rip + 0x5adc]
0x1c9e8:      fstp     xword ptr [rsp]
0x1c9eb:      mov     rdi, rbx
0x1c9ee:      call    0x4f10
0x1c9f3:      mov     rdi, rbx
0x1c9f6:      mov     ebx, 0xffffffff
0x1c9fb:      call    0x49b0
0x1ca00:      pop     rsi
0x1ca01:      pop     rdi
0x1ca02:      mov     rdx, rax
0x1ca05:      mov     r12, rax
0x1ca08:      mov     rbp, qword ptr [rsp + 8]
0x1ca0d:      mov     rsi, qword ptr [rsp]
0x1ca11:      sub     rbp, rdx
0x1ca14:      mov     rdi, rbp
0x1ca17:      call    0x4d40
0x1ca1c:      lea     r8, [rbp + r12]

```

```

0x1ca21:    test    byte ptr [rsp + 0x48], 4
0x1ca26:    jne     0x1c830
0x1ca2c:    test    byte ptr [rsp + 0x48], 0x80
0x1ca31:    je      0x1ca4f
0x1ca33:    cmp     ebx, -1
0x1ca36:    je      0x1cbb8
0x1ca3c:    mov     eax, dword ptr [rsp + 0x48]
0x1ca40:    and     eax, 0x100
0x1ca45:    mov     esi, eax
0x1ca47:    or      esi, ebx
0x1ca49:    jne     0x1cd08
0x1ca4f:    mov     rax, qword ptr [rsp + 8]
0x1ca54:    mov     byte ptr [rax], 0
0x1ca57:    mov     rax, qword ptr [rsp + 0x88]
0x1ca5f:    sub     rax, qword ptr fs:[0x28]
0x1ca68:    jne     0x1d0eb
0x1ca6e:    add     rsp, 0x98
0x1ca75:    mov     rax, rbp
0x1ca78:    pop     rbx
0x1ca79:    pop     rbp
0x1ca7a:    pop     r12
0x1ca7c:    pop     r13
0x1ca7e:    pop     r14
0x1ca80:    pop     r15
0x1ca82:    ret
0x1ca83:    nop     dword ptr [rax + rax]
0x1ca88:    mov     r13, rbx
0x1ca8b:    mov     rsi, qword ptr [rsp + 0x10]
0x1ca90:    xor     ebx, ebx
0x1ca92:    jmp     0x1c8e2
0x1c8e2
0x1ca97:    nop     word ptr [rax + rax]
0x1caa0:    mov     ebx, dword ptr [rsp + 0x20]
0x1caa4:    jmp     0x1ca2c
0x1ca2c
0x1caa6:    nop     word ptr cs:[rax + rax]
0x1cab0:    test    esi, esi
0x1cab2:    setne   sil
0x1cab6:    movzx   esi, sil
0x1caba:    add     ebx, 1
0x1cabd:    cmp     r8, r9
0x1cac0:    ja      0x1cf18
0x1cac6:    cmp     ebx, 8
0x1cac9:    je      0x1c7af
0x1cacf:    mov     rax, rcx
0x1cad2:    xor     edx, edx
0x1cad4:    mov     ecx, esi
0x1cad6:    div     r8

```

```

0x1cad9:      sar      ecx, 1
0x1cadb:      mov      r9, rax
0x1cade:      lea      eax, [rdx + rdx*4]
0x1cae1:      xor      edx, edx
0x1cae3:      lea      eax, [rdi + rax*2]
0x1cae6:      div      r11d
0x1cae9:      lea      edx, [rcx + rdx*2]
0x1caec:      mov      edi, eax
0x1caee:      mov      rcx, r9
0x1caf1:      add      esi, edx
0x1caf3:      cmp      r11d, edx
0x1caf6:      ja       0x1cab0
0x1caf8:      cmp      r11d, esi
0x1cafb:      setb     sil
0x1caff:      movzx    esi, sil
0x1cb03:      add      esi, 2
0x1cb06:      jmp      0x1caba
0x1caba
0x1cb08:      nop      dword ptr [rax + rax]
0x1cb10:      fld      st(0)
0x1cb12:      jmp      0x1cb20
0x1cb20
0x1cb14:      nop      dword ptr [rax]
0x1cb18:      fxch     st(1)
0x1cb1a:      jmp      0x1cb20
0x1cb20
0x1cb1c:      fxch     st(1)
0x1cb1e:      nop
0x1cb20:      fstp     xword ptr [rsp + 0x20]
0x1cb24:      sub      rsp, 0x10
0x1cb28:      mov      esi, 1
0x1cb2d:      xor      eax, eax
0x1cb2f:      mov      r12, qword ptr [rsp + 0x10]
0x1cb34:      lea      rcx, [rip + 0x598f]
0x1cb3b:      mov      rdx, -1
0x1cb42:      mov      rdi, r12
0x1cb45:      fstp     xword ptr [rsp]
0x1cb48:      call     0x4f10
0x1cb4d:      mov      rdi, r12
0x1cb50:      call     0x49b0
0x1cb55:      mov      rdx, rax
0x1cb58:      pop      rax
0x1cb59:      pop      rcx
0x1cb5a:      fld      xword ptr [rsp + 0x20]
0x1cb5e:      cmp      rdx, rbp
0x1cb61:      ja       0x1cd20
0x1cb67:      test     byte ptr [rsp + 0x48], 8
0x1cb6c:      je       0x1cb88

```

```

0x1cb6e:      mov     rax, qword ptr [rsp]
0x1cb72:      cmp     byte ptr [rax + rdx - 1], 0x30
0x1cb77:      je      0x1ce40
0x1cb7d:      fstp     st(0)
0x1cb7f:      jmp     0x1cb8a
0x1cb8a:
0x1cb81:      nop     dword ptr [rax]
0x1cb88:      fstp     st(0)
0x1cb8a:      mov     r12, rdx
0x1cb8d:      sub     r12, r15
0x1cb90:      jmp     0x1ca08
0x1ca08:
0x1cb95:      nop     dword ptr [rax]
0x1cb98:      mov     r10d, dword ptr [rsp + 0x48]
0x1cb9d:      xor     esi, esi
0x1cb9f:      xor     edi, edi
0x1cba1:      and     r10d, 0x10
0x1cba5:      jne     0x1c79c
0x1cbab:      mov     ebx, 0xffffffff
0x1cbb0:      jmp     0x1c7af
0x1c7af:
0x1cbb5:      nop     dword ptr [rax]
0x1cbb8:      mov     rcx, qword ptr [rsp + 0x18]
0x1cbbd:      cmp     rcx, 1
0x1cbc1:      jbe     0x1cfca
0x1cbc7:      mov     edx, dword ptr [rsp + 0x34]
0x1cbcb:      mov     ebx, 1
0x1cbd0:      mov     eax, 1
0x1cbd5:      nop     dword ptr [rax]
0x1cbd8:      imul    rax, rdx
0x1cbdc:      cmp     rcx, rax
0x1cbdf:      jbe     0x1cbe9
0x1cbe1:      add     ebx, 1
0x1cbe4:      cmp     ebx, 8
0x1cbe7:      jne     0x1cbd8
0x1cbe9:      mov     esi, dword ptr [rsp + 0x48]
0x1cbec:      mov     eax, esi
0x1cbef:      and     eax, 0x100
0x1cbf4:      and     esi, 0x40
0x1cbf7:      je      0x1cc18
0x1cbf9:      mov     rsi, qword ptr [rsp]
0x1cbfd:      lea     rdi, [rsi + 0x289]
0x1cc04:      mov     byte ptr [rsi + 0x288], 0x20
0x1cc0b:      mov     qword ptr [rsp + 8], rdi
0x1cc10:      test    ebx, ebx
0x1cc12:      je      0x1d105
0x1cc18:      mov     rsi, qword ptr [rsp + 8]
0x1cc1d:      mov     r9d, dword ptr [rsp + 0x30]

```

```

0x1cc22:    lea     rdx, [rsi + 1]
0x1cc26:    test    r9d, r9d
0x1cc29:    jne     0x1cc34
0x1cc2b:    cmp     ebx, 1
0x1cc2e:    je      0x1cf00
0x1cc34:    movsxd  rbx, ebx
0x1cc37:    lea     rcx, [rip + 0x58e2]
0x1cc3e:    movzx   ecx, byte ptr [rcx + rbx]
0x1cc42:    mov     byte ptr [rsi], cl
0x1cc44:    test    eax, eax
0x1cc46:    je      0x1cf0b
0x1cc4c:    mov     r8d, dword ptr [rsp + 0x30]
0x1cc51:    test    r8d, r8d
0x1cc54:    jne     0x1ce30
0x1cc5a:    lea     rax, [rdx + 1]
0x1cc5e:    mov     byte ptr [rdx], 0x42
0x1cc61:    mov     qword ptr [rsp + 8], rax
0x1cc66:    jmp     0x1ca4f
0x1ca4f:
0x1cc6b:    nop     dword ptr [rax + rax]
0x1cc70:    sub     rsp, 0x10
0x1cc74:    mov     rdx, -1
0x1cc7b:    mov     esi, 1
0x1cc80:    xor     eax, eax
0x1cc82:    fld     st(0)
0x1cc84:    fstp    xword ptr [rsp]
0x1cc87:    mov     r12, qword ptr [rsp + 0x10]
0x1cc8c:    lea     rcx, [rip + 0x5837]
0x1cc93:    mov     rdi, r12
0x1cc96:    fstp    xword ptr [rsp + 0x30]
0x1cc9a:    call    0x4f10
0x1cc9f:    mov     rdi, r12
0x1cca2:    call    0x49b0
0x1cca7:    pop     rsi
0x1cca8:    pop     rdi
0x1cca9:    fld     xword ptr [rsp + 0x20]
0x1ccad:    cmp     rax, rbp
0x1ccb0:    mov     rdx, rax
0x1ccb3:    jbe     0x1cb67
0x1ccb9:    fld     dword ptr [rip + 0x5875]
0x1ccbf:    fmul    st(1), st(0)
0x1ccc1:    fdivp   st(1)
0x1ccc3:    sub     rsp, 0x10
0x1ccc7:    mov     rdx, -1
0x1ccce:    xor     eax, eax
0x1ccd0:    mov     r15, qword ptr [rsp + 0x10]
0x1ccd5:    lea     rcx, [rip + 0x57e8]
0x1ccdc:    mov     esi, 1

```



```

0x1cce1:      mov     rdi, r15
0x1cce4:      fstp    xword ptr [rsp]
0x1cce7:      call   0x4f10
0x1ccec:      mov     rdi, r15
0x1ccef:      call   0x49b0
0x1ccf4:      pop     r13
0x1ccf6:      pop     r15
0x1ccf8:      mov     rdx, rax
0x1ccfb:      mov     r12, rax
0x1ccfe:      jmp     0x1ca08
0x1ca08:
0x1cd03:      nop     dword ptr [rax + rax]
0x1cd08:      test   byte ptr [rsp + 0x48], 0x40
0x1cd0d:      je      0x1cc10
0x1cd13:      jmp     0x1cbf9
0x1cbf9:
0x1cd18:      nop     dword ptr [rax + rax]
0x1cd20:      fld     dword ptr [rip + 0x580e]
0x1cd26:      fmul    st(1), st(0)
0x1cd28:      fld     xword ptr [rip + 0x5812]
0x1cd2e:      fcomip  st(2)
0x1cd30:      jbe     0x1ccc1
0x1cd32:      fld     dword ptr [rip + 0x57f8]
0x1cd38:      fxch    st(2)
0x1cd3a:      fcomi   st(2)
0x1cd3c:      jae     0x1ce90
0x1cd42:      fstp    st(2)
0x1cd44:      fxch    st(1)
0x1cd46:      fnstcw  word ptr [rsp + 0x4e]
0x1cd4a:      movzx   eax, word ptr [rsp + 0x4e]
0x1cd4f:      or      ah, 0xc
0x1cd52:      mov     word ptr [rsp + 0x4c], ax
0x1cd57:      fld     st(0)
0x1cd59:      fldcw   word ptr [rsp + 0x4c]
0x1cd5d:      fistp   qword ptr [rsp + 0x20]
0x1cd61:      fldcw   word ptr [rsp + 0x4e]
0x1cd65:      mov     rax, qword ptr [rsp + 0x20]
0x1cd6a:      mov     qword ptr [rsp + 0x20], rax
0x1cd6f:      fild    qword ptr [rsp + 0x20]
0x1cd73:      test    rax, rax
0x1cd76:      jns     0x1cd7e
0x1cd78:      fadd    dword ptr [rip + 0x57ae]
0x1cd7e:      mov     edx, dword ptr [rsp + 0x10]
0x1cd82:      test    edx, edx
0x1cd84:      jne     0x1cdb0
0x1cd86:      fucomi  st(1)
0x1cd88:      fstp    st(1)
0x1cd8a:      jp      0x1cd92

```

```

0x1cd8c:      je      0x1cdb2
0x1cd8e:      fstp     st(0)
0x1cd90:      jmp     0x1cd94
0x1cd94:
0x1cd92:      fstp     st(0)
0x1cd94:      add      rax, 1
0x1cd98:      mov      qword ptr [rsp + 0x10], rax
0x1cd9d:      fild      qword ptr [rsp + 0x10]
0x1cda1:      js       0x1cef2
0x1cda7:      jmp      0x1cdb2
0x1cdb2:
0x1cda9:      nop       dword ptr [rax]
0x1cdb0:      fstp     st(1)
0x1cdb2:      fxch     st(1)
0x1cdb4:      jmp      0x1ccc1
0x1ccc1:
0x1cdb9:      nop       dword ptr [rax]
0x1cdc0:      mov      rax, rcx
0x1cdc3:      movsxd   rsi, esi
0x1cdc6:      and      eax, 1
0x1cdc9:      add      rax, rsi
0x1dcc:      setne    al
0x1cdcf:      movzx    eax, al
0x1cdd2:      add      eax, edi
0x1cdd4:      cmp      eax, 5
0x1cdd7:      jg       0x1c7cf
0x1cddd:      jmp      0x1c7e8
0x1c7e8:
0x1cde2:      nop       word ptr [rax + rax]
0x1cde8:      mov      esi, 2
0x1cded:      mov      eax, 3
0x1cdf2:      cmovb    esi, eax
0x1cdf5:      jmp      0x1c78d
0x1c78d:
0x1cdfa:      nop       word ptr [rax + rax]
0x1ce00:      fnstcw   word ptr [rsp + 0x4e]
0x1ce04:      fsubr     st(1), st(0)
0x1ce06:      fxch     st(1)
0x1ce08:      movzx    eax, word ptr [rsp + 0x4e]
0x1ce0d:      or       ah, 0xc
0x1ce10:      mov      word ptr [rsp + 0x4c], ax
0x1ce15:      fldcw    word ptr [rsp + 0x4c]
0x1ce19:      fistp     qword ptr [rsp + 0x20]
0x1ce1d:      fldcw    word ptr [rsp + 0x4e]
0x1ce21:      mov      rax, qword ptr [rsp + 0x20]
0x1ce26:      btc      rax, 0x3f
0x1ce2b:      jmp      0x1c6e2
0x1c6e2:

```

```

0x1ce30:    mov     byte ptr [rsi + 1], 0x69
0x1ce34:    lea     rdx, [rsi + 2]
0x1ce38:    jmp     0x1cc5a
0x1cc5a:
0x1ce3d:    nop     dword ptr [rax]
0x1ce40:    fld     dword ptr [rip + 0x56ee]
0x1ce46:    cmp     dword ptr [rsp + 0x10], 1
0x1ce4b:    fmul    st(1), st(0)
0x1ce4d:    jne     0x1cd28
0x1ce53:    jmp     0x1ccc1
0x1ccc1:
0x1ce58:    nop     dword ptr [rax + rax]
0x1ce60:    fnstcw  word ptr [rsp + 0x4e]
0x1ce64:    fsubr   st(1), st(0)
0x1ce66:    fxch    st(1)
0x1ce68:    movzx   eax, word ptr [rsp + 0x4e]
0x1ce6d:    or      ah, 0xc
0x1ce70:    mov     word ptr [rsp + 0x4c], ax
0x1ce75:    fldcw   word ptr [rsp + 0x4c]
0x1ce79:    fistp   qword ptr [rsp + 0x20]
0x1ce7d:    fldcw   word ptr [rsp + 0x4e]
0x1ce81:    mov     rax, qword ptr [rsp + 0x20]
0x1ce86:    btc     rax, 0x3f
0x1ce8b:    jmp     0x1c983
0x1c983:
0x1ce90:    fnstcw  word ptr [rsp + 0x4e]
0x1ce94:    fsubr   st(2), st(0)
0x1ce96:    fxch    st(2)
0x1ce98:    movzx   eax, word ptr [rsp + 0x4e]
0x1ce9d:    or      ah, 0xc
0x1cea0:    mov     word ptr [rsp + 0x4c], ax
0x1cea5:    fldcw   word ptr [rsp + 0x4c]
0x1cea9:    fistp   qword ptr [rsp + 0x20]
0x1cead:    fldcw   word ptr [rsp + 0x4e]
0x1ceb1:    fxch    st(1)
0x1ceb3:    mov     rax, qword ptr [rsp + 0x20]
0x1ceb8:    btc     rax, 0x3f
0x1cebd:    jmp     0x1cd6a
0x1cd6a:
0x1cec2:    cmp     ebx, 8
0x1cec5:    je      0x1c7e8
0x1cecb:    add     ebx, 1
0x1cece:    test    byte ptr [rsp + 0x48], 8
0x1ced3:    je      0x1d039
0x1ced9:    mov     byte ptr [r8 - 1], 0x31
0x1cede:    lea     rbp, [r8 - 1]
0x1cee2:    jmp     0x1ca21
0x1ca21:

```

```

0x1cee7:      fadd     dword ptr [rip + 0x563f]
0x1ceed:      jmp      0x1c9ca
0x1c9ca:
0x1cef2:      fadd     dword ptr [rip + 0x5634]
0x1cef8:      fxch     st(1)
0x1cefa:      jmp      0x1ccc1
0x1ccc1:
0x1ceff:      nop
0x1cf00:      mov      byte ptr [rsi], 0x6b
0x1cf03:      test     eax, eax
0x1cf05:      jne      0x1cc5a
0x1cf0b:      mov      qword ptr [rsp + 8], rdx
0x1cf10:      jmp      0x1ca4f
0x1ca4f:
0x1cf15:      nop      dword ptr [rax]
0x1cf18:      cmp      r9, 9
0x1cf1c:      ja       0x1c7af
0x1cf22:      cmp      dword ptr [rsp + 0x10], 1
0x1cf27:      je       0x1cfa0
0x1cf29:      mov      r12d, dword ptr [rsp + 0x10]
0x1cf2e:      test     r12d, r12d
0x1cf31:      jne      0x1cfac
0x1cf33:      test     esi, esi
0x1cf35:      je       0x1cfac
0x1cf37:      lea      edx, [rax + 1]
0x1cf3a:      cmp      eax, 9
0x1cf3d:      je       0x1d024
0x1cf43:      lea      eax, [rdx + 0x30]
0x1cf46:      mov      rdi, qword ptr [rsp]
0x1cf4a:      lea      r8, [rdi + 0x287]
0x1cf51:      mov      byte ptr [rdi + 0x287], al
0x1cf57:      mov      eax, ebp
0x1cf59:      sub      r8, rbp
0x1cf5c:      cmp      ebp, 8
0x1cf5f:      jae      0x1cfd1
0x1cf61:      test     bpl, 4
0x1cf65:      jne      0x1d0d6
0x1cf6b:      test     eax, eax
0x1cf6d:      je       0x1cf7a
0x1cf6f:      movzx    edx, byte ptr [r15]
0x1cf73:      mov      byte ptr [r8], dl
0x1cf76:      test     al, 2
0x1cf78:      jne      0x1cf8e
0x1cf7a:      xor      esi, esi
0x1cf7c:      xor      edi, edi
0x1cf7e:      cmp      dword ptr [rsp + 0x10], 1
0x1cf83:      jne      0x1c7bf
0x1cf89:      jmp      0x1c7e8

```

```

0x1c7e8
0x1cf8e:    movzx    edx, word ptr [r15 + rax - 2]
0x1cf94:    mov      word ptr [r8 + rax - 2], dx
0x1cf9a:    jmp      0x1cf7a
0x1cf7a
0x1cf9c:    nop      dword ptr [rax]
0x1cfa0:    mov      edx, eax
0x1cfa2:    and      edx, 1
0x1cfa5:    add      edx, esi
0x1cfa7:    cmp      edx, 2
0x1cfaa:    jg        0x1cf37
0x1cfac:    test     eax, eax
0x1cfae:    jne      0x1d117
0x1cfb4:    mov      r8, qword ptr [rsp + 8]
0x1cfb9:    test     byte ptr [rsp + 0x48], 8
0x1cfbe:    jne      0x1cf7c
0x1cfc0:    mov      eax, 0x30
0x1cfc5:    jmp      0x1cf46
0x1cf46
0x1cfca:    xor      ebx, ebx
0x1cfcc:    jmp      0x1ca3c
0x1ca3c
0x1cfd1:    mov      rax, qword ptr [r15]
0x1cfd4:    lea      rsi, [r8 + 8]
0x1cfd8:    mov      rdi, r15
0x1cfdb:    and      rsi, 0xfffffffffffffff8
0x1cfd9:    mov      qword ptr [r8], rax
0x1cfe2:    mov      eax, ebp
0x1cfe4:    mov      rdx, qword ptr [r15 + rax - 8]
0x1cfe9:    mov      qword ptr [r8 + rax - 8], rdx
0x1cfec:    mov      rax, r8
0x1cff1:    sub      rax, rsi
0x1cff4:    sub      rdi, rax
0x1cff7:    add      eax, ebp
0x1cff9:    and      eax, 0xffffffff8
0x1cffc:    cmp      eax, 8
0x1cffc:    jb        0x1cf7a
0x1d005:    and      eax, 0xffffffff8
0x1d008:    mov      r9d, eax
0x1d00b:    xor      eax, eax
0x1d00d:    mov      edx, eax
0x1d00f:    add      eax, 8
0x1d012:    mov      r11, qword ptr [rdi + rdx]
0x1d016:    mov      qword ptr [rsi + rdx], r11
0x1d01a:    cmp      eax, r9d
0x1d01d:    jb        0x1d00d
0x1d01f:    jmp      0x1cf7a
0x1cf7a

```

```

0x1d024:    lea     rcx, [r9 + 1]
0x1d028:    cmp     r9, 9
0x1d02c:    je      0x1d0cc
0x1d032:    xor     esi, esi
0x1d034:    jmp     0x1cfb4
0x1cfb4
0x1d039:    mov     rax, rbp
0x1d03c:    mov     byte ptr [r8 - 1], 0x30
0x1d041:    not     rax
0x1d044:    add     r8, rax
0x1d047:    mov     eax, ebp
0x1d049:    cmp     ebp, 8
0x1d04c:    jae     0x1d080
0x1d04e:    and     ebp, 4
0x1d051:    jne     0x1d0f0
0x1d057:    test    eax, eax
0x1d059:    je      0x1ced9
0x1d05f:    movzx   edx, byte ptr [r15]
0x1d063:    mov     byte ptr [r8], dl
0x1d066:    test    al, 2
0x1d068:    je      0x1ced9
0x1d06e:    movzx   edx, word ptr [r15 + rax - 2]
0x1d074:    mov     word ptr [r8 + rax - 2], dx
0x1d07a:    jmp     0x1ced9
0x1ced9
0x1d07f:    nop
0x1d080:    mov     rax, qword ptr [r15]
0x1d083:    lea     rcx, [r8 + 8]
0x1d087:    and     rcx, 0xffffffffffff8
0x1d08b:    mov     qword ptr [r8], rax
0x1d08e:    mov     eax, ebp
0x1d090:    mov     rdx, qword ptr [r15 + rax - 8]
0x1d095:    mov     qword ptr [r8 + rax - 8], rdx
0x1d09a:    mov     rax, r8
0x1d09d:    sub     rax, rcx
0x1d0a0:    sub     r15, rax
0x1d0a3:    add     eax, ebp
0x1d0a5:    and     eax, 0xffffffff8
0x1d0a8:    cmp     eax, 8
0x1d0ab:    jb      0x1ced9
0x1d0b1:    and     eax, 0xffffffff8
0x1d0b4:    xor     edx, edx
0x1d0b6:    mov     esi, edx
0x1d0b8:    add     edx, 8
0x1d0bb:    mov     rdi, qword ptr [r15 + rsi]
0x1d0bf:    mov     qword ptr [rcx + rsi], rdi
0x1d0c3:    cmp     edx, eax
0x1d0c5:    jb      0x1d0b6

```

```

0x1d0c7:      jmp      0x1ced9
0x1ced9
0x1d0cc:      mov      r8, qword ptr [rsp + 8]
0x1d0d1:      jmp      0x1cf7a
0x1cf7a
0x1d0d6:      mov      edx, dword ptr [r15]
0x1d0d9:      mov      dword ptr [r8], edx
0x1d0dc:      mov      edx, dword ptr [r15 + rax - 4]
0x1d0e1:      mov      dword ptr [r8 + rax - 4], edx
0x1d0e6:      jmp      0x1cf7a
0x1cf7a
0x1d0eb:      call     0x49d0
0x1d0f0:      mov      edx, dword ptr [r15]
0x1d0f3:      mov      dword ptr [r8], edx
0x1d0f6:      mov      edx, dword ptr [r15 + rax - 4]
0x1d0fb:      mov      dword ptr [r8 + rax - 4], edx
0x1d100:      jmp      0x1ced9
0x1ced9
0x1d105:      mov      rdx, qword ptr [rsp + 8]
0x1d10a:      test     eax, eax
0x1d10c:      jne      0x1cc5a
0x1d112:      jmp      0x1ca4f
0x1ca4f
0x1d117:      add      eax, 0x30
0x1d11a:      jmp      0x1cf46
0x1cf46
0x1d11f:      nop
0x1d120:      endbr64
0x1d124:      push     r14
0x1d126:      push     r13
0x1d128:      mov      r13, rsi
0x1d12b:      push     r12
0x1d12d:      push     rbp
0x1d12e:      mov      rbp, rdx
0x1d131:      push     rbx
0x1d132:      mov      rbx, rdi
0x1d135:      sub      rsp, 0x10
0x1d139:      mov      rax, qword ptr fs:[0x28]
0x1d142:      mov      qword ptr [rsp + 8], rax
0x1d147:      xor      eax, eax
0x1d149:      test     rdi, rdi
0x1d14c:      je       0x1d270
0x1d152:      xor      r12d, r12d
0x1d155:      cmp      byte ptr [rbx], 0x27
0x1d158:      jne      0x1d164
0x1d15a:      add      rbx, 1
0x1d15e:      mov      r12d, 4
0x1d164:      lea      r14, [rip + 0x53ad]

```

```

0x1d16b:    mov     ecx, 4
0x1d170:    lea     rsi, [rip + 0xb819]
0x1d177:    mov     rdi, rbx
0x1d17a:    mov     rdx, r14
0x1d17d:    call    0x1dc80
0x1d182:    test    eax, eax
0x1d184:    js      0x1d1c0
0x1d186:    cdqe
0x1d188:    mov     qword ptr [rbp], 1
0x1d190:    or      r12d, dword ptr [r14 + rax*4]
0x1d194:    xor     eax, eax
0x1d196:    mov     dword ptr [r13], r12d
0x1d19a:    mov     rdx, qword ptr [rsp + 8]
0x1d19f:    sub     rdx, qword ptr fs:[0x28]
0x1d1a8:    jne     0x1d2fd
0x1d1ae:    add     rsp, 0x10
0x1d1b2:    pop     rbx
0x1d1b3:    pop     rbp
0x1d1b4:    pop     r12
0x1d1b6:    pop     r13
0x1d1b8:    pop     r14
0x1d1ba:    ret
0x1d1bb:    nop     dword ptr [rax + rax]
0x1d1c0:    xor     edx, edx
0x1d1c2:    mov     rsi, rsp
0x1d1c5:    lea     r8, [rip + 0x5329]
0x1d1cc:    mov     rcx, rbp
0x1d1cf:    mov     rdi, rbx
0x1d1d2:    call    0x1d7e0
0x1d1d7:    test    eax, eax
0x1d1d9:    jne     0x1d228
0x1d1db:    movzx    ecx, byte ptr [rbx]
0x1d1de:    lea     edx, [rcx - 0x30]
0x1d1e1:    mov     rcx, qword ptr [rsp]
0x1d1e5:    cmp     dl, 9
0x1d1e8:    ja      0x1d200
0x1d1ea:    jmp     0x1d217
0x1d217:
0x1d1ec:    nop     dword ptr [rax]
0x1d1f0:    movzx    edi, byte ptr [rbx + 1]
0x1d1f4:    add     rbx, 1
0x1d1f8:    lea     edx, [rdi - 0x30]
0x1d1fb:    cmp     dl, 9
0x1d1fe:    jbe     0x1d217
0x1d200:    cmp     rcx, rbx
0x1d203:    jne     0x1d1f0
0x1d205:    cmp     byte ptr [rcx - 1], 0x42
0x1d209:    je      0x1d2d0

```



```

0x1d20f:      or      r12b, 0x80
0x1d213:      or      r12d, 0x20
0x1d217:      mov     rdx, qword ptr [rbp]
0x1d21b:      mov     dword ptr [r13], r12d
0x1d21f:      jmp     0x1d234
0x1d234:
0x1d221:      nop     dword ptr [rax]
0x1d228:      mov     dword ptr [r13], 0
0x1d230:      mov     rdx, qword ptr [rbp]
0x1d234:      test    rdx, rdx
0x1d237:      jne     0x1d19a
0x1d23d:      lea     rdi, [rip + 0x52a1]
0x1d244:      call    0x47e0
0x1d249:      cmp     rax, 1
0x1d24d:      sbb     rax, rax
0x1d250:      and     eax, 0x200
0x1d255:      add     rax, 0x200
0x1d25b:      mov     qword ptr [rbp], rax
0x1d25f:      mov     eax, 4
0x1d264:      jmp     0x1d19a
0x1d19a:
0x1d269:      nop     dword ptr [rax]
0x1d270:      lea     rdi, [rip + 0x5259]
0x1d277:      call    0x47e0
0x1d27c:      mov     rbx, rax
0x1d27f:      test    rax, rax
0x1d282:      jne     0x1d152
0x1d288:      lea     rdi, [rip + 0x524c]
0x1d28f:      call    0x47e0
0x1d294:      mov     rbx, rax
0x1d297:      test    rax, rax
0x1d29a:      jne     0x1d152
0x1d2a0:      lea     rdi, [rip + 0x523e]
0x1d2a7:      call    0x47e0
0x1d2ac:      test    rax, rax
0x1d2af:      je      0x1d2e6
0x1d2b1:      mov     qword ptr [rbp], 0x200
0x1d2b9:      xor     eax, eax
0x1d2bb:      mov     dword ptr [r13], 0
0x1d2c3:      jmp     0x1d19a
0x1d19a:
0x1d2c8:      nop     dword ptr [rax + rax]
0x1d2d0:      or      r12d, 0x180
0x1d2d7:      cmp     byte ptr [rcx - 2], 0x69
0x1d2db:      jne     0x1d217
0x1d2e1:      jmp     0x1d213
0x1d213:
0x1d2e6:      mov     qword ptr [rbp], 0x400

```

```

0x1d2ee:    xor     eax, eax
0x1d2f0:    mov     dword ptr [r13], 0
0x1d2f8:    jmp     0x1d19a
0x1d19a:
0x1d2fd:    call    0x49d0
0x1d302:    nop     word ptr cs:[rax + rax]
0x1d30c:    nop     dword ptr [rax]
0x1d310:    endbr64
0x1d314:    push    r12
0x1d316:    push    rbp
0x1d317:    mov     ebp, edi
0x1d319:    push    rbx
0x1d31a:    mov     rbx, qword ptr [rip + 0xe81f]
0x1d321:    test    rbx, rbx
0x1d324:    jne     0x1d339
0x1d326:    jmp     0x1d350
0x1d350:
0x1d328:    nop     dword ptr [rax + rax]
0x1d330:    mov     rbx, qword ptr [rbx + 8]
0x1d334:    test    rbx, rbx
0x1d337:    je      0x1d350
0x1d339:    cmp     dword ptr [rbx], ebp
0x1d33b:    jne     0x1d330
0x1d33d:    xor     eax, eax
0x1d33f:    cmp     byte ptr [rbx + 0x10], 0
0x1d343:    je      0x1d349
0x1d345:    lea     rax, [rbx + 0x10]
0x1d349:    pop     rbx
0x1d34a:    pop     rbp
0x1d34b:    pop     r12
0x1d34d:    ret
0x1d34e:    nop
0x1d350:    mov     edi, ebp
0x1d352:    lea     r12, [rip + 0x508c]
0x1d359:    call    0x4970
0x1d35e:    mov     edi, 0x11
0x1d363:    test    rax, rax
0x1d366:    je      0x1d377
0x1d368:    mov     r12, qword ptr [rax]
0x1d36b:    mov     rdi, r12
0x1d36e:    call    0x49b0
0x1d373:    lea     rdi, [rax + 0x11]
0x1d377:    call    0x1d670
0x1d37c:    mov     rsi, r12
0x1d37f:    mov     dword ptr [rax], ebp
0x1d381:    lea     rdi, [rax + 0x10]
0x1d385:    mov     rbx, rax
0x1d388:    call    0x48b0

```

```

0x1d38d:    mov     rax, qword ptr [rip + 0xe7ac]
0x1d394:    mov     qword ptr [rip + 0xe7a5], rbx
0x1d39b:    mov     qword ptr [rbx + 8], rax
0x1d39f:    jmp     0x1d33d
0x1d33d:
0x1d3a1:    nop     word ptr cs:[rax + rax]
0x1d3ac:    nop     dword ptr [rax]
0x1d3b0:    endbr64
0x1d3b4:    push    r12
0x1d3b6:    push    rbp
0x1d3b7:    mov     rbp, rdi
0x1d3ba:    push    rbx
0x1d3bb:    mov     rbx, qword ptr [rip + 0xe77e]
0x1d3c2:    test    rbx, rbx
0x1d3c5:    je      0x1d400
0x1d3c7:    movzx   r12d, byte ptr [rdi]
0x1d3cb:    jmp     0x1d3d9
0x1d3d9:
0x1d3cd:    nop     dword ptr [rax]
0x1d3d0:    mov     rbx, qword ptr [rbx + 8]
0x1d3d4:    test    rbx, rbx
0x1d3d7:    je      0x1d400
0x1d3d9:    cmp     byte ptr [rbx + 0x10], r12b
0x1d3dd:    jne     0x1d3d0
0x1d3df:    lea     rdi, [rbx + 0x10]
0x1d3e3:    mov     rsi, rbp
0x1d3e6:    call    0x4b30
0x1d3eb:    test    eax, eax
0x1d3ed:    jne     0x1d3d0
0x1d3ef:    mov     r12, rbx
0x1d3f2:    pop     rbx
0x1d3f3:    pop     rbp
0x1d3f4:    mov     rax, r12
0x1d3f7:    pop     r12
0x1d3f9:    ret
0x1d3fa:    nop     word ptr [rax + rax]
0x1d400:    mov     rbx, qword ptr [rip + 0xe731]
0x1d407:    test    rbx, rbx
0x1d40a:    je      0x1d448
0x1d40c:    movzx   r12d, byte ptr [rbp]
0x1d411:    jmp     0x1d421
0x1d421:
0x1d413:    nop     dword ptr [rax + rax]
0x1d418:    mov     rbx, qword ptr [rbx + 8]
0x1d41c:    test    rbx, rbx
0x1d41f:    je      0x1d448
0x1d421:    cmp     byte ptr [rbx + 0x10], r12b
0x1d425:    jne     0x1d418

```

```

0x1d427:    lea     rdi, [rbx + 0x10]
0x1d42b:    mov     rsi, rbp
0x1d42e:    call    0x4b30
0x1d433:    test    eax, eax
0x1d435:    jne     0x1d418
0x1d437:    xor     r12d, r12d
0x1d43a:    pop     rbx
0x1d43b:    pop     rbp
0x1d43c:    mov     rax, r12
0x1d43f:    pop     r12
0x1d441:    ret
0x1d442:    nop     word ptr [rax + rax]
0x1d448:    mov     rdi, rbp
0x1d44b:    call    0x4b60
0x1d450:    mov     rdi, rbp
0x1d453:    mov     r12, rax
0x1d456:    call    0x49b0
0x1d45b:    lea     rdi, [rax + 0x11]
0x1d45f:    call    0x1d670
0x1d464:    mov     rsi, rbp
0x1d467:    lea     rdi, [rax + 0x10]
0x1d46b:    mov     rbx, rax
0x1d46e:    call    0x48b0
0x1d473:    test    r12, r12
0x1d476:    je      0x1d4a0
0x1d478:    mov     eax, dword ptr [r12 + 0x10]
0x1d47d:    mov     dword ptr [rbx], eax
0x1d47f:    mov     rax, qword ptr [rip + 0xe6ba]
0x1d486:    mov     qword ptr [rip + 0xe6b3], rbx
0x1d48d:    mov     qword ptr [rbx + 8], rax
0x1d491:    jmp     0x1d3ef
0x1d3ef:
0x1d496:    nop     word ptr cs:[rax + rax]
0x1d4a0:    mov     rax, qword ptr [rip + 0xe691]
0x1d4a7:    mov     qword ptr [rip + 0xe68a], rbx
0x1d4ae:    mov     qword ptr [rbx + 8], rax
0x1d4b2:    mov     rax, r12
0x1d4b5:    pop     rbx
0x1d4b6:    pop     rbp
0x1d4b7:    pop     r12
0x1d4b9:    ret
0x1d4ba:    nop     word ptr [rax + rax]
0x1d4c0:    endbr64
0x1d4c4:    push    r12
0x1d4c6:    push    rbp
0x1d4c7:    mov     ebp, edi
0x1d4c9:    push    rbx
0x1d4ca:    mov     rbx, qword ptr [rip + 0xe65f]

```

```

0x1d4d1:      test    rbx, rbx
0x1d4d4:      jne     0x1d4e9
0x1d4d6:      jmp     0x1d500
0x1d500:
0x1d4d8:      nop     dword ptr [rax + rax]
0x1d4e0:      mov     rbx, qword ptr [rbx + 8]
0x1d4e4:      test    rbx, rbx
0x1d4e7:      je      0x1d500
0x1d4e9:      cmp     dword ptr [rbx], ebp
0x1d4eb:      jne     0x1d4e0
0x1d4ed:      xor     eax, eax
0x1d4ef:      cmp     byte ptr [rbx + 0x10], 0
0x1d4f3:      je      0x1d4f9
0x1d4f5:      lea     rax, [rbx + 0x10]
0x1d4f9:      pop     rbx
0x1d4fa:      pop     rbp
0x1d4fb:      pop     r12
0x1d4fd:      ret
0x1d4fe:      nop
0x1d500:      mov     edi, ebp
0x1d502:      lea     r12, [rip + 0x4edc]
0x1d509:      call    0x4a10
0x1d50e:      mov     edi, 0x11
0x1d513:      test    rax, rax
0x1d516:      je      0x1d527
0x1d518:      mov     r12, qword ptr [rax]
0x1d51b:      mov     rdi, r12
0x1d51e:      call    0x49b0
0x1d523:      lea     rdi, [rax + 0x11]
0x1d527:      call    0x1d670
0x1d52c:      mov     rsi, r12
0x1d52f:      mov     dword ptr [rax], ebp
0x1d531:      lea     rdi, [rax + 0x10]
0x1d535:      mov     rbx, rax
0x1d538:      call    0x48b0
0x1d53d:      mov     rax, qword ptr [rip + 0xe5ec]
0x1d544:      mov     qword ptr [rip + 0xe5e5], rbx
0x1d54b:      mov     qword ptr [rbx + 8], rax
0x1d54f:      jmp     0x1d4ed
0x1d4ed:
0x1d551:      nop     word ptr cs:[rax + rax]
0x1d55c:      nop     dword ptr [rax]
0x1d560:      endbr64
0x1d564:      push    r12
0x1d566:      push    rbp
0x1d567:      mov     rbp, rdi
0x1d56a:      push    rbx
0x1d56b:      mov     rbx, qword ptr [rip + 0xe5be]

```

```

0x1d572:      test    rbx, rbx
0x1d575:      je      0x1d5b0
0x1d577:      movzx   r12d, byte ptr [rdi]
0x1d57b:      jmp     0x1d589
0x1d589:
0x1d57d:      nop     dword ptr [rax]
0x1d580:      mov     rbx, qword ptr [rbx + 8]
0x1d584:      test    rbx, rbx
0x1d587:      je      0x1d5b0
0x1d589:      cmp     byte ptr [rbx + 0x10], r12b
0x1d58d:      jne     0x1d580
0x1d58f:      lea     rdi, [rbx + 0x10]
0x1d593:      mov     rsi, rbp
0x1d596:      call    0x4b30
0x1d59b:      test    eax, eax
0x1d59d:      jne     0x1d580
0x1d59f:      mov     r12, rbx
0x1d5a2:      pop     rbx
0x1d5a3:      pop     rbp
0x1d5a4:      mov     rax, r12
0x1d5a7:      pop     r12
0x1d5a9:      ret
0x1d5aa:      nop     word ptr [rax + rax]
0x1d5b0:      mov     rbx, qword ptr [rip + 0xe571]
0x1d5b7:      test    rbx, rbx
0x1d5ba:      je      0x1d5f8
0x1d5bc:      movzx   r12d, byte ptr [rbp]
0x1d5c1:      jmp     0x1d5d1
0x1d5d1:
0x1d5c3:      nop     dword ptr [rax + rax]
0x1d5c8:      mov     rbx, qword ptr [rbx + 8]
0x1d5cc:      test    rbx, rbx
0x1d5cf:      je      0x1d5f8
0x1d5d1:      cmp     byte ptr [rbx + 0x10], r12b
0x1d5d5:      jne     0x1d5c8
0x1d5d7:      lea     rdi, [rbx + 0x10]
0x1d5db:      mov     rsi, rbp
0x1d5de:      call    0x4b30
0x1d5e3:      test    eax, eax
0x1d5e5:      jne     0x1d5c8
0x1d5e7:      xor     r12d, r12d
0x1d5ea:      pop     rbx
0x1d5eb:      pop     rbp
0x1d5ec:      mov     rax, r12
0x1d5ef:      pop     r12
0x1d5f1:      ret
0x1d5f2:      nop     word ptr [rax + rax]
0x1d5f8:      mov     rdi, rbp

```

```

0x1d5fb:      call    0x4bc0
0x1d600:      mov     rdi, rbp
0x1d603:      mov     r12, rax
0x1d606:      call    0x49b0
0x1d60b:      lea     rdi, [rax + 0x11]
0x1d60f:      call    0x1d670
0x1d614:      mov     rsi, rbp
0x1d617:      lea     rdi, [rax + 0x10]
0x1d61b:      mov     rbx, rax
0x1d61e:      call    0x48b0
0x1d623:      test    r12, r12
0x1d626:      je      0x1d650
0x1d628:      mov     eax, dword ptr [r12 + 0x10]
0x1d62d:      mov     dword ptr [rbx], eax
0x1d62f:      mov     rax, qword ptr [rip + 0xe4fa]
0x1d636:      mov     qword ptr [rip + 0xe4f3], rbx
0x1d63d:      mov     qword ptr [rbx + 8], rax
0x1d641:      jmp     0x1d59f
0x1d59f:
0x1d646:      nop     word ptr cs:[rax + rax]
0x1d650:      mov     rax, qword ptr [rip + 0xe4d1]
0x1d657:      mov     qword ptr [rip + 0xe4ca], rbx
0x1d65e:      mov     qword ptr [rbx + 8], rax
0x1d662:      mov     rax, r12
0x1d665:      pop     rbx
0x1d666:      pop     rbp
0x1d667:      pop     r12
0x1d669:      ret
0x1d66a:      nop     word ptr [rax + rax]
0x1d670:      endbr64
0x1d674:      push    rbx
0x1d675:      mov     rbx, rdi
0x1d678:      call    0x4c20
0x1d67d:      test    rax, rax
0x1d680:      jne     0x1d687
0x1d682:      test    rbx, rbx
0x1d685:      jne     0x1d689
0x1d687:      pop     rbx
0x1d688:      ret
0x1d689:      call    0x18cf0
0x1d68e:      nop
0x1d690:      endbr64
0x1d694:      push    rbx
0x1d695:      mov     rbx, rsi
0x1d698:      call    0x4ce0
0x1d69d:      test    rax, rax
0x1d6a0:      jne     0x1d6a7
0x1d6a2:      test    rbx, rbx

```

```

0x1d6a5:    jne     0x1d6a9
0x1d6a7:    pop     rbx
0x1d6a8:    ret
0x1d6a9:    call    0x18cf0
0x1d6ae:    nop
0x1d6b0:    endbr64
0x1d6b4:    push    rbx
0x1d6b5:    mov     rax, rsi
0x1d6b8:    mov     rsi, qword ptr [rsi]
0x1d6bb:    test    rdi, rdi
0x1d6be:    je      0x1d6f0
0x1d6c0:    movabs  rdx, 0xaaaaaaaaaaaaa9
0x1d6ca:    cmp     rsi, rdx
0x1d6cd:    ja      0x1d701
0x1d6cf:    lea     rdx, [rsi + 1]
0x1d6d3:    shr     rdx, 1
0x1d6d6:    add     rsi, rdx
0x1d6d9:    setne   bl
0x1d6dc:    mov     qword ptr [rax], rsi
0x1d6df:    call    0x4ce0
0x1d6e4:    test    rax, rax
0x1d6e7:    jne     0x1d6ed
0x1d6e9:    test    bl, bl
0x1d6eb:    jne     0x1d701
0x1d6ed:    pop     rbx
0x1d6ee:    ret
0x1d6ef:    nop
0x1d6f0:    mov     ebx, 1
0x1d6f5:    test    rsi, rsi
0x1d6f8:    jne     0x1d6dc
0x1d6fa:    mov     esi, 0x40
0x1d6ff:    jmp     0x1d6dc
0x1d6dc:
0x1d701:    call    0x18cf0
0x1d706:    nop     word ptr cs:[rax + rax]
0x1d710:    endbr64
0x1d714:    push    rbx
0x1d715:    mov     esi, 1
0x1d71a:    mov     rbx, rdi
0x1d71d:    call    0x4b10
0x1d722:    test    rax, rax
0x1d725:    jne     0x1d72c
0x1d727:    test    rbx, rbx
0x1d72a:    jne     0x1d72e
0x1d72c:    pop     rbx
0x1d72d:    ret
0x1d72e:    call    0x18cf0
0x1d733:    nop     word ptr cs:[rax + rax]

```



```

0x1d73e:      nop
0x1d740:      endbr64
0x1d744:      sub     rsp, 8
0x1d748:      call    0x4b10
0x1d74d:      test    rax, rax
0x1d750:      je      0x1d757
0x1d752:      add     rsp, 8
0x1d756:      ret
0x1d757:      call    0x18cf0
0x1d75c:      nop     dword ptr [rax]
0x1d760:      endbr64
0x1d764:      push    r12
0x1d766:      mov     r12, rsi
0x1d769:      push    rbp
0x1d76a:      mov     rbp, rdi
0x1d76d:      mov     rdi, rsi
0x1d770:      sub     rsp, 8
0x1d774:      call    0x4c20
0x1d779:      mov     rdi, rax
0x1d77c:      test    rax, rax
0x1d77f:      jne     0x1d786
0x1d781:      test    r12, r12
0x1d784:      jne     0x1d798
0x1d786:      add     rsp, 8
0x1d78a:      mov     rdx, r12
0x1d78d:      mov     rsi, rbp
0x1d790:      pop     rbp
0x1d791:      pop     r12
0x1d793:      jmp     0x4bb0
0x4bb0
0x1d798:      call    0x18cf0
0x1d79d:      nop     dword ptr [rax]
0x1d7a0:      endbr64
0x1d7a4:      push    r12
0x1d7a6:      push    rbp
0x1d7a7:      mov     rbp, rdi
0x1d7aa:      sub     rsp, 8
0x1d7ae:      call    0x49b0
0x1d7b3:      lea     r12, [rax + 1]
0x1d7b7:      mov     rdi, r12
0x1d7ba:      call    0x4c20
0x1d7bf:      test    rax, rax
0x1d7c2:      je      0x1d7d9
0x1d7c4:      add     rsp, 8
0x1d7c8:      mov     rdx, r12
0x1d7cb:      mov     rsi, rbp
0x1d7ce:      mov     rdi, rax
0x1d7d1:      pop     rbp

```

```

0x1d7d2:    pop     r12
0x1d7d4:    jmp     0x4bb0
0x4bb0:
0x1d7d9:    call    0x18cf0
0x1d7de:    nop
0x1d7e0:    endbr64
0x1d7e4:    push    r15
0x1d7e6:    mov     r15, rsi
0x1d7e9:    push    r14
0x1d7eb:    mov     r14, rcx
0x1d7ee:    push    r13
0x1d7f0:    mov     r13d, edx
0x1d7f3:    push    r12
0x1d7f5:    mov     r12, r8
0x1d7f8:    push    rbp
0x1d7f9:    mov     rbp, rdi
0x1d7fc:    push    rbx
0x1d7fd:    sub     rsp, 0x28
0x1d801:    movzx   ebx, byte ptr [rdi]
0x1d804:    mov     rax, qword ptr fs:[0x28]
0x1d80d:    mov     qword ptr [rsp + 0x18], rax
0x1d812:    xor     eax, eax
0x1d814:    lea     rax, [rsp + 0x10]
0x1d819:    test    rsi, rsi
0x1d81c:    cmovbe  r15, rax
0x1d820:    call    0x4f00
0x1d825:    mov     rcx, qword ptr [rax]
0x1d828:    mov     rax, rbp
0x1d82b:    jmp     0x1d838
0x1d838:
0x1d82d:    nop     dword ptr [rax]
0x1d830:    movzx   ebx, byte ptr [rax + 1]
0x1d834:    add     rax, 1
0x1d838:    movzx   edx, bl
0x1d83b:    test    byte ptr [rcx + rdx*2 + 1], 0x20
0x1d840:    jne     0x1d830
0x1d842:    cmp     bl, 0x2d
0x1d845:    je      0x1d8ca
0x1d84b:    call    0x4860
0x1d850:    mov     edx, r13d
0x1d853:    mov     rsi, r15
0x1d856:    mov     rdi, rbp
0x1d859:    mov     dword ptr [rax], 0
0x1d85f:    mov     qword ptr [rsp + 8], rax
0x1d864:    call    0x4dd0
0x1d869:    mov     r13, qword ptr [r15]
0x1d86c:    mov     rcx, qword ptr [rsp + 8]
0x1d871:    mov     rbx, rax

```

```

0x1d874:      cmp     r13, rbp
0x1d877:      je      0x1d8d8
0x1d879:      mov     eax, dword ptr [rcx]
0x1d87b:      test    eax, eax
0x1d87d:      jne     0x1d8c0
0x1d87f:      xor     ebp, ebp
0x1d881:      test    r12, r12
0x1d884:      je      0x1d893
0x1d886:      movzx    edx, byte ptr [r13]
0x1d88b:      test    dl, dl
0x1d88d:      jne     0x1d958
0x1d893:      mov     qword ptr [r14], rbx
0x1d896:      mov     rax, qword ptr [rsp + 0x18]
0x1d89b:      sub     rax, qword ptr fs:[0x28]
0x1d8a4:      jne     0x1dbf9
0x1d8aa:      add     rsp, 0x28
0x1d8ae:      mov     eax, ebp
0x1d8b0:      pop     rbx
0x1d8b1:      pop     rbp
0x1d8b2:      pop     r12
0x1d8b4:      pop     r13
0x1d8b6:      pop     r14
0x1d8b8:      pop     r15
0x1d8ba:      ret
0x1d8bb:      nop     dword ptr [rax + rax]
0x1d8c0:      mov     ebp, 1
0x1d8c5:      cmp     eax, 0x22
0x1d8c8:      je      0x1d881
0x1d8ca:      mov     ebp, 4
0x1d8cf:      jmp     0x1d896
0x1d896:
0x1d8d1:      nop     dword ptr [rax]
0x1d8d8:      test    r12, r12
0x1d8db:      je      0x1d8ca
0x1d8dd:      movzx    edx, byte ptr [rbp]
0x1d8e1:      test    dl, dl
0x1d8e3:      je      0x1d8ca
0x1d8e5:      movsx    esi, dl
0x1d8e8:      mov     rdi, r12
0x1d8eb:      mov     byte ptr [rsp + 8], dl
0x1d8ef:      xor     ebp, ebp
0x1d8f1:      call    0x4a00
0x1d8f6:      mov     ebx, 1
0x1d8fb:      movzx    edx, byte ptr [rsp + 8]
0x1d900:      test    rax, rax
0x1d903:      je      0x1d8ca
0x1d905:      mov     esi, 0x30
0x1d90a:      mov     rdi, r12

```

```

0x1d90d:    mov     byte ptr [rsp + 8], dl
0x1d911:    call    0x4a00
0x1d916:    movzx   edx, byte ptr [rsp + 8]
0x1d91b:    test    rax, rax
0x1d91e:    je      0x1d97c
0x1d920:    movzx   eax, byte ptr [r13 + 1]
0x1d925:    cmp     al, 0x44
0x1d927:    je      0x1d9a2
0x1d929:    cmp     al, 0x69
0x1d92b:    je      0x1dbe0
0x1d931:    cmp     al, 0x42
0x1d933:    je      0x1d9a2
0x1d935:    sub     edx, 0x42
0x1d938:    cmp     dl, 0x35
0x1d93b:    ja      0x1d971
0x1d93d:    lea     rcx, [rip + 0x4c0c]
0x1d944:    movzx   edx, dl
0x1d947:    movsxd  rax, dword ptr [rcx + rdx*4]
0x1d94b:    add     rax, rcx
0x1d94e:    jmp     rax
rax
->->->                                indirect JMP
0x1bde0:    cmp     ebx, dword ptr [rsp + 4]
0x1bde4:    jge     0x1bff0
0x1bdea:    lea     eax, [rbx + 1]
0x1bded:    mov     byte ptr [r15 + rbx], 0x5c
0x1bdf2:    movsxd  rdx, eax
0x1bdf5:    mov     dword ptr [rsp + 8], 1
0x1bdfd:    movsxd  rbx, eax
0x1be00:    mov     byte ptr [r15 + rdx], 0
0x1be05:    jmp     0x1b74d
0x1b74d
0x1be0a:    nop     word ptr [rax + rax]
0x1be10:    add     dword ptr [rsp + 4], 0x40
0x1be15:    movsxd  rsi, dword ptr [rsp + 4]
0x1be1a:    mov     rdi, r15
0x1be1d:    call    0x1d690
0x1be22:    mov     r15, rax
0x1be25:    jmp     0x1b778
0x1b778
0x1be2a:    nop     word ptr [rax + rax]
0x1be30:    xor     edx, edx
0x1be32:    test    r13d, r13d
0x1be35:    je      0x1be4d
0x1be37:    movzx   eax, byte ptr [rbp + r14 - 1]
0x1be3d:    cmp     al, 0x27
0x1be3f:    je      0x1c078
0x1be45:    cmp     al, 0x22

```

```

0x1be47:    je      0x1c078
0x1be4d:    lea     rsi, [rsp + 0x6c]
0x1be52:    mov     rdi, rbp
0x1be55:    xor     eax, eax
0x1be57:    call    0x1ac70
0x1be5c:    mov     r12d, dword ptr [rsp + 0x6c]
0x1be61:    mov     r14, rax
0x1be64:    jmp     0x1b8b3
0x1b8b3:
0x1be69:    nop     dword ptr [rax]
0x1be70:    add     dword ptr [rsp + 0x6c], 2
0x1be75:    mov     rdi, r15
0x1be78:    call    0x49b0
0x1be7d:    mov     r13d, dword ptr [rip + 0xfc88]
0x1be84:    cmp     eax, r13d
0x1be87:    jl      0x1c400
0x1be8d:    mov     qword ptr [rsp + 0x48], rbp
0x1be92:    xor     r14d, r14d
0x1be95:    mov     rbp, r15
0x1be98:    mov     r15d, eax
0x1be9b:    mov     dword ptr [rsp + 0x50], 1
0x1bea3:    jmp     0x1beba
0x1beba:
0x1bea5:    nop     dword ptr [rax]
0x1bea8:    add     r14d, 1
0x1beac:    lea     eax, [r13 + r14]
0x1beb1:    cmp     eax, r15d
0x1beb4:    jg      0x1c050
0x1beba:    mov     rsi, qword ptr [rip + 0xfc57]
0x1bec1:    movsxd  rbx, r14d
0x1bec4:    lea     rdi, [rbp + rbx]
0x1bec9:    movzx   eax, byte ptr [rsi]
0x1becc:    cmp     byte ptr [rdi], al
0x1bece:    jne     0x1bea8
0x1bed0:    movsxd  rdx, r13d
0x1bed3:    call    0x4890
0x1bed8:    test    eax, eax
0x1beda:    jne     0x1bea8
0x1bedc:    mov     ecx, dword ptr [rip + 0xfc26]
0x1bee2:    add     ecx, r15d
0x1bee5:    sub     ecx, r13d
0x1bee8:    lea     edi, [rcx + 1]
0x1beeb:    mov     dword ptr [rsp + 0x50], ecx
0x1beef:    movsxd  rdi, edi
0x1bef2:    call    0x1d670
0x1bef7:    mov     rdx, rbx
0x1befa:    mov     rsi, rbp
0x1befd:    mov     rdi, rax

```

```

0x1bf00:      mov     r13, rax
0x1bf03:      call    0x4880
0x1bf08:      movsxd  rdx, dword ptr [rip + 0xfbf9]
0x1bf0f:      lea     rdi, [r13 + rbx]
0x1bf14:      mov     rsi, qword ptr [rip + 0xfbf5]
0x1bf1b:      call    0x4880
0x1bf20:      movsxd  rsi, dword ptr [rip + 0xfbe5]
0x1bf27:      movsxd  rdi, dword ptr [rip + 0xfbda]
0x1bf2e:      lea     edx, [rsi + r14]
0x1bf32:      add     rdi, rbx
0x1bf35:      add     rsi, rbx
0x1bf38:      sub     r15d, edx
0x1bf3b:      add     rsi, rbp
0x1bf3e:      add     rdi, r13
0x1bf41:      movsxd  rdx, r15d
0x1bf44:      call    0x4880
0x1bf49:      movsxd  rcx, dword ptr [rsp + 0x50]
0x1bf4e:      mov     rdi, rbp
0x1bf51:      mov     byte ptr [r13 + rcx], 0
0x1bf57:      call    0x18d10
0x1bf5c:      mov     edi, dword ptr [rsp + 0x18]
0x1bf60:      test    edi, edi
0x1bf62:      jne     0x1c1a8
0x1bf68:      mov     rbp, qword ptr [rsp + 0x48]
0x1bf6d:      mov     r14d, dword ptr [rsp + 0x6c]
0x1bf72:      mov     r15, r13
0x1bf75:      jmp     0x1baa5
0x1baa5:
0x1bf7a:      nop     word ptr [rax + rax]
0x1bf80:      add     edx, 1
0x1bf83:      mov     esi, 0x24
0x1bf88:      mov     edi, 0x24
0x1bf8d:      xor     eax, eax
0x1bf8f:      mov     dword ptr [rsp + 0x6c], edx
0x1bf93:      mov     rdx, r14
0x1bf96:      call    0x1b420
0x1bf9b:      mov     r13, rax
0x1bf9e:      lea     rax, [rip + 0xfb7b]
0x1bfa5:      cmp     r13, rax
0x1bfa8:      je      0x1bdae
0x1bfae:      test    r13, r13
0x1bfb1:      jne     0x1b93c
0x1bfb7:      jmp     0x1bd54
0x1bd54:
0x1bfbc:      nop     dword ptr [rax]
0x1bfc0:      cmp     r12b, 0x23
0x1bfc4:      jne     0x1be30
0x1bfca:      lea     r12d, [r13 + 2]

```

```

0x1bfce:      mov     r14, r15
0x1bfd1:      mov     dword ptr [rsp + 0x6c], r12d
0x1bfd6:      jmp     0x1b8b3
0x1b8b3:
0x1bfdb:      nop     dword ptr [rax + rax]
0x1bfe0:      mov     qword ptr [rsp + 0x10], rdx
0x1bfe5:      jmp     0x1b806
0x1b806:
0x1bfea:      nop     word ptr [rax + rax]
0x1bff0:      add     dword ptr [rsp + 4], 0x40
0x1bff5:      movsxd  rsi, dword ptr [rsp + 4]
0x1bffa:      mov     rdi, r15
0x1bffd:      call    0x1d690
0x1c002:      mov     r15, rax
0x1c005:      jmp     0x1bdea
0x1bdea:
0x1c00a:      nop     word ptr [rax + rax]
0x1c010:      mov     r13, qword ptr [rip + 0xfad9]
0x1c017:      add     edx, 1
0x1c01a:      mov     dword ptr [rsp + 0x6c], edx
0x1c01e:      test    r13, r13
0x1c021:      je      0x1c080
0x1c023:      mov     rdi, r13
0x1c026:      call    0x49b0
0x1c02b:      lea     rdi, [rax + 1]
0x1c02f:      call    0x1d670
0x1c034:      mov     rsi, r13
0x1c037:      mov     rdi, rax
0x1c03a:      call    0x48b0
0x1c03f:      mov     r13, rax
0x1c042:      jmp     0x1b92c
0x1b92c:
0x1c047:      nop     word ptr [rax + rax]
0x1c050:      cmp     dword ptr [rsp + 0x18], 1
0x1c055:      mov     r15, rbp
0x1c058:      mov     r14d, dword ptr [rsp + 0x6c]
0x1c05d:      mov     rbp, qword ptr [rsp + 0x48]
0x1c062:      jle     0x1c417
0x1c068:      mov     dword ptr [rsp + 0x18], 0
0x1c070:      jmp     0x1baa5
0x1baa5:
0x1c075:      nop     dword ptr [rax]
0x1c078:      movsx   edx, al
0x1c07b:      jmp     0x1be4d
0x1be4d:
0x1c080:      mov     edi, 1
0x1c085:      call    0x1d670
0x1c08a:      mov     byte ptr [rax], 0

```

```

0x1c08d:    mov     r13, rax
0x1c090:    jmp     0x1b92c
0x1b92c:
0x1c095:    nop     dword ptr [rax]
0x1c098:    mov     rdi, qword ptr [rsp + 0x10]
0x1c09d:    call    0x49b0
0x1c0a2:    lea     r12d, [rbx + rax]
0x1c0a6:    mov     eax, dword ptr [rsp + 4]
0x1c0aa:    cmp     r12d, eax
0x1c0ad:    jl      0x1c0e0
0x1c0af:    movsxd  r14, eax
0x1c0b2:    mov     rdi, r15
0x1c0b5:    mov     r15d, eax
0x1c0b8:    sub     r14, -0x80
0x1c0bc:    nop     dword ptr [rax]
0x1c0c0:    mov     rsi, r14
0x1c0c3:    sub     r15d, -0x80
0x1c0c7:    sub     r14, -0x80
0x1c0cb:    call    0x1d690
0x1c0d0:    mov     rdi, rax
0x1c0d3:    cmp     r12d, r15d
0x1c0d6:    jge     0x1c0c0
0x1c0d8:    mov     dword ptr [rsp + 4], r15d
0x1c0dd:    mov     r15, rax
0x1c0e0:    movsxd  rdi, ebx
0x1c0e3:    mov     rsi, qword ptr [rsp + 0x10]
0x1c0e8:    movsxd  rbx, r12d
0x1c0eb:    add     rdi, r15
0x1c0ee:    call    0x48b0
0x1c0f3:    jmp     0x1bb18
0x1bb18:
0x1c0f8:    nop     dword ptr [rax + rax]
0x1c100:    xor     edi, edi
0x1c102:    lea     r8d, [rdx + 1]
0x1c106:    movsxd  rax, r8d
0x1c109:    movzx   eax, byte ptr [rbp + rax]
0x1c10e:    lea     ecx, [rax - 0x30]
0x1c111:    cmp     cl, 9
0x1c114:    jbe     0x1c283
0x1c11a:    cmp     al, 0x24
0x1c11c:    je      0x1c3e5
0x1c122:    mov     dword ptr [rsp + 0x6c], r8d
0x1c127:    test    al, al
0x1c129:    je      0x1c3f6
0x1c12f:    cmp     al, 0x3a
0x1c131:    je      0x1c3f6
0x1c137:    test    edi, edi
0x1c139:    jne     0x1bdb3

```



```

0x1c13f:    xor     esi, esi
0x1c141:    jmp     0x1bd98
0x1bd98:
0x1c146:    nop     word ptr cs:[rax + rax]
0x1c150:    test    esi, esi
0x1c152:    je      0x1bd54
0x1c158:    add     edx, 1
0x1c15b:    xor     edi, edi
0x1c15d:    movsxd  rcx, edx
0x1c160:    sub     eax, 0x30
0x1c163:    lea     edx, [rdi + rdi*4]
0x1c166:    movsx   eax, al
0x1c169:    lea     edi, [rax + rdx*2]
0x1c16c:    movzx   eax, byte ptr [rbp + rcx]
0x1c171:    mov     edx, ecx
0x1c173:    add     rcx, 1
0x1c177:    lea     r8d, [rax - 0x30]
0x1c17b:    cmp     r8b, 9
0x1c17f:    jbe     0x1c160
0x1c181:    cmp     al, 0x5e
0x1c183:    je      0x1c478
0x1c189:    cmp     al, 0x2a
0x1c18b:    je      0x1c250
0x1c191:    cmp     al, 0x2d
0x1c193:    je      0x1c102
0x1c199:    mov     dword ptr [rsp + 0x6c], edx
0x1c19d:    mov     esi, edi
0x1c19f:    jmp     0x1bd98
0x1bd98:
0x1c1a4:    nop     dword ptr [rax]
0x1c1a8:    mov     rdi, r13
0x1c1ab:    mov     rbp, r13
0x1c1ae:    add     r14d, dword ptr [rip + 0xf953]
0x1c1b5:    call    0x49b0
0x1c1ba:    add     dword ptr [rsp + 0x18], 1
0x1c1bf:    mov     r13d, dword ptr [rip + 0xf946]
0x1c1c6:    mov     dword ptr [rsp + 0x50], 0
0x1c1ce:    mov     r15d, eax
0x1c1d1:    jmp     0x1bea8
0x1bea8:
0x1c1d6:    nop     word ptr cs:[rax + rax]
0x1c1e0:    lea     edi, [r12 + 5]
0x1c1e5:    movsxd  rdi, edi
0x1c1e8:    call    0x1d670
0x1c1ed:    mov     r11d, 0x733a
0x1c1f3:    mov     rsi, qword ptr [rsp + 0x20]
0x1c1f8:    mov     rbp, rax
0x1c1fb:    movzx   eax, byte ptr [rip + 0xe5eb]

```

```

0x1c202:    mov     word ptr [rbp + 2], r11w
0x1c207:    lea     rdi, [rbp + 4]
0x1c20b:    mov     byte ptr [rbp + 1], al
0x1c20e:    mov     byte ptr [rbp], al
0x1c211:    call    0x48b0
0x1c216:    lea     eax, [r12 + 4]
0x1c21b:    mov     dword ptr [rsp + 0xc], eax
0x1c21f:    test    eax, eax
0x1c221:    jg      0x1b6e6
0x1c227:    mov     rax, qword ptr [rsp + 0x30]
0x1c22c:    mov     qword ptr [rax], r15
0x1c22f:    cmp     qword ptr [rsp + 0x20], rbp
0x1c234:    je      0x1b67a
0x1c23a:    mov     rdi, rbp
0x1c23d:    call    0x18d10
0x1c242:    mov     dword ptr [rsp + 0x3c], 0
0x1c24a:    jmp     0x1b7c0
0x1b7c0:
0x1c24f:    nop
0x1c250:    cmp     al, 0x5e
0x1c252:    je      0x1c478
0x1c258:    add     edx, 1
0x1c25b:    mov     esi, 0x24
0x1c260:    mov     dword ptr [rsp + 0x6c], edx
0x1c264:    jmp     0x1bd98
0x1bd98:
0x1c269:    add     eax, 1
0x1c26c:    mov     rdi, qword ptr [rsp + 0x10]
0x1c271:    movsxd  rsi, eax
0x1c274:    call    0x1d690
0x1c279:    mov     qword ptr [rsp + 0x10], rax
0x1c27e:    jmp     0x1baeb
0x1baeb:
0x1c283:    add     edx, 2
0x1c286:    xor     esi, esi
0x1c288:    movsxd  rdx, edx
0x1c28b:    nop     dword ptr [rax + rax]
0x1c290:    sub     eax, 0x30
0x1c293:    lea     ecx, [rsi + rsi*4]
0x1c296:    mov     r8d, edx
0x1c299:    movsx   eax, al
0x1c29c:    lea     esi, [rax + rcx*2]
0x1c29f:    movzx   eax, byte ptr [rbp + rdx]
0x1c2a4:    add     rdx, 1
0x1c2a8:    lea     ecx, [rax - 0x30]
0x1c2ab:    cmp     cl, 9
0x1c2ae:    jbe     0x1c290
0x1c2b0:    cmp     esi, 0x24

```

```

0x1c2b3:      sete    al
0x1c2b6:      cmp     esi, edi
0x1c2b8:      setge   dl
0x1c2bb:      or      eax, edx
0x1c2bd:      mov     dword ptr [rsp + 0x6c], r8d
0x1c2c2:      test    al, al
0x1c2c4:      je      0x1bdb3
0x1c2ca:      jmp     0x1bd98
0x1bd98:
0x1c2cf:      nop
0x1c2d0:      xor     ebx, ebx
0x1c2d2:      mov     byte ptr [rax], 0
0x1c2d5:      mov     rdi, r10
0x1c2d8:      mov     qword ptr [rsp + 0x48], r8
0x1c2dd:      call    0x18d10
0x1c2e2:      mov     r8, qword ptr [rsp + 0x48]
0x1c2e7:      mov     dword ptr [rip + 0xf81b], ebx
0x1c2ed:      mov     qword ptr [rip + 0xf81c], r8
0x1c2f4:      jmp     0x1be75
0x1be75:
0x1c2f9:      nop     dword ptr [rax]
0x1c300:      cmp     byte ptr [r10 + r14 + 1], 0x26
0x1c306:      lea     rax, [r14 + 1]
0x1c30a:      jne     0x1bcd8
0x1c310:      add     r13d, 1
0x1c314:      mov     r14, rax
0x1c317:      jmp     0x1bcd8
0x1bcd8:
0x1c31c:      nop     dword ptr [rax]
0x1c320:      add     r15d, r15d
0x1c323:      mov     rdi, rbp
0x1c326:      movsxd  rsi, r15d
0x1c329:      call    0x1d690
0x1c32e:      mov     r10, qword ptr [rip + 0xf7db]
0x1c335:      mov     rbp, rax
0x1c338:      jmp     0x1bce1
0x1bce1:
0x1c33d:      lea     r15d, [rax + r15*2]
0x1c341:      mov     rdi, rbp
0x1c344:      movsxd  rsi, r15d
0x1c347:      call    0x1d690
0x1c34c:      mov     r10, qword ptr [rip + 0xf7bd]
0x1c353:      mov     rbp, rax
0x1c356:      jmp     0x1bd1f
0x1bd1f:
0x1c35b:      xor     eax, eax
0x1c35d:      mov     rdi, r15
0x1c360:      call    0x1a380

```

```

0x1c365:      mov     eax, 2
0x1c36a:      jmp     0x1b67c
0x1b67c
0x1c36f:      mov     r8, rbp
0x1c372:      movsxd  rax, ebx
0x1c375:      mov     r15, qword ptr [rsp + 0x58]
0x1c37a:      mov     rbp, qword ptr [rsp + 0x50]
0x1c37f:      add     rax, r8
0x1c382:      jmp     0x1c2d2
0x1c2d2
0x1c387:      cmp     qword ptr [rip + 0xf789], 0
0x1c38f:      mov     eax, dword ptr [rip + 0xf777]
0x1c395:      jne     0x1bc29
0x1c39b:      mov     rbx, qword ptr [rip + 0xf74e]
0x1c3a2:      test    rbx, rbx
0x1c3a5:      je      0x1c3b0
0x1c3a7:      cmp     byte ptr [rbx], 0
0x1c3aa:      jne     0x1c442
0x1c3b0:      mov     dword ptr [rip + 0xf752], 0
0x1c3ba:      mov     r14, r15
0x1c3bd:      mov     r15, qword ptr [rsp + 0x28]
0x1c3c2:      mov     edx, dword ptr [rsp + 0x6c]
0x1c3c6:      mov     ecx, 2
0x1c3cb:      mov     esi, dword ptr [rsp + 0x44]
0x1c3cf:      jmp     0x1b9f7
0x1b9f7
0x1c3d4:      mov     dword ptr [rsp + 0x28], 0
0x1c3dc:      mov     edx, dword ptr [rsp + 0x44]
0x1c3e0:      jmp     0x1bad1
0x1bad1
0x1c3e5:      add     edx, 2
0x1c3e8:      mov     esi, 0x24
0x1c3ed:      mov     dword ptr [rsp + 0x6c], edx
0x1c3f1:      jmp     0x1bd98
0x1bd98
0x1c3f6:      mov     esi, 0xffffffff
0x1c3fb:      jmp     0x1bd98
0x1bd98
0x1c400:      mov     r14, r15
0x1c403:      mov     r15, qword ptr [rsp + 0x28]
0x1c408:      jmp     0x1c3c2
0x1c3c2
0x1c40a:      mov     rax, qword ptr [rsp + 0x30]
0x1c40f:      mov     qword ptr [rax], r15
0x1c412:      jmp     0x1b67a
0x1b67a
0x1c417:      mov     esi, dword ptr [rsp + 0x50]
0x1c41b:      test    esi, esi

```

```

0x1c41d:      je      0x1baa5
0x1c423:      mov     edx, r14d
0x1c426:      mov     ecx, 2
0x1c42b:      mov     r14, r15
0x1c42e:      mov     r15, qword ptr [rsp + 0x28]
0x1c433:      jmp     0x1c3cb
0x1c3cb
0x1c435:      xor     ecx, ecx
0x1c437:      mov     edx, r12d
0x1c43a:      mov     esi, r13d
0x1c43d:      jmp     0x1bdbbe
0x1bdbbe
0x1c442:      mov     rdi, rbx
0x1c445:      call    0x49b0
0x1c44a:      lea     rdi, [rax + 1]
0x1c44e:      call    0x1d670
0x1c453:      mov     rsi, rbx
0x1c456:      mov     rdi, rax
0x1c459:      call    0x48b0
0x1c45e:      mov     rdi, rax
0x1c461:      mov     qword ptr [rip + 0xf6b0], rax
0x1c468:      call    0x49b0
0x1c46d:      mov     dword ptr [rip + 0xf699], eax
0x1c473:      jmp     0x1bc29
0x1bc29
0x1c478:      cmp     edi, 1
0x1c47b:      lea     r8d, [rdx + 1]
0x1c47f:      setle   al
0x1c482:      jmp     0x1c2bd
0x1c2bd
0x1c487:      call    0x49d0
0x1c48c:      nop     dword ptr [rax]
0x1c490:      endbr64
0x1c494:      push    rbx
0x1c495:      mov     rbx, rdx
0x1c498:      add     rsp, -0x80
0x1c49c:      mov     rax, qword ptr fs:[0x28]
0x1c4a5:      mov     qword ptr [rsp + 0x78], rax
0x1c4aa:      xor     eax, eax
0x1c4ac:      mov     rsi, rsp
0x1c4af:      call    0x4cb0
0x1c4b4:      test    eax, eax
0x1c4b6:      js      0x1c514
0x1c4b8:      mov     rax, qword ptr [rsp + 8]
0x1c4bd:      test    rax, rax
0x1c4c0:      cmove   rax, qword ptr [rsp]
0x1c4c5:      mov     qword ptr [rbx], rax
0x1c4c8:      mov     rax, qword ptr [rsp + 0x10]

```

```

0x1c4cd:    mov     qword ptr [rbx + 8], rax
0x1c4d1:    mov     rax, qword ptr [rsp + 0x18]
0x1c4d6:    mov     qword ptr [rbx + 0x10], rax
0x1c4da:    mov     rax, qword ptr [rsp + 0x20]
0x1c4df:    mov     qword ptr [rbx + 0x18], rax
0x1c4e3:    shr     rax, 0x3f
0x1c4e7:    mov     byte ptr [rbx + 0x20], al
0x1c4ea:    mov     rax, qword ptr [rsp + 0x28]
0x1c4ef:    mov     qword ptr [rbx + 0x28], rax
0x1c4f3:    mov     rax, qword ptr [rsp + 0x30]
0x1c4f8:    mov     qword ptr [rbx + 0x30], rax
0x1c4fc:    xor     eax, eax
0x1c4fe:    mov     rdx, qword ptr [rsp + 0x78]
0x1c503:    sub     rdx, qword ptr fs:[0x28]
0x1c50c:    jne     0x1c51b
0x1c50e:    sub     rsp, -0x80
0x1c512:    pop     rbx
0x1c513:    ret
0x1c514:    mov     eax, 0xffffffff
0x1c519:    jmp     0x1c4fe
0x1c4fe
0x1c51b:    call    0x49d0
0x1c520:    endbr64
0x1c524:    push    r15
0x1c526:    push    r14
0x1c528:    push    r13
0x1c52a:    mov     r13, r8
0x1c52d:    push    r12
0x1c52f:    mov     r12, rcx
0x1c532:    push    rbp
0x1c533:    push    rbx
0x1c534:    mov     rbx, rdi
0x1c537:    sub     rsp, 0x98
0x1c53e:    mov     qword ptr [rsp], rsi
0x1c542:    mov     qword ptr [rsp + 0x18], r8
0x1c547:    mov     dword ptr [rsp + 0x48], edx
0x1c54b:    mov     rax, qword ptr fs:[0x28]
0x1c554:    mov     qword ptr [rsp + 0x88], rax
0x1c55c:    mov     eax, edx
0x1c55e:    and     edx, 0x20
0x1c561:    mov     dword ptr [rsp + 0x30], edx
0x1c565:    and     eax, 3
0x1c568:    cmp     edx, 1
0x1c56b:    mov     dword ptr [rsp + 0x10], eax
0x1c56f:    sbb     eax, eax
0x1c571:    and     eax, 0xffffffff8
0x1c574:    add     eax, 0x400
0x1c579:    mov     dword ptr [rsp + 0x34], eax

```

```

0x1c57d:    call    0x4900
0x1c582:    mov     r15, qword ptr [rax]
0x1c585:    mov     r14, rax
0x1c588:    mov     rdi, r15
0x1c58b:    call    0x49b0
0x1c590:    mov     rcx, qword ptr [r14 + 0x10]
0x1c594:    mov     r14, qword ptr [r14 + 8]
0x1c598:    mov     rbp, rax
0x1c59b:    sub     rax, 1
0x1c59f:    cmp     rax, 0x10
0x1c5a3:    mov     eax, 1
0x1c5a8:    mov     rdi, r14
0x1c5ab:    mov     qword ptr [rsp + 0x38], rcx
0x1c5b0:    cmovae  rbp, rax
0x1c5b4:    lea     rax, [rip + 0x5123]
0x1c5bb:    cmovae  r15, rax
0x1c5bf:    call    0x49b0
0x1c5c4:    cmp     rax, 0x10
0x1c5c8:    lea     rax, [rip + 0x5e16]
0x1c5cf:    cmova   r14, rax
0x1c5d3:    mov     rax, qword ptr [rsp]
0x1c5d7:    add     rax, 0x288
0x1c5dd:    mov     qword ptr [rsp + 8], rax
0x1c5e2:    cmp     r13, r12
0x1c5e5:    ja      0x1c740
0x1c5eb:    mov     rax, r12
0x1c5ee:    xor     edx, edx
0x1c5f0:    div     r13
0x1c5f3:    test    rdx, rdx
0x1c5f6:    jne     0x1c604
0x1c5f8:    mul     rbx
0x1c5fb:    mov     rcx, rax
0x1c5fe:    jno     0x1cb98
0x1c604:    mov     qword ptr [rsp + 0x20], r12
0x1c609:    fild    qword ptr [rsp + 0x20]
0x1c60d:    test    r12, r12
0x1c610:    js      0x1c928
0x1c616:    mov     rax, qword ptr [rsp + 0x18]
0x1c61b:    mov     qword ptr [rsp + 0x20], rax
0x1c620:    fild    qword ptr [rsp + 0x20]
0x1c624:    test    rax, rax
0x1c627:    js      0x1c908
0x1c62d:    mov     qword ptr [rsp + 0x20], rbx
0x1c632:    fdivp   st(1)
0x1c634:    fild    qword ptr [rsp + 0x20]
0x1c638:    test    rbx, rbx
0x1c63b:    js      0x1c8f0
0x1c641:    fmulp   st(1)

```

```

0x1c643:    test    byte ptr [rsp + 0x48], 0x10
0x1c648:    je      0x1c938
0x1c64e:    fild     dword ptr [rsp + 0x34]
0x1c652:    xor      ebx, ebx
0x1c654:    fld      st(0)
0x1c656:    jmp      0x1c664
0x1c664:
0x1c658:    nop      dword ptr [rax + rax]
0x1c660:    fstp     st(1)
0x1c662:    fxch     st(2)
0x1c664:    fld      st(0)
0x1c666:    add      ebx, 1
0x1c669:    fmul     st(2)
0x1c66b:    fxch     st(3)
0x1c66d:    fcomi    st(3)
0x1c66f:    jb       0x1c680
0x1c671:    cmp      ebx, 8
0x1c674:    jne      0x1c660
0x1c676:    fstp     st(2)
0x1c678:    fstp     st(2)
0x1c67a:    jmp      0x1c684
0x1c684:
0x1c67c:    nop      dword ptr [rax]
0x1c680:    fstp     st(2)
0x1c682:    fstp     st(2)
0x1c684:    lea      r15, [rbp + 1]
0x1c688:    fdivrp   st(1)
0x1c68a:    cmp      dword ptr [rsp + 0x30], 1
0x1c68f:    adc      rbp, 2
0x1c693:    cmp      dword ptr [rsp + 0x10], 1
0x1c698:    je       0x1cc70
0x1c69e:    fld      xword ptr [rip + 0x5e9c]
0x1c6a4:    fcomip   st(1)
0x1c6a6:    jbe      0x1cb10
0x1c6ac:    fld      dword ptr [rip + 0x5e7e]
0x1c6b2:    fxch     st(1)
0x1c6b4:    fcomi    st(1)
0x1c6b6:    jae      0x1ce00
0x1c6bc:    fstp     st(1)
0x1c6be:    fnstcw   word ptr [rsp + 0x4e]
0x1c6c2:    movzx    eax, word ptr [rsp + 0x4e]
0x1c6c7:    or       ah, 0xc
0x1c6ca:    mov      word ptr [rsp + 0x4c], ax
0x1c6cf:    fld      st(0)
0x1c6d1:    fldcw    word ptr [rsp + 0x4c]
0x1c6d5:    fistp    qword ptr [rsp + 0x20]
0x1c6d9:    fldcw    word ptr [rsp + 0x4e]
0x1c6dd:    mov      rax, qword ptr [rsp + 0x20]

```



```

0x1c6e2:    mov     qword ptr [rsp + 0x20], rax
0x1c6e7:    fild    qword ptr [rsp + 0x20]
0x1c6eb:    test    rax, rax
0x1c6ee:    jns     0x1c6f6
0x1c6f0:    fadd     dword ptr [rip + 0x5e36]
0x1c6f6:    mov     ecx, dword ptr [rsp + 0x10]
0x1c6fa:    test    ecx, ecx
0x1c6fc:    jne     0x1cb18
0x1c702:    fxch     st(1)
0x1c704:    fucomi   st(1)
0x1c706:    jp      0x1c712
0x1c708:    je      0x1cb20
0x1c70e:    fstp     st(1)
0x1c710:    jmp     0x1c718
0x1c718:
0x1c712:    fstp     st(1)
0x1c714:    nop      dword ptr [rax]
0x1c718:    add     rax, 1
0x1c71c:    mov     qword ptr [rsp + 0x20], rax
0x1c721:    fild    qword ptr [rsp + 0x20]
0x1c725:    jns     0x1cb1c
0x1c72b:    fadd     dword ptr [rip + 0x5dfb]
0x1c731:    fxch     st(1)
0x1c733:    jmp     0x1cb20
0x1cb20:
0x1c738:    nop      dword ptr [rax + rax]
0x1c740:    test    r12, r12
0x1c743:    je      0x1c604
0x1c749:    mov     rax, qword ptr [rsp + 0x18]
0x1c74e:    xor     edx, edx
0x1c750:    div     r12
0x1c753:    mov     r8, rax
0x1c756:    test    rdx, rdx
0x1c759:    jne     0x1c604
0x1c75f:    mov     rax, rbx
0x1c762:    xor     edx, edx
0x1c764:    div     r8
0x1c767:    mov     rcx, rax
0x1c76a:    lea     rax, [rdx + rdx*4]
0x1c76e:    xor     edx, edx
0x1c770:    add     rax, rax
0x1c773:    div     r8
0x1c776:    add     rdx, rdx
0x1c779:    mov     edi, eax
0x1c77b:    cmp     r8, rdx
0x1c77e:    jbe     0x1cde8
0x1c784:    xor     esi, esi
0x1c786:    test    rdx, rdx

```

```

0x1c789:      setne    sil
0x1c78d:      mov     r10d, dword ptr [rsp + 0x48]
0x1c792:      and     r10d, 0x10
0x1c796:      je      0x1cbab
0x1c79c:      mov     r8d, dword ptr [rsp + 0x34]
0x1c7a1:      xor     ebx, ebx
0x1c7a3:      mov     r11, r8
0x1c7a6:      cmp     r8, rcx
0x1c7a9:      jbe     0x1cacf
0x1c7af:      cmp     dword ptr [rsp + 0x10], 1
0x1c7b4:      mov     r8, qword ptr [rsp + 8]
0x1c7b9:      je      0x1cdc0
0x1c7bf:      mov     r11d, dword ptr [rsp + 0x10]
0x1c7c4:      test    r11d, r11d
0x1c7c7:      jne     0x1c7e8
0x1c7c9:      add     esi, edi
0x1c7cb:      test    esi, esi
0x1c7cd:      jle     0x1c7e8
0x1c7cf:      add     rcx, 1
0x1c7d3:      test    r10d, r10d
0x1c7d6:      je      0x1c7e8
0x1c7d8:      mov     eax, dword ptr [rsp + 0x34]
0x1c7dc:      cmp     rax, rcx
0x1c7df:      je      0x1cec2
0x1c7e5:      nop     dword ptr [rax]
0x1c7e8:      movabs  rsi, 0xffffffffcccccd
0x1c7f2:      mov     rbp, r8
0x1c7f5:      nop     dword ptr [rax]
0x1c7f8:      mov     rax, rcx
0x1c7fb:      sub     rbp, 1
0x1c7ff:      mul     rsi
0x1c802:      mov     rax, rcx
0x1c805:      shr     rdx, 3
0x1c809:      lea     rdi, [rdx + rdx*4]
0x1c80d:      add     rdi, rdi
0x1c810:      sub     rax, rdi
0x1c813:      add     eax, 0x30
0x1c816:      mov     byte ptr [rbp], al
0x1c819:      mov     rax, rcx
0x1c81c:      mov     rcx, rdx
0x1c81f:      cmp     rax, 9
0x1c823:      ja      0x1c7f8
0x1c825:      test    byte ptr [rsp + 0x48], 4
0x1c82a:      je      0x1ca2c
0x1c830:      mov     rdi, r14
0x1c833:      mov     r12, r8
0x1c836:      mov     qword ptr [rsp + 0x40], r8
0x1c83b:      mov     r13, -1

```

```

0x1c842:      call    0x49b0
0x1c847:      sub     r12, rbp
0x1c84a:      mov     rsi, rbp
0x1c84d:      mov     ecx, 0x29
0x1c852:      mov     r15, rax
0x1c855:      lea     rax, [rsp + 0x50]
0x1c85a:      mov     rdx, r12
0x1c85d:      mov     rdi, rax
0x1c860:      mov     qword ptr [rsp + 0x10], rax
0x1c865:      call    0x4b70
0x1c86a:      mov     dword ptr [rsp + 0x20], ebx
0x1c86e:      mov     rbp, qword ptr [rsp + 0x40]
0x1c873:      mov     rbx, r12
0x1c876:      mov     r12, qword ptr [rsp + 0x38]
0x1c87b:      jmp     0x1c8bb
0x1c8bb:
0x1c87d:      nop     dword ptr [rax]
0x1c880:      cmp     r13, rbx
0x1c883:      mov     rax, qword ptr [rsp + 0x10]
0x1c888:      cmova   r13, rbx
0x1c88c:      sub     rbx, r13
0x1c88f:      lea     rsi, [rax + rbx]
0x1c893:      sub     rbp, r13
0x1c896:      mov     rdx, r13
0x1c899:      mov     rdi, rbp
0x1c89c:      call    0x4bb0
0x1c8a1:      test    rbx, rbx
0x1c8a4:      je      0x1caa0
0x1c8aa:      sub     rbp, r15
0x1c8ad:      mov     rdx, r15
0x1c8b0:      mov     rsi, r14
0x1c8b3:      mov     rdi, rbp
0x1c8b6:      call    0x4bb0
0x1c8bb:      movzx   eax, byte ptr [r12]
0x1c8c0:      test    al, al
0x1c8c2:      je      0x1c880
0x1c8c4:      cmp     al, 0x7e
0x1c8c6:      ja      0x1ca88
0x1c8cc:      cmp     rax, rbx
0x1c8cf:      cmova   rax, rbx
0x1c8d3:      sub     rbx, rax
0x1c8d6:      mov     r13, rax
0x1c8d9:      mov     rax, qword ptr [rsp + 0x10]
0x1c8de:      lea     rsi, [rax + rbx]
0x1c8e2:      add     r12, 1
0x1c8e6:      jmp     0x1c893
0x1c893:
0x1c8e8:      nop     dword ptr [rax + rax]

```

```

0x1c8f0:      fadd     dword ptr [rip + 0x5c36]
0x1c8f6:      fmulp    st(1)
0x1c8f8:      test    byte ptr [rsp + 0x48], 0x10
0x1c8fd:      jne      0x1c64e
0x1c903:      jmp      0x1c938
0x1c938:
0x1c905:      nop      dword ptr [rax]
0x1c908:      fadd     dword ptr [rip + 0x5c1e]
0x1c90e:      mov      qword ptr [rsp + 0x20], rbx
0x1c913:      fdivp     st(1)
0x1c915:      fild     qword ptr [rsp + 0x20]
0x1c919:      test     rbx, rbx
0x1c91c:      jns      0x1c641
0x1c922:      jmp      0x1c8f0
0x1c8f0:
0x1c924:      nop      dword ptr [rax]
0x1c928:      fadd     dword ptr [rip + 0x5bfe]
0x1c92e:      jmp      0x1c616
0x1c616:
0x1c933:      nop      dword ptr [rax + rax]
0x1c938:      cmp      dword ptr [rsp + 0x10], 1
0x1c93d:      je       0x1c9ca
0x1c943:      fld      xword ptr [rip + 0x5bf7]
0x1c949:      fcomip    st(1)
0x1c94b:      jbe      0x1c9ca
0x1c94d:      fld      dword ptr [rip + 0x5bdd]
0x1c953:      fxch     st(1)
0x1c955:      fcomi     st(1)
0x1c957:      jae      0x1ce60
0x1c95d:      fstp     st(1)
0x1c95f:      fnstcw   word ptr [rsp + 0x4e]
0x1c963:      movzx    eax, word ptr [rsp + 0x4e]
0x1c968:      or       ah, 0xc
0x1c96b:      mov      word ptr [rsp + 0x4c], ax
0x1c970:      fld      st(0)
0x1c972:      fldcw    word ptr [rsp + 0x4c]
0x1c976:      fistp    qword ptr [rsp + 0x20]
0x1c97a:      fldcw    word ptr [rsp + 0x4e]
0x1c97e:      mov      rax, qword ptr [rsp + 0x20]
0x1c983:      mov      qword ptr [rsp + 0x20], rax
0x1c988:      fild     qword ptr [rsp + 0x20]
0x1c98c:      test     rax, rax
0x1c98f:      jns      0x1c997
0x1c991:      fadd     dword ptr [rip + 0x5b95]
0x1c997:      mov      r8d, dword ptr [rsp + 0x10]
0x1c99c:      test     r8d, r8d
0x1c99f:      jne      0x1c9c8
0x1c9a1:      fxch     st(1)

```

```

0x1c9a3:      fucomip st(1)
0x1c9a5:      jp      0x1c9ad
0x1c9a7:      je      0x1c9ca
0x1c9a9:      fstp    st(0)
0x1c9ab:      jmp     0x1c9af
0x1c9af
0x1c9ad:      fstp    st(0)
0x1c9af:      add     rax, 1
0x1c9b3:      mov     qword ptr [rsp + 0x10], rax
0x1c9b8:      fild    qword ptr [rsp + 0x10]
0x1c9bc:      js      0x1cee7
0x1c9c2:      jmp     0x1c9ca
0x1c9ca
0x1c9c4:      nop     dword ptr [rax]
0x1c9c8:      fstp    st(1)
0x1c9ca:      sub     rsp, 0x10
0x1c9ce:      mov     rdx, -1
0x1c9d5:      mov     esi, 1
0x1c9da:      xor     eax, eax
0x1c9dc:      mov     rbx, qword ptr [rsp + 0x10]
0x1c9e1:      lea     rcx, [rip + 0x5adc]
0x1c9e8:      fstp    xword ptr [rsp]
0x1c9eb:      mov     rdi, rbx
0x1c9ee:      call    0x4f10
0x1c9f3:      mov     rdi, rbx
0x1c9f6:      mov     ebx, 0xffffffff
0x1c9fb:      call    0x49b0
0x1ca00:      pop     rsi
0x1ca01:      pop     rdi
0x1ca02:      mov     rdx, rax
0x1ca05:      mov     r12, rax
0x1ca08:      mov     rbp, qword ptr [rsp + 8]
0x1ca0d:      mov     rsi, qword ptr [rsp]
0x1ca11:      sub     rbp, rdx
0x1ca14:      mov     rdi, rbp
0x1ca17:      call    0x4d40
0x1ca1c:      lea     r8, [rbp + r12]
0x1ca21:      test    byte ptr [rsp + 0x48], 4
0x1ca26:      jne     0x1c830
0x1ca2c:      test    byte ptr [rsp + 0x48], 0x80
0x1ca31:      je      0x1ca4f
0x1ca33:      cmp     ebx, -1
0x1ca36:      je      0x1cbb8
0x1ca3c:      mov     eax, dword ptr [rsp + 0x48]
0x1ca40:      and     eax, 0x100
0x1ca45:      mov     esi, eax
0x1ca47:      or      esi, ebx
0x1ca49:      jne     0x1cd08

```

```

0x1ca4f:    mov     rax, qword ptr [rsp + 8]
0x1ca54:    mov     byte ptr [rax], 0
0x1ca57:    mov     rax, qword ptr [rsp + 0x88]
0x1ca5f:    sub     rax, qword ptr fs:[0x28]
0x1ca68:    jne     0x1d0eb
0x1ca6e:    add     rsp, 0x98
0x1ca75:    mov     rax, rbp
0x1ca78:    pop     rbx
0x1ca79:    pop     rbp
0x1ca7a:    pop     r12
0x1ca7c:    pop     r13
0x1ca7e:    pop     r14
0x1ca80:    pop     r15
0x1ca82:    ret
0x1ca83:    nop     dword ptr [rax + rax]
0x1ca88:    mov     r13, rbx
0x1ca8b:    mov     rsi, qword ptr [rsp + 0x10]
0x1ca90:    xor     ebx, ebx
0x1ca92:    jmp     0x1c8e2
0x1c8e2
0x1ca97:    nop     word ptr [rax + rax]
0x1caa0:    mov     ebx, dword ptr [rsp + 0x20]
0x1caa4:    jmp     0x1ca2c
0x1ca2c
0x1caa6:    nop     word ptr cs:[rax + rax]
0x1cab0:    test    esi, esi
0x1cab2:    setne   sil
0x1cab6:    movzx   esi, sil
0x1caba:    add     ebx, 1
0x1cabd:    cmp     r8, r9
0x1cac0:    ja      0x1cf18
0x1cac6:    cmp     ebx, 8
0x1cac9:    je      0x1c7af
0x1cacf:    mov     rax, rcx
0x1cad2:    xor     edx, edx
0x1cad4:    mov     ecx, esi
0x1cad6:    div     r8
0x1cad9:    sar     ecx, 1
0x1cadb:    mov     r9, rax
0x1cade:    lea     eax, [rdx + rdx*4]
0x1cae1:    xor     edx, edx
0x1cae3:    lea     eax, [rdi + rax*2]
0x1cae6:    div     r11d
0x1cae9:    lea     edx, [rcx + rdx*2]
0x1caec:    mov     edi, eax
0x1caee:    mov     rcx, r9
0x1caf1:    add     esi, edx
0x1caf3:    cmp     r11d, edx

```

```

0x1caf6:      ja      0x1cab0
0x1caf8:      cmp     r11d, esi
0x1cafb:      setb    sil
0x1caff:      movzx   esi, sil
0x1cb03:      add     esi, 2
0x1cb06:      jmp     0x1caba
0x1caba:
0x1cb08:      nop     dword ptr [rax + rax]
0x1cb10:      fld     st(0)
0x1cb12:      jmp     0x1cb20
0x1cb20:
0x1cb14:      nop     dword ptr [rax]
0x1cb18:      fxch    st(1)
0x1cb1a:      jmp     0x1cb20
0x1cb20:
0x1cb1c:      fxch    st(1)
0x1cb1e:      nop
0x1cb20:      fstp    xword ptr [rsp + 0x20]
0x1cb24:      sub     rsp, 0x10
0x1cb28:      mov     esi, 1
0x1cb2d:      xor     eax, eax
0x1cb2f:      mov     r12, qword ptr [rsp + 0x10]
0x1cb34:      lea     rcx, [rip + 0x598f]
0x1cb3b:      mov     rdx, -1
0x1cb42:      mov     rdi, r12
0x1cb45:      fstp    xword ptr [rsp]
0x1cb48:      call    0x4f10
0x1cb4d:      mov     rdi, r12
0x1cb50:      call    0x49b0
0x1cb55:      mov     rdx, rax
0x1cb58:      pop     rax
0x1cb59:      pop     rcx
0x1cb5a:      fld     xword ptr [rsp + 0x20]
0x1cb5e:      cmp     rdx, rbp
0x1cb61:      ja      0x1cd20
0x1cb67:      test    byte ptr [rsp + 0x48], 8
0x1cb6c:      je      0x1cb88
0x1cb6e:      mov     rax, qword ptr [rsp]
0x1cb72:      cmp     byte ptr [rax + rdx - 1], 0x30
0x1cb77:      je      0x1ce40
0x1cb7d:      fstp    st(0)
0x1cb7f:      jmp     0x1cb8a
0x1cb8a:
0x1cb81:      nop     dword ptr [rax]
0x1cb88:      fstp    st(0)
0x1cb8a:      mov     r12, rdx
0x1cb8d:      sub     r12, r15
0x1cb90:      jmp     0x1ca08

```

```

0x1ca08
0x1cb95:    nop        dword ptr [rax]
0x1cb98:    mov        r10d, dword ptr [rsp + 0x48]
0x1cb9d:    xor        esi, esi
0x1cb9f:    xor        edi, edi
0x1cba1:    and        r10d, 0x10
0x1cba5:    jne        0x1c79c
0x1cbab:    mov        ebx, 0xffffffff
0x1cbb0:    jmp        0x1c7af
0x1c7af
0x1cbb5:    nop        dword ptr [rax]
0x1cbb8:    mov        rcx, qword ptr [rsp + 0x18]
0x1cbbd:    cmp        rcx, 1
0x1cbc1:    jbe        0x1cfca
0x1cbc7:    mov        edx, dword ptr [rsp + 0x34]
0x1cbcb:    mov        ebx, 1
0x1cbd0:    mov        eax, 1
0x1cbd5:    nop        dword ptr [rax]
0x1cbd8:    imul       rax, rdx
0x1cbdc:    cmp        rcx, rax
0x1cbdf:    jbe        0x1cbe9
0x1cbe1:    add        ebx, 1
0x1cbe4:    cmp        ebx, 8
0x1cbe7:    jne        0x1cbd8
0x1cbe9:    mov        esi, dword ptr [rsp + 0x48]
0x1cbec:    mov        eax, esi
0x1cbef:    and        eax, 0x100
0x1cbf4:    and        esi, 0x40
0x1cbf7:    je         0x1cc18
0x1cbf9:    mov        rsi, qword ptr [rsp]
0x1cbfd:    lea        rdi, [rsi + 0x289]
0x1cc04:    mov        byte ptr [rsi + 0x288], 0x20
0x1cc0b:    mov        qword ptr [rsp + 8], rdi
0x1cc10:    test       ebx, ebx
0x1cc12:    je         0x1d105
0x1cc18:    mov        rsi, qword ptr [rsp + 8]
0x1cc1d:    mov        r9d, dword ptr [rsp + 0x30]
0x1cc22:    lea        rdx, [rsi + 1]
0x1cc26:    test       r9d, r9d
0x1cc29:    jne        0x1cc34
0x1cc2b:    cmp        ebx, 1
0x1cc2e:    je         0x1cf00
0x1cc34:    movsxd     rbx, ebx
0x1cc37:    lea        rcx, [rip + 0x58e2]
0x1cc3e:    movzx      ecx, byte ptr [rcx + rbx]
0x1cc42:    mov        byte ptr [rsi], cl
0x1cc44:    test       eax, eax
0x1cc46:    je         0x1cf0b

```



```

0x1cc4c:    mov     r8d, dword ptr [rsp + 0x30]
0x1cc51:    test    r8d, r8d
0x1cc54:    jne     0x1ce30
0x1cc5a:    lea     rax, [rdx + 1]
0x1cc5e:    mov     byte ptr [rdx], 0x42
0x1cc61:    mov     qword ptr [rsp + 8], rax
0x1cc66:    jmp     0x1ca4f
0x1ca4f:
0x1cc6b:    nop     dword ptr [rax + rax]
0x1cc70:    sub     rsp, 0x10
0x1cc74:    mov     rdx, -1
0x1cc7b:    mov     esi, 1
0x1cc80:    xor     eax, eax
0x1cc82:    fld     st(0)
0x1cc84:    fstp    xword ptr [rsp]
0x1cc87:    mov     r12, qword ptr [rsp + 0x10]
0x1cc8c:    lea     rcx, [rip + 0x5837]
0x1cc93:    mov     rdi, r12
0x1cc96:    fstp    xword ptr [rsp + 0x30]
0x1cc9a:    call    0x4f10
0x1cc9f:    mov     rdi, r12
0x1cca2:    call    0x49b0
0x1cca7:    pop     rsi
0x1cca8:    pop     rdi
0x1cca9:    fld     xword ptr [rsp + 0x20]
0x1ccad:    cmp     rax, rbp
0x1ccb0:    mov     rdx, rax
0x1ccb3:    jbe     0x1cb67
0x1ccb9:    fld     dword ptr [rip + 0x5875]
0x1ccbfb:    fmul    st(1), st(0)
0x1ccc1:    fdivp   st(1)
0x1ccc3:    sub     rsp, 0x10
0x1ccc7:    mov     rdx, -1
0x1ccce:    xor     eax, eax
0x1ccd0:    mov     r15, qword ptr [rsp + 0x10]
0x1ccd5:    lea     rcx, [rip + 0x57e8]
0x1ccdc:    mov     esi, 1
0x1cce1:    mov     rdi, r15
0x1cce4:    fstp    xword ptr [rsp]
0x1cce7:    call    0x4f10
0x1ccec:    mov     rdi, r15
0x1ccef:    call    0x49b0
0x1ccf4:    pop     r13
0x1ccf6:    pop     r15
0x1ccf8:    mov     rdx, rax
0x1ccfb:    mov     r12, rax
0x1ccfe:    jmp     0x1ca08
0x1ca08:

```

```

0x1cd03:      nop      dword ptr [rax + rax]
0x1cd08:      test     byte ptr [rsp + 0x48], 0x40
0x1cd0d:      je       0x1cc10
0x1cd13:      jmp      0x1cbf9
0x1cbf9
0x1cd18:      nop      dword ptr [rax + rax]
0x1cd20:      fld      dword ptr [rip + 0x580e]
0x1cd26:      fmul     st(1), st(0)
0x1cd28:      fld      xword ptr [rip + 0x5812]
0x1cd2e:      fcomip   st(2)
0x1cd30:      jbe      0x1ccc1
0x1cd32:      fld      dword ptr [rip + 0x57f8]
0x1cd38:      fxch     st(2)
0x1cd3a:      fcomi    st(2)
0x1cd3c:      jae      0x1ce90
0x1cd42:      fstp     st(2)
0x1cd44:      fxch     st(1)
0x1cd46:      fnstcw   word ptr [rsp + 0x4e]
0x1cd4a:      movzx    eax, word ptr [rsp + 0x4e]
0x1cd4f:      or       ah, 0xc
0x1cd52:      mov      word ptr [rsp + 0x4c], ax
0x1cd57:      fld      st(0)
0x1cd59:      fldcw    word ptr [rsp + 0x4c]
0x1cd5d:      fistp    qword ptr [rsp + 0x20]
0x1cd61:      fldcw    word ptr [rsp + 0x4e]
0x1cd65:      mov      rax, qword ptr [rsp + 0x20]
0x1cd6a:      mov      qword ptr [rsp + 0x20], rax
0x1cd6f:      fild     qword ptr [rsp + 0x20]
0x1cd73:      test     rax, rax
0x1cd76:      jns      0x1cd7e
0x1cd78:      fadd     dword ptr [rip + 0x57ae]
0x1cd7e:      mov      edx, dword ptr [rsp + 0x10]
0x1cd82:      test     edx, edx
0x1cd84:      jne      0x1cdb0
0x1cd86:      fucomi   st(1)
0x1cd88:      fstp     st(1)
0x1cd8a:      jp       0x1cd92
0x1cd8c:      je       0x1cdb2
0x1cd8e:      fstp     st(0)
0x1cd90:      jmp      0x1cd94
0x1cd94
0x1cd92:      fstp     st(0)
0x1cd94:      add      rax, 1
0x1cd98:      mov      qword ptr [rsp + 0x10], rax
0x1cd9d:      fild     qword ptr [rsp + 0x10]
0x1cda1:      js       0x1cef2
0x1cda7:      jmp      0x1cdb2
0x1cdb2

```

```

0x1cda9:      nop      dword ptr [rax]
0x1cdb0:      fstp      st(1)
0x1cdb2:      fxch      st(1)
0x1cdb4:      jmp       0x1ccc1
0x1ccc1
0x1cdb9:      nop      dword ptr [rax]
0x1cdc0:      mov       rax, rcx
0x1cdc3:      movsxd    rsi, esi
0x1cdc6:      and       eax, 1
0x1cdc9:      add       rax, rsi
0x1cdcc:      setne     al
0x1cdcf:      movzx     eax, al
0x1cdd2:      add       eax, edi
0x1cdd4:      cmp       eax, 5
0x1cdd7:      jg        0x1c7cf
0x1cddd:      jmp       0x1c7e8
0x1c7e8
0x1cde2:      nop      word ptr [rax + rax]
0x1cde8:      mov       esi, 2
0x1cded:      mov       eax, 3
0x1cdf2:      cmovb     esi, eax
0x1cdf5:      jmp       0x1c78d
0x1c78d
0x1cdfa:      nop      word ptr [rax + rax]
0x1ce00:      fnstcw    word ptr [rsp + 0x4e]
0x1ce04:      fsubr     st(1), st(0)
0x1ce06:      fxch      st(1)
0x1ce08:      movzx     eax, word ptr [rsp + 0x4e]
0x1ce0d:      or        ah, 0xc
0x1ce10:      mov       word ptr [rsp + 0x4c], ax
0x1ce15:      fldcw     word ptr [rsp + 0x4c]
0x1ce19:      fistp     qword ptr [rsp + 0x20]
0x1ce1d:      fldcw     word ptr [rsp + 0x4e]
0x1ce21:      mov       rax, qword ptr [rsp + 0x20]
0x1ce26:      btc       rax, 0x3f
0x1ce2b:      jmp       0x1c6e2
0x1c6e2
0x1ce30:      mov       byte ptr [rsi + 1], 0x69
0x1ce34:      lea       rdx, [rsi + 2]
0x1ce38:      jmp       0x1cc5a
0x1cc5a
0x1ce3d:      nop      dword ptr [rax]
0x1ce40:      fld       dword ptr [rip + 0x56ee]
0x1ce46:      cmp       dword ptr [rsp + 0x10], 1
0x1ce4b:      fmul      st(1), st(0)
0x1ce4d:      jne       0x1cd28
0x1ce53:      jmp       0x1ccc1
0x1ccc1

```

```

0x1ce58:      nop      dword ptr [rax + rax]
0x1ce60:      fnstcw   word ptr [rsp + 0x4e]
0x1ce64:      fsubr     st(1), st(0)
0x1ce66:      fxch      st(1)
0x1ce68:      movzx     eax, word ptr [rsp + 0x4e]
0x1ce6d:      or        ah, 0xc
0x1ce70:      mov       word ptr [rsp + 0x4c], ax
0x1ce75:      fldcw     word ptr [rsp + 0x4c]
0x1ce79:      fistp     qword ptr [rsp + 0x20]
0x1ce7d:      fldcw     word ptr [rsp + 0x4e]
0x1ce81:      mov       rax, qword ptr [rsp + 0x20]
0x1ce86:      btc       rax, 0x3f
0x1ce8b:      jmp       0x1c983
0x1c983
0x1ce90:      fnstcw     word ptr [rsp + 0x4e]
0x1ce94:      fsubr      st(2), st(0)
0x1ce96:      fxch       st(2)
0x1ce98:      movzx     eax, word ptr [rsp + 0x4e]
0x1ce9d:      or        ah, 0xc
0x1cea0:      mov       word ptr [rsp + 0x4c], ax
0x1cea5:      fldcw     word ptr [rsp + 0x4c]
0x1cea9:      fistp     qword ptr [rsp + 0x20]
0x1cead:      fldcw     word ptr [rsp + 0x4e]
0x1ceb1:      fxch      st(1)
0x1ceb3:      mov       rax, qword ptr [rsp + 0x20]
0x1ceb8:      btc       rax, 0x3f
0x1cebd:      jmp       0x1cd6a
0x1cd6a
0x1cec2:      cmp       ebx, 8
0x1cec5:      je        0x1c7e8
0x1cecb:      add       ebx, 1
0x1cece:      test     byte ptr [rsp + 0x48], 8
0x1ced3:      je        0x1d039
0x1ced9:      mov      byte ptr [r8 - 1], 0x31
0x1cede:      lea      rbp, [r8 - 1]
0x1cee2:      jmp      0x1ca21
0x1ca21
0x1cee7:      fadd     dword ptr [rip + 0x563f]
0x1ceed:      jmp      0x1c9ca
0x1c9ca
0x1cef2:      fadd     dword ptr [rip + 0x5634]
0x1cef8:      fxch     st(1)
0x1cefa:      jmp      0x1ccc1
0x1ccc1
0x1ceff:      nop
0x1cf00:      mov     byte ptr [rsi], 0x6b
0x1cf03:      test    eax, eax
0x1cf05:      jne     0x1cc5a

```

```

0x1cf0b:    mov     qword ptr [rsp + 8], rdx
0x1cf10:    jmp     0x1ca4f
0x1ca4f:
0x1cf15:    nop     dword ptr [rax]
0x1cf18:    cmp     r9, 9
0x1cf1c:    ja      0x1c7af
0x1cf22:    cmp     dword ptr [rsp + 0x10], 1
0x1cf27:    je      0x1cfa0
0x1cf29:    mov     r12d, dword ptr [rsp + 0x10]
0x1cf2e:    test    r12d, r12d
0x1cf31:    jne     0x1cfac
0x1cf33:    test    esi, esi
0x1cf35:    je      0x1cfac
0x1cf37:    lea     edx, [rax + 1]
0x1cf3a:    cmp     eax, 9
0x1cf3d:    je      0x1d024
0x1cf43:    lea     eax, [rdx + 0x30]
0x1cf46:    mov     rdi, qword ptr [rsp]
0x1cf4a:    lea     r8, [rdi + 0x287]
0x1cf51:    mov     byte ptr [rdi + 0x287], al
0x1cf57:    mov     eax, ebp
0x1cf59:    sub     r8, rbp
0x1cf5c:    cmp     ebp, 8
0x1cf5f:    jae     0x1cfd1
0x1cf61:    test    bpl, 4
0x1cf65:    jne     0x1d0d6
0x1cf6b:    test    eax, eax
0x1cf6d:    je      0x1cf7a
0x1cf6f:    movzx   edx, byte ptr [r15]
0x1cf73:    mov     byte ptr [r8], dl
0x1cf76:    test    al, 2
0x1cf78:    jne     0x1cf8e
0x1cf7a:    xor     esi, esi
0x1cf7c:    xor     edi, edi
0x1cf7e:    cmp     dword ptr [rsp + 0x10], 1
0x1cf83:    jne     0x1c7bf
0x1cf89:    jmp     0x1c7e8
0x1c7e8:
0x1cf8e:    movzx   edx, word ptr [r15 + rax - 2]
0x1cf94:    mov     word ptr [r8 + rax - 2], dx
0x1cf9a:    jmp     0x1cf7a
0x1cf7a:
0x1cf9c:    nop     dword ptr [rax]
0x1cfa0:    mov     edx, eax
0x1cfa2:    and     edx, 1
0x1cfa5:    add     edx, esi
0x1cfa7:    cmp     edx, 2
0x1cfaa:    jg      0x1cf37

```

```

0x1cfac:    test    eax, eax
0x1cfae:    jne     0x1d117
0x1cfb4:    mov     r8, qword ptr [rsp + 8]
0x1cfb9:    test    byte ptr [rsp + 0x48], 8
0x1cfbe:    jne     0x1cf7c
0x1cfc0:    mov     eax, 0x30
0x1cfc5:    jmp     0x1cf46
0x1cf46:
0x1cfca:    xor     ebx, ebx
0x1cfcc:    jmp     0x1ca3c
0x1ca3c:
0x1cfd1:    mov     rax, qword ptr [r15]
0x1cfd4:    lea     rsi, [r8 + 8]
0x1cfd8:    mov     rdi, r15
0x1cfdb:    and     rsi, 0xffffffffffffff8
0x1cfd9:    mov     qword ptr [r8], rax
0x1cfe2:    mov     eax, ebp
0x1cfe4:    mov     rdx, qword ptr [r15 + rax - 8]
0x1cfe9:    mov     qword ptr [r8 + rax - 8], rdx
0x1cfee:    mov     rax, r8
0x1cff1:    sub     rax, rsi
0x1cff4:    sub     rdi, rax
0x1cff7:    add     eax, ebp
0x1cff9:    and     eax, 0xffffffff8
0x1cffc:    cmp     eax, 8
0x1cff9:    jb      0x1cf7a
0x1d005:    and     eax, 0xffffffff8
0x1d008:    mov     r9d, eax
0x1d00b:    xor     eax, eax
0x1d00d:    mov     edx, eax
0x1d00f:    add     eax, 8
0x1d012:    mov     r11, qword ptr [rdi + rdx]
0x1d016:    mov     qword ptr [rsi + rdx], r11
0x1d01a:    cmp     eax, r9d
0x1d01d:    jb      0x1d00d
0x1d01f:    jmp     0x1cf7a
0x1cf7a:
0x1d024:    lea     rcx, [r9 + 1]
0x1d028:    cmp     r9, 9
0x1d02c:    je      0x1d0cc
0x1d032:    xor     esi, esi
0x1d034:    jmp     0x1cfb4
0x1cfb4:
0x1d039:    mov     rax, rbp
0x1d03c:    mov     byte ptr [r8 - 1], 0x30
0x1d041:    not     rax
0x1d044:    add     r8, rax
0x1d047:    mov     eax, ebp

```

```

0x1d049:      cmp     ebp, 8
0x1d04c:      jae     0x1d080
0x1d04e:      and     ebp, 4
0x1d051:      jne     0x1d0f0
0x1d057:      test    eax, eax
0x1d059:      je      0x1ced9
0x1d05f:      movzx    edx, byte ptr [r15]
0x1d063:      mov     byte ptr [r8], dl
0x1d066:      test    al, 2
0x1d068:      je      0x1ced9
0x1d06e:      movzx    edx, word ptr [r15 + rax - 2]
0x1d074:      mov     word ptr [r8 + rax - 2], dx
0x1d07a:      jmp     0x1ced9
0x1ced9
0x1d07f:      nop
0x1d080:      mov     rax, qword ptr [r15]
0x1d083:      lea     rcx, [r8 + 8]
0x1d087:      and     rcx, 0xfffffffffffffff8
0x1d08b:      mov     qword ptr [r8], rax
0x1d08e:      mov     eax, ebp
0x1d090:      mov     rdx, qword ptr [r15 + rax - 8]
0x1d095:      mov     qword ptr [r8 + rax - 8], rdx
0x1d09a:      mov     rax, r8
0x1d09d:      sub     rax, rcx
0x1d0a0:      sub     r15, rax
0x1d0a3:      add     eax, ebp
0x1d0a5:      and     eax, 0xffffffff8
0x1d0a8:      cmp     eax, 8
0x1d0ab:      jb      0x1ced9
0x1d0b1:      and     eax, 0xffffffff8
0x1d0b4:      xor     edx, edx
0x1d0b6:      mov     esi, edx
0x1d0b8:      add     edx, 8
0x1d0bb:      mov     rdi, qword ptr [r15 + rsi]
0x1d0bf:      mov     qword ptr [rcx + rsi], rdi
0x1d0c3:      cmp     edx, eax
0x1d0c5:      jb      0x1d0b6
0x1d0c7:      jmp     0x1ced9
0x1ced9
0x1d0cc:      mov     r8, qword ptr [rsp + 8]
0x1d0d1:      jmp     0x1cf7a
0x1cf7a
0x1d0d6:      mov     edx, dword ptr [r15]
0x1d0d9:      mov     dword ptr [r8], edx
0x1d0dc:      mov     edx, dword ptr [r15 + rax - 4]
0x1d0e1:      mov     dword ptr [r8 + rax - 4], edx
0x1d0e6:      jmp     0x1cf7a
0x1cf7a

```

```

0x1d0eb:      call    0x49d0
0x1d0f0:      mov     edx, dword ptr [r15]
0x1d0f3:      mov     dword ptr [r8], edx
0x1d0f6:      mov     edx, dword ptr [r15 + rax - 4]
0x1d0fb:      mov     dword ptr [r8 + rax - 4], edx
0x1d100:      jmp     0x1ced9
0x1ced9
0x1d105:      mov     rdx, qword ptr [rsp + 8]
0x1d10a:      test    eax, eax
0x1d10c:      jne     0x1cc5a
0x1d112:      jmp     0x1ca4f
0x1ca4f
0x1d117:      add     eax, 0x30
0x1d11a:      jmp     0x1cf46
0x1cf46
0x1d11f:      nop
0x1d120:      endbr64
0x1d124:      push    r14
0x1d126:      push    r13
0x1d128:      mov     r13, rsi
0x1d12b:      push    r12
0x1d12d:      push    rbp
0x1d12e:      mov     rbp, rdx
0x1d131:      push    rbx
0x1d132:      mov     rbx, rdi
0x1d135:      sub     rsp, 0x10
0x1d139:      mov     rax, qword ptr fs:[0x28]
0x1d142:      mov     qword ptr [rsp + 8], rax
0x1d147:      xor     eax, eax
0x1d149:      test    rdi, rdi
0x1d14c:      je      0x1d270
0x1d152:      xor     r12d, r12d
0x1d155:      cmp     byte ptr [rbx], 0x27
0x1d158:      jne     0x1d164
0x1d15a:      add     rbx, 1
0x1d15e:      mov     r12d, 4
0x1d164:      lea     r14, [rip + 0x53ad]
0x1d16b:      mov     ecx, 4
0x1d170:      lea     rsi, [rip + 0xb819]
0x1d177:      mov     rdi, rbx
0x1d17a:      mov     rdx, r14
0x1d17d:      call    0x1dc80
0x1d182:      test    eax, eax
0x1d184:      js      0x1d1c0
0x1d186:      cdqe
0x1d188:      mov     qword ptr [rbp], 1
0x1d190:      or      r12d, dword ptr [r14 + rax*4]
0x1d194:      xor     eax, eax

```



```

0x1d196:      mov     dword ptr [r13], r12d
0x1d19a:      mov     rdx, qword ptr [rsp + 8]
0x1d19f:      sub     rdx, qword ptr fs:[0x28]
0x1d1a8:      jne     0x1d2fd
0x1d1ae:      add     rsp, 0x10
0x1d1b2:      pop     rbx
0x1d1b3:      pop     rbp
0x1d1b4:      pop     r12
0x1d1b6:      pop     r13
0x1d1b8:      pop     r14
0x1d1ba:      ret
0x1d1bb:      nop     dword ptr [rax + rax]
0x1d1c0:      xor     edx, edx
0x1d1c2:      mov     rsi, rsp
0x1d1c5:      lea     r8, [rip + 0x5329]
0x1d1cc:      mov     rcx, rbp
0x1d1cf:      mov     rdi, rbx
0x1d1d2:      call    0x1d7e0
0x1d1d7:      test    eax, eax
0x1d1d9:      jne     0x1d228
0x1d1db:      movzx    ecx, byte ptr [rbx]
0x1d1de:      lea     edx, [rcx - 0x30]
0x1d1e1:      mov     rcx, qword ptr [rsp]
0x1d1e5:      cmp     dl, 9
0x1d1e8:      ja      0x1d200
0x1d1ea:      jmp     0x1d217
0x1d217:
0x1d1ec:      nop     dword ptr [rax]
0x1d1f0:      movzx    edi, byte ptr [rbx + 1]
0x1d1f4:      add     rbx, 1
0x1d1f8:      lea     edx, [rdi - 0x30]
0x1d1fb:      cmp     dl, 9
0x1d1fe:      jbe     0x1d217
0x1d200:      cmp     rcx, rbx
0x1d203:      jne     0x1d1f0
0x1d205:      cmp     byte ptr [rcx - 1], 0x42
0x1d209:      je      0x1d2d0
0x1d20f:      or      r12b, 0x80
0x1d213:      or      r12d, 0x20
0x1d217:      mov     rdx, qword ptr [rbp]
0x1d21b:      mov     dword ptr [r13], r12d
0x1d21f:      jmp     0x1d234
0x1d234:
0x1d221:      nop     dword ptr [rax]
0x1d228:      mov     dword ptr [r13], 0
0x1d230:      mov     rdx, qword ptr [rbp]
0x1d234:      test    rdx, rdx
0x1d237:      jne     0x1d19a

```

```

0x1d23d:    lea     rdi, [rip + 0x52a1]
0x1d244:    call   0x47e0
0x1d249:    cmp     rax, 1
0x1d24d:    sbb     rax, rax
0x1d250:    and     eax, 0x200
0x1d255:    add     rax, 0x200
0x1d25b:    mov     qword ptr [rbp], rax
0x1d25f:    mov     eax, 4
0x1d264:    jmp     0x1d19a
0x1d19a:
0x1d269:    nop     dword ptr [rax]
0x1d270:    lea     rdi, [rip + 0x5259]
0x1d277:    call   0x47e0
0x1d27c:    mov     rbx, rax
0x1d27f:    test    rax, rax
0x1d282:    jne     0x1d152
0x1d288:    lea     rdi, [rip + 0x524c]
0x1d28f:    call   0x47e0
0x1d294:    mov     rbx, rax
0x1d297:    test    rax, rax
0x1d29a:    jne     0x1d152
0x1d2a0:    lea     rdi, [rip + 0x523e]
0x1d2a7:    call   0x47e0
0x1d2ac:    test    rax, rax
0x1d2af:    je      0x1d2e6
0x1d2b1:    mov     qword ptr [rbp], 0x200
0x1d2b9:    xor     eax, eax
0x1d2bb:    mov     dword ptr [r13], 0
0x1d2c3:    jmp     0x1d19a
0x1d19a:
0x1d2c8:    nop     dword ptr [rax + rax]
0x1d2d0:    or      r12d, 0x180
0x1d2d7:    cmp     byte ptr [rcx - 2], 0x69
0x1d2db:    jne     0x1d217
0x1d2e1:    jmp     0x1d213
0x1d213:
0x1d2e6:    mov     qword ptr [rbp], 0x400
0x1d2ee:    xor     eax, eax
0x1d2f0:    mov     dword ptr [r13], 0
0x1d2f8:    jmp     0x1d19a
0x1d19a:
0x1d2fd:    call   0x49d0
0x1d302:    nop     word ptr cs:[rax + rax]
0x1d30c:    nop     dword ptr [rax]
0x1d310:    endbr64
0x1d314:    push    r12
0x1d316:    push    rbp
0x1d317:    mov     ebp, edi

```

```

0x1d319:    push    rbx
0x1d31a:    mov     rbx, qword ptr [rip + 0xe81f]
0x1d321:    test    rbx, rbx
0x1d324:    jne     0x1d339
0x1d326:    jmp     0x1d350
0x1d350:
0x1d328:    nop     dword ptr [rax + rax]
0x1d330:    mov     rbx, qword ptr [rbx + 8]
0x1d334:    test    rbx, rbx
0x1d337:    je      0x1d350
0x1d339:    cmp     dword ptr [rbx], ebp
0x1d33b:    jne     0x1d330
0x1d33d:    xor     eax, eax
0x1d33f:    cmp     byte ptr [rbx + 0x10], 0
0x1d343:    je      0x1d349
0x1d345:    lea     rax, [rbx + 0x10]
0x1d349:    pop     rbx
0x1d34a:    pop     rbp
0x1d34b:    pop     r12
0x1d34d:    ret
0x1d34e:    nop
0x1d350:    mov     edi, ebp
0x1d352:    lea     r12, [rip + 0x508c]
0x1d359:    call    0x4970
0x1d35e:    mov     edi, 0x11
0x1d363:    test    rax, rax
0x1d366:    je      0x1d377
0x1d368:    mov     r12, qword ptr [rax]
0x1d36b:    mov     rdi, r12
0x1d36e:    call    0x49b0
0x1d373:    lea     rdi, [rax + 0x11]
0x1d377:    call    0x1d670
0x1d37c:    mov     rsi, r12
0x1d37f:    mov     dword ptr [rax], ebp
0x1d381:    lea     rdi, [rax + 0x10]
0x1d385:    mov     rbx, rax
0x1d388:    call    0x48b0
0x1d38d:    mov     rax, qword ptr [rip + 0xe7ac]
0x1d394:    mov     qword ptr [rip + 0xe7a5], rbx
0x1d39b:    mov     qword ptr [rbx + 8], rax
0x1d39f:    jmp     0x1d33d
0x1d33d:
0x1d3a1:    nop     word ptr cs:[rax + rax]
0x1d3ac:    nop     dword ptr [rax]
0x1d3b0:    endbr64
0x1d3b4:    push    r12
0x1d3b6:    push    rbp
0x1d3b7:    mov     rbp, rdi

```

```

0x1d3ba:    push    rbx
0x1d3bb:    mov     rbx, qword ptr [rip + 0xe77e]
0x1d3c2:    test    rbx, rbx
0x1d3c5:    je      0x1d400
0x1d3c7:    movzx   r12d, byte ptr [rdi]
0x1d3cb:    jmp     0x1d3d9
0x1d3d9:
0x1d3cd:    nop     dword ptr [rax]
0x1d3d0:    mov     rbx, qword ptr [rbx + 8]
0x1d3d4:    test    rbx, rbx
0x1d3d7:    je      0x1d400
0x1d3d9:    cmp     byte ptr [rbx + 0x10], r12b
0x1d3dd:    jne     0x1d3d0
0x1d3df:    lea     rdi, [rbx + 0x10]
0x1d3e3:    mov     rsi, rbp
0x1d3e6:    call    0x4b30
0x1d3eb:    test    eax, eax
0x1d3ed:    jne     0x1d3d0
0x1d3ef:    mov     r12, rbx
0x1d3f2:    pop     rbx
0x1d3f3:    pop     rbp
0x1d3f4:    mov     rax, r12
0x1d3f7:    pop     r12
0x1d3f9:    ret
0x1d3fa:    nop     word ptr [rax + rax]
0x1d400:    mov     rbx, qword ptr [rip + 0xe731]
0x1d407:    test    rbx, rbx
0x1d40a:    je      0x1d448
0x1d40c:    movzx   r12d, byte ptr [rbp]
0x1d411:    jmp     0x1d421
0x1d421:
0x1d413:    nop     dword ptr [rax + rax]
0x1d418:    mov     rbx, qword ptr [rbx + 8]
0x1d41c:    test    rbx, rbx
0x1d41f:    je      0x1d448
0x1d421:    cmp     byte ptr [rbx + 0x10], r12b
0x1d425:    jne     0x1d418
0x1d427:    lea     rdi, [rbx + 0x10]
0x1d42b:    mov     rsi, rbp
0x1d42e:    call    0x4b30
0x1d433:    test    eax, eax
0x1d435:    jne     0x1d418
0x1d437:    xor     r12d, r12d
0x1d43a:    pop     rbx
0x1d43b:    pop     rbp
0x1d43c:    mov     rax, r12
0x1d43f:    pop     r12
0x1d441:    ret

```

```

0x1d442:      nop      word ptr [rax + rax]
0x1d448:      mov      rdi, rbp
0x1d44b:      call     0x4b60
0x1d450:      mov      rdi, rbp
0x1d453:      mov      r12, rax
0x1d456:      call     0x49b0
0x1d45b:      lea      rdi, [rax + 0x11]
0x1d45f:      call     0x1d670
0x1d464:      mov      rsi, rbp
0x1d467:      lea      rdi, [rax + 0x10]
0x1d46b:      mov      rbx, rax
0x1d46e:      call     0x48b0
0x1d473:      test     r12, r12
0x1d476:      je       0x1d4a0
0x1d478:      mov      eax, dword ptr [r12 + 0x10]
0x1d47d:      mov      dword ptr [rbx], eax
0x1d47f:      mov      rax, qword ptr [rip + 0xe6ba]
0x1d486:      mov      qword ptr [rip + 0xe6b3], rbx
0x1d48d:      mov      qword ptr [rbx + 8], rax
0x1d491:      jmp      0x1d3ef
0x1d3ef
0x1d496:      nop      word ptr cs:[rax + rax]
0x1d4a0:      mov      rax, qword ptr [rip + 0xe691]
0x1d4a7:      mov      qword ptr [rip + 0xe68a], rbx
0x1d4ae:      mov      qword ptr [rbx + 8], rax
0x1d4b2:      mov      rax, r12
0x1d4b5:      pop      rbx
0x1d4b6:      pop      rbp
0x1d4b7:      pop      r12
0x1d4b9:      ret
0x1d4ba:      nop      word ptr [rax + rax]
0x1d4c0:      endbr64
0x1d4c4:      push     r12
0x1d4c6:      push     rbp
0x1d4c7:      mov      ebp, edi
0x1d4c9:      push     rbx
0x1d4ca:      mov      rbx, qword ptr [rip + 0xe65f]
0x1d4d1:      test     rbx, rbx
0x1d4d4:      jne      0x1d4e9
0x1d4d6:      jmp      0x1d500
0x1d500
0x1d4d8:      nop      dword ptr [rax + rax]
0x1d4e0:      mov      rbx, qword ptr [rbx + 8]
0x1d4e4:      test     rbx, rbx
0x1d4e7:      je       0x1d500
0x1d4e9:      cmp      dword ptr [rbx], ebp
0x1d4eb:      jne      0x1d4e0
0x1d4ed:      xor      eax, eax

```

```

0x1d4ef:      cmp     byte ptr [rbx + 0x10], 0
0x1d4f3:      je      0x1d4f9
0x1d4f5:      lea     rax, [rbx + 0x10]
0x1d4f9:      pop     rbx
0x1d4fa:      pop     rbp
0x1d4fb:      pop     r12
0x1d4fd:      ret
0x1d4fe:      nop
0x1d500:      mov     edi, ebp
0x1d502:      lea     r12, [rip + 0x4edc]
0x1d509:      call    0x4a10
0x1d50e:      mov     edi, 0x11
0x1d513:      test    rax, rax
0x1d516:      je      0x1d527
0x1d518:      mov     r12, qword ptr [rax]
0x1d51b:      mov     rdi, r12
0x1d51e:      call    0x49b0
0x1d523:      lea     rdi, [rax + 0x11]
0x1d527:      call    0x1d670
0x1d52c:      mov     rsi, r12
0x1d52f:      mov     dword ptr [rax], ebp
0x1d531:      lea     rdi, [rax + 0x10]
0x1d535:      mov     rbx, rax
0x1d538:      call    0x48b0
0x1d53d:      mov     rax, qword ptr [rip + 0xe5ec]
0x1d544:      mov     qword ptr [rip + 0xe5e5], rbx
0x1d54b:      mov     qword ptr [rbx + 8], rax
0x1d54f:      jmp     0x1d4ed
0x1d4ed:
0x1d551:      nop     word ptr cs:[rax + rax]
0x1d55c:      nop     dword ptr [rax]
0x1d560:      endbr64
0x1d564:      push    r12
0x1d566:      push    rbp
0x1d567:      mov     rbp, rdi
0x1d56a:      push    rbx
0x1d56b:      mov     rbx, qword ptr [rip + 0xe5be]
0x1d572:      test    rbx, rbx
0x1d575:      je      0x1d5b0
0x1d577:      movzx   r12d, byte ptr [rdi]
0x1d57b:      jmp     0x1d589
0x1d589:
0x1d57d:      nop     dword ptr [rax]
0x1d580:      mov     rbx, qword ptr [rbx + 8]
0x1d584:      test    rbx, rbx
0x1d587:      je      0x1d5b0
0x1d589:      cmp     byte ptr [rbx + 0x10], r12b
0x1d58d:      jne     0x1d580

```

```

0x1d58f:    lea     rdi, [rbx + 0x10]
0x1d593:    mov     rsi, rbp
0x1d596:    call    0x4b30
0x1d59b:    test    eax, eax
0x1d59d:    jne     0x1d580
0x1d59f:    mov     r12, rbx
0x1d5a2:    pop     rbx
0x1d5a3:    pop     rbp
0x1d5a4:    mov     rax, r12
0x1d5a7:    pop     r12
0x1d5a9:    ret
0x1d5aa:    nop     word ptr [rax + rax]
0x1d5b0:    mov     rbx, qword ptr [rip + 0xe571]
0x1d5b7:    test    rbx, rbx
0x1d5ba:    je      0x1d5f8
0x1d5bc:    movzx   r12d, byte ptr [rbp]
0x1d5c1:    jmp     0x1d5d1
0x1d5d1:
0x1d5c3:    nop     dword ptr [rax + rax]
0x1d5c8:    mov     rbx, qword ptr [rbx + 8]
0x1d5cc:    test    rbx, rbx
0x1d5cf:    je      0x1d5f8
0x1d5d1:    cmp     byte ptr [rbx + 0x10], r12b
0x1d5d5:    jne     0x1d5c8
0x1d5d7:    lea     rdi, [rbx + 0x10]
0x1d5db:    mov     rsi, rbp
0x1d5de:    call    0x4b30
0x1d5e3:    test    eax, eax
0x1d5e5:    jne     0x1d5c8
0x1d5e7:    xor     r12d, r12d
0x1d5ea:    pop     rbx
0x1d5eb:    pop     rbp
0x1d5ec:    mov     rax, r12
0x1d5ef:    pop     r12
0x1d5f1:    ret
0x1d5f2:    nop     word ptr [rax + rax]
0x1d5f8:    mov     rdi, rbp
0x1d5fb:    call    0x4bc0
0x1d600:    mov     rdi, rbp
0x1d603:    mov     r12, rax
0x1d606:    call    0x49b0
0x1d60b:    lea     rdi, [rax + 0x11]
0x1d60f:    call    0x1d670
0x1d614:    mov     rsi, rbp
0x1d617:    lea     rdi, [rax + 0x10]
0x1d61b:    mov     rbx, rax
0x1d61e:    call    0x48b0
0x1d623:    test    r12, r12

```

```

0x1d626:      je      0x1d650
0x1d628:      mov     eax, dword ptr [r12 + 0x10]
0x1d62d:      mov     dword ptr [rbx], eax
0x1d62f:      mov     rax, qword ptr [rip + 0xe4fa]
0x1d636:      mov     qword ptr [rip + 0xe4f3], rbx
0x1d63d:      mov     qword ptr [rbx + 8], rax
0x1d641:      jmp     0x1d59f
0x1d59f:
0x1d646:      nop     word ptr cs:[rax + rax]
0x1d650:      mov     rax, qword ptr [rip + 0xe4d1]
0x1d657:      mov     qword ptr [rip + 0xe4ca], rbx
0x1d65e:      mov     qword ptr [rbx + 8], rax
0x1d662:      mov     rax, r12
0x1d665:      pop     rbx
0x1d666:      pop     rbp
0x1d667:      pop     r12
0x1d669:      ret
0x1d66a:      nop     word ptr [rax + rax]
0x1d670:      endbr64
0x1d674:      push    rbx
0x1d675:      mov     rbx, rdi
0x1d678:      call    0x4c20
0x1d67d:      test    rax, rax
0x1d680:      jne     0x1d687
0x1d682:      test    rbx, rbx
0x1d685:      jne     0x1d689
0x1d687:      pop     rbx
0x1d688:      ret
0x1d689:      call    0x18cf0
0x1d68e:      nop
0x1d690:      endbr64
0x1d694:      push    rbx
0x1d695:      mov     rbx, rsi
0x1d698:      call    0x4ce0
0x1d69d:      test    rax, rax
0x1d6a0:      jne     0x1d6a7
0x1d6a2:      test    rbx, rbx
0x1d6a5:      jne     0x1d6a9
0x1d6a7:      pop     rbx
0x1d6a8:      ret
0x1d6a9:      call    0x18cf0
0x1d6ae:      nop
0x1d6b0:      endbr64
0x1d6b4:      push    rbx
0x1d6b5:      mov     rax, rsi
0x1d6b8:      mov     rsi, qword ptr [rsi]
0x1d6bb:      test    rdi, rdi
0x1d6be:      je      0x1d6f0

```



```

0x1d6c0:      movabs  rdx, 0xaaaaaaaaaaaaa9
0x1d6ca:      cmp     rsi, rdx
0x1d6cd:      ja      0x1d701
0x1d6cf:      lea     rdx, [rsi + 1]
0x1d6d3:      shr     rdx, 1
0x1d6d6:      add     rsi, rdx
0x1d6d9:      setne   bl
0x1d6dc:      mov     qword ptr [rax], rsi
0x1d6df:      call    0x4ce0
0x1d6e4:      test    rax, rax
0x1d6e7:      jne     0x1d6ed
0x1d6e9:      test    bl, bl
0x1d6eb:      jne     0x1d701
0x1d6ed:      pop     rbx
0x1d6ee:      ret
0x1d6ef:      nop
0x1d6f0:      mov     ebx, 1
0x1d6f5:      test    rsi, rsi
0x1d6f8:      jne     0x1d6dc
0x1d6fa:      mov     esi, 0x40
0x1d6ff:      jmp     0x1d6dc
0x1d6dc:
0x1d701:      call    0x18cf0
0x1d706:      nop     word ptr cs:[rax + rax]
0x1d710:      endbr64
0x1d714:      push    rbx
0x1d715:      mov     esi, 1
0x1d71a:      mov     rbx, rdi
0x1d71d:      call    0x4b10
0x1d722:      test    rax, rax
0x1d725:      jne     0x1d72c
0x1d727:      test    rbx, rbx
0x1d72a:      jne     0x1d72e
0x1d72c:      pop     rbx
0x1d72d:      ret
0x1d72e:      call    0x18cf0
0x1d733:      nop     word ptr cs:[rax + rax]
0x1d73e:      nop
0x1d740:      endbr64
0x1d744:      sub     rsp, 8
0x1d748:      call    0x4b10
0x1d74d:      test    rax, rax
0x1d750:      je      0x1d757
0x1d752:      add     rsp, 8
0x1d756:      ret
0x1d757:      call    0x18cf0
0x1d75c:      nop     dword ptr [rax]
0x1d760:      endbr64

```

```

0x1d764:    push    r12
0x1d766:    mov     r12, rsi
0x1d769:    push    rbp
0x1d76a:    mov     rbp, rdi
0x1d76d:    mov     rdi, rsi
0x1d770:    sub     rsp, 8
0x1d774:    call    0x4c20
0x1d779:    mov     rdi, rax
0x1d77c:    test    rax, rax
0x1d77f:    jne     0x1d786
0x1d781:    test    r12, r12
0x1d784:    jne     0x1d798
0x1d786:    add     rsp, 8
0x1d78a:    mov     rdx, r12
0x1d78d:    mov     rsi, rbp
0x1d790:    pop     rbp
0x1d791:    pop     r12
0x1d793:    jmp     0x4bb0
0x4bb0
0x1d798:    call    0x18cf0
0x1d79d:    nop     dword ptr [rax]
0x1d7a0:    endbr64
0x1d7a4:    push    r12
0x1d7a6:    push    rbp
0x1d7a7:    mov     rbp, rdi
0x1d7aa:    sub     rsp, 8
0x1d7ae:    call    0x49b0
0x1d7b3:    lea     r12, [rax + 1]
0x1d7b7:    mov     rdi, r12
0x1d7ba:    call    0x4c20
0x1d7bf:    test    rax, rax
0x1d7c2:    je      0x1d7d9
0x1d7c4:    add     rsp, 8
0x1d7c8:    mov     rdx, r12
0x1d7cb:    mov     rsi, rbp
0x1d7ce:    mov     rdi, rax
0x1d7d1:    pop     rbp
0x1d7d2:    pop     r12
0x1d7d4:    jmp     0x4bb0
0x4bb0
0x1d7d9:    call    0x18cf0
0x1d7de:    nop
0x1d7e0:    endbr64
0x1d7e4:    push    r15
0x1d7e6:    mov     r15, rsi
0x1d7e9:    push    r14
0x1d7eb:    mov     r14, rcx
0x1d7ee:    push    r13

```

```

0x1d7f0:    mov     r13d, edx
0x1d7f3:    push    r12
0x1d7f5:    mov     r12, r8
0x1d7f8:    push    rbp
0x1d7f9:    mov     rbp, rdi
0x1d7fc:    push    rbx
0x1d7fd:    sub     rsp, 0x28
0x1d801:    movzx   ebx, byte ptr [rdi]
0x1d804:    mov     rax, qword ptr fs:[0x28]
0x1d80d:    mov     qword ptr [rsp + 0x18], rax
0x1d812:    xor     eax, eax
0x1d814:    lea     rax, [rsp + 0x10]
0x1d819:    test    rsi, rsi
0x1d81c:    cmovbe  r15, rax
0x1d820:    call    0x4f00
0x1d825:    mov     rcx, qword ptr [rax]
0x1d828:    mov     rax, rbp
0x1d82b:    jmp     0x1d838
0x1d838:
0x1d82d:    nop     dword ptr [rax]
0x1d830:    movzx   ebx, byte ptr [rax + 1]
0x1d834:    add     rax, 1
0x1d838:    movzx   edx, bl
0x1d83b:    test    byte ptr [rcx + rdx*2 + 1], 0x20
0x1d840:    jne     0x1d830
0x1d842:    cmp     bl, 0x2d
0x1d845:    je      0x1d8ca
0x1d84b:    call    0x4860
0x1d850:    mov     edx, r13d
0x1d853:    mov     rsi, r15
0x1d856:    mov     rdi, rbp
0x1d859:    mov     dword ptr [rax], 0
0x1d85f:    mov     qword ptr [rsp + 8], rax
0x1d864:    call    0x4dd0
0x1d869:    mov     r13, qword ptr [r15]
0x1d86c:    mov     rcx, qword ptr [rsp + 8]
0x1d871:    mov     rbx, rax
0x1d874:    cmp     r13, rbp
0x1d877:    je      0x1d8d8
0x1d879:    mov     eax, dword ptr [rcx]
0x1d87b:    test    eax, eax
0x1d87d:    jne     0x1d8c0
0x1d87f:    xor     ebp, ebp
0x1d881:    test    r12, r12
0x1d884:    je      0x1d893
0x1d886:    movzx   edx, byte ptr [r13]
0x1d88b:    test    dl, dl
0x1d88d:    jne     0x1d958

```

```

0x1d893:    mov     qword ptr [r14], rbx
0x1d896:    mov     rax, qword ptr [rsp + 0x18]
0x1d89b:    sub     rax, qword ptr fs:[0x28]
0x1d8a4:    jne     0x1dbf9
0x1d8aa:    add     rsp, 0x28
0x1d8ae:    mov     eax, ebp
0x1d8b0:    pop     rbx
0x1d8b1:    pop     rbp
0x1d8b2:    pop     r12
0x1d8b4:    pop     r13
0x1d8b6:    pop     r14
0x1d8b8:    pop     r15
0x1d8ba:    ret
0x1d8bb:    nop     dword ptr [rax + rax]
0x1d8c0:    mov     ebp, 1
0x1d8c5:    cmp     eax, 0x22
0x1d8c8:    je      0x1d881
0x1d8ca:    mov     ebp, 4
0x1d8cf:    jmp     0x1d896
0x1d896:
0x1d8d1:    nop     dword ptr [rax]
0x1d8d8:    test    r12, r12
0x1d8db:    je      0x1d8ca
0x1d8dd:    movzx   edx, byte ptr [rbp]
0x1d8e1:    test    dl, dl
0x1d8e3:    je      0x1d8ca
0x1d8e5:    movsx   esi, dl
0x1d8e8:    mov     rdi, r12
0x1d8eb:    mov     byte ptr [rsp + 8], dl
0x1d8ef:    xor     ebp, ebp
0x1d8f1:    call    0x4a00
0x1d8f6:    mov     ebx, 1
0x1d8fb:    movzx   edx, byte ptr [rsp + 8]
0x1d900:    test    rax, rax
0x1d903:    je      0x1d8ca
0x1d905:    mov     esi, 0x30
0x1d90a:    mov     rdi, r12
0x1d90d:    mov     byte ptr [rsp + 8], dl
0x1d911:    call    0x4a00
0x1d916:    movzx   edx, byte ptr [rsp + 8]
0x1d91b:    test    rax, rax
0x1d91e:    je      0x1d97c
0x1d920:    movzx   eax, byte ptr [r13 + 1]
0x1d925:    cmp     al, 0x44
0x1d927:    je      0x1d9a2
0x1d929:    cmp     al, 0x69
0x1d92b:    je      0x1dbe0
0x1d931:    cmp     al, 0x42

```

```

0x1d933:      je      0x1d9a2
0x1d935:      sub     edx, 0x42
0x1d938:      cmp     dl, 0x35
0x1d93b:      ja      0x1d971
0x1d93d:      lea     rcx, [rip + 0x4c0c]
0x1d944:      movzx   edx, dl
0x1d947:      movsxd  rax, dword ptr [rcx + rdx*4]
0x1d94b:      add     rax, rcx
0x1d94e:      jmp     rax
rax
->->->                                     indirect JMP
0x1bd48:      cmp     al, 0x39
0x1bd4a:      jg      0x1bd80
0x1bd4c:      cmp     al, 0x2f
0x1bd4e:      jg      0x1c150
0x1bd54:      mov     rdi, r14
0x1bd57:      call    0x49b0
0x1bd5c:      lea     rdi, [rax + 1]
0x1bd60:      call    0x1d670
0x1bd65:      mov     rsi, r14
0x1bd68:      mov     rdi, rax
0x1bd6b:      call    0x48b0
0x1bd70:      mov     r14, rax
0x1bd73:      jmp     0x1b963
0x1b963
0x1bd78:      nop     dword ptr [rax + rax]
0x1bd80:      cmp     al, 0x5e
0x1bd82:      jne     0x1bd54
0x1bd84:      add     edx, 1
0x1bd87:      mov     edi, 1
0x1bd8c:      mov     esi, 1
0x1bd91:      mov     dword ptr [rsp + 0x6c], edx
0x1bd95:      nop     dword ptr [rax]
0x1bd98:      mov     rdx, r14
0x1bd9b:      xor     eax, eax
0x1bd9d:      call    0x1b420
0x1bda2:      mov     r13, rax
0x1bda5:      test    rax, rax
0x1bda8:      jne     0x1b92c
0x1bdae:      mov     r8d, dword ptr [rsp + 0x6c]
0x1bdb3:      mov     ecx, 1
0x1bdb8:      mov     edx, r8d
0x1bdbb:      mov     esi, r12d
0x1bdbe:      mov     rdi, rbp
0x1bdc1:      xor     eax, eax
0x1bdc3:      call    0x19fe0
0x1bdc8:      mov     rdi, qword ptr [rsp + 0x10]
0x1bdcd:      mov     rbx, rax

```

```

0x1bdd0:      call    0x18d10
0x1bdd5:      jmp     0x1ba16
0x1ba16
0x1bdda:      nop     word ptr [rax + rax]
0x1bde0:      cmp     ebx, dword ptr [rsp + 4]
0x1bde4:      jge     0x1bff0
0x1bdea:      lea     eax, [rbx + 1]
0x1bded:      mov     byte ptr [r15 + rbx], 0x5c
0x1bdf2:      movsxd  rdx, eax
0x1bdf5:      mov     dword ptr [rsp + 8], 1
0x1bdfd:      movsxd  rbx, eax
0x1be00:      mov     byte ptr [r15 + rdx], 0
0x1be05:      jmp     0x1b74d
0x1b74d
0x1be0a:      nop     word ptr [rax + rax]
0x1be10:      add     dword ptr [rsp + 4], 0x40
0x1be15:      movsxd  rsi, dword ptr [rsp + 4]
0x1be1a:      mov     rdi, r15
0x1be1d:      call    0x1d690
0x1be22:      mov     r15, rax
0x1be25:      jmp     0x1b778
0x1b778
0x1be2a:      nop     word ptr [rax + rax]
0x1be30:      xor     edx, edx
0x1be32:      test    r13d, r13d
0x1be35:      je      0x1be4d
0x1be37:      movzx   eax, byte ptr [rbp + r14 - 1]
0x1be3d:      cmp     al, 0x27
0x1be3f:      je      0x1c078
0x1be45:      cmp     al, 0x22
0x1be47:      je      0x1c078
0x1be4d:      lea     rsi, [rsp + 0x6c]
0x1be52:      mov     rdi, rbp
0x1be55:      xor     eax, eax
0x1be57:      call    0x1ac70
0x1be5c:      mov     r12d, dword ptr [rsp + 0x6c]
0x1be61:      mov     r14, rax
0x1be64:      jmp     0x1b8b3
0x1b8b3
0x1be69:      nop     dword ptr [rax]
0x1be70:      add     dword ptr [rsp + 0x6c], 2
0x1be75:      mov     rdi, r15
0x1be78:      call    0x49b0
0x1be7d:      mov     r13d, dword ptr [rip + 0xfc88]
0x1be84:      cmp     eax, r13d
0x1be87:      jl      0x1c400
0x1be8d:      mov     qword ptr [rsp + 0x48], rbp
0x1be92:      xor     r14d, r14d

```

```

0x1be95:    mov     rbp, r15
0x1be98:    mov     r15d, eax
0x1be9b:    mov     dword ptr [rsp + 0x50], 1
0x1bea3:    jmp     0x1beba
0x1beba:
0x1bea5:    nop     dword ptr [rax]
0x1bea8:    add     r14d, 1
0x1beac:    lea     eax, [r13 + r14]
0x1beb1:    cmp     eax, r15d
0x1beb4:    jg      0x1c050
0x1beba:    mov     rsi, qword ptr [rip + 0xfc57]
0x1bec1:    movsxd  rbx, r14d
0x1bec4:    lea     rdi, [rbp + rbx]
0x1bec9:    movzx   eax, byte ptr [rsi]
0x1becc:    cmp     byte ptr [rdi], al
0x1bece:    jne     0x1bea8
0x1bed0:    movsxd  rdx, r13d
0x1bed3:    call    0x4890
0x1bed8:    test    eax, eax
0x1beda:    jne     0x1bea8
0x1bedc:    mov     ecx, dword ptr [rip + 0xfc26]
0x1bee2:    add     ecx, r15d
0x1bee5:    sub     ecx, r13d
0x1bee8:    lea     edi, [rcx + 1]
0x1beeb:    mov     dword ptr [rsp + 0x50], ecx
0x1beef:    movsxd  rdi, edi
0x1bef2:    call    0x1d670
0x1bef7:    mov     rdx, rbx
0x1befa:    mov     rsi, rbp
0x1befd:    mov     rdi, rax
0x1bf00:    mov     r13, rax
0x1bf03:    call    0x4880
0x1bf08:    movsxd  rdx, dword ptr [rip + 0xfbf9]
0x1bf0f:    lea     rdi, [r13 + rbx]
0x1bf14:    mov     rsi, qword ptr [rip + 0xfbf5]
0x1bf1b:    call    0x4880
0x1bf20:    movsxd  rsi, dword ptr [rip + 0xfbe5]
0x1bf27:    movsxd  rdi, dword ptr [rip + 0xfbda]
0x1bf2e:    lea     edx, [rsi + r14]
0x1bf32:    add     rdi, rbx
0x1bf35:    add     rsi, rbx
0x1bf38:    sub     r15d, edx
0x1bf3b:    add     rsi, rbp
0x1bf3e:    add     rdi, r13
0x1bf41:    movsxd  rdx, r15d
0x1bf44:    call    0x4880
0x1bf49:    movsxd  rcx, dword ptr [rsp + 0x50]
0x1bf4e:    mov     rdi, rbp

```

```

0x1bf51:    mov     byte ptr [r13 + rcx], 0
0x1bf57:    call   0x18d10
0x1bf5c:    mov     edi, dword ptr [rsp + 0x18]
0x1bf60:    test    edi, edi
0x1bf62:    jne     0x1c1a8
0x1bf68:    mov     rbp, qword ptr [rsp + 0x48]
0x1bf6d:    mov     r14d, dword ptr [rsp + 0x6c]
0x1bf72:    mov     r15, r13
0x1bf75:    jmp     0x1baa5
0x1baa5:
0x1bf7a:    nop     word ptr [rax + rax]
0x1bf80:    add     edx, 1
0x1bf83:    mov     esi, 0x24
0x1bf88:    mov     edi, 0x24
0x1bf8d:    xor     eax, eax
0x1bf8f:    mov     dword ptr [rsp + 0x6c], edx
0x1bf93:    mov     rdx, r14
0x1bf96:    call   0x1b420
0x1bf9b:    mov     r13, rax
0x1bf9e:    lea     rax, [rip + 0xfb7b]
0x1bfa5:    cmp     r13, rax
0x1bfa8:    je      0x1bdae
0x1bfae:    test    r13, r13
0x1bfb1:    jne     0x1b93c
0x1bfb7:    jmp     0x1bd54
0x1bd54:
0x1bfbc:    nop     dword ptr [rax]
0x1bfc0:    cmp     r12b, 0x23
0x1bfc4:    jne     0x1be30
0x1bfca:    lea     r12d, [r13 + 2]
0x1bfce:    mov     r14, r15
0x1bfd1:    mov     dword ptr [rsp + 0x6c], r12d
0x1bfd6:    jmp     0x1b8b3
0x1b8b3:
0x1bfdb:    nop     dword ptr [rax + rax]
0x1bfe0:    mov     qword ptr [rsp + 0x10], rdx
0x1bfe5:    jmp     0x1b806
0x1b806:
0x1bfea:    nop     word ptr [rax + rax]
0x1bff0:    add     dword ptr [rsp + 4], 0x40
0x1bff5:    movsxd  rsi, dword ptr [rsp + 4]
0x1bffa:    mov     rdi, r15
0x1bffd:    call   0x1d690
0x1c002:    mov     r15, rax
0x1c005:    jmp     0x1bdea
0x1bdea:
0x1c00a:    nop     word ptr [rax + rax]
0x1c010:    mov     r13, qword ptr [rip + 0xfad9]

```



```

0x1c017:      add     edx, 1
0x1c01a:      mov     dword ptr [rsp + 0x6c], edx
0x1c01e:      test    r13, r13
0x1c021:      je      0x1c080
0x1c023:      mov     rdi, r13
0x1c026:      call    0x49b0
0x1c02b:      lea     rdi, [rax + 1]
0x1c02f:      call    0x1d670
0x1c034:      mov     rsi, r13
0x1c037:      mov     rdi, rax
0x1c03a:      call    0x48b0
0x1c03f:      mov     r13, rax
0x1c042:      jmp     0x1b92c
0x1b92c
0x1c047:      nop     word ptr [rax + rax]
0x1c050:      cmp     dword ptr [rsp + 0x18], 1
0x1c055:      mov     r15, rbp
0x1c058:      mov     r14d, dword ptr [rsp + 0x6c]
0x1c05d:      mov     rbp, qword ptr [rsp + 0x48]
0x1c062:      jle     0x1c417
0x1c068:      mov     dword ptr [rsp + 0x18], 0
0x1c070:      jmp     0x1baa5
0x1baa5
0x1c075:      nop     dword ptr [rax]
0x1c078:      movsx   edx, al
0x1c07b:      jmp     0x1be4d
0x1be4d
0x1c080:      mov     edi, 1
0x1c085:      call    0x1d670
0x1c08a:      mov     byte ptr [rax], 0
0x1c08d:      mov     r13, rax
0x1c090:      jmp     0x1b92c
0x1b92c
0x1c095:      nop     dword ptr [rax]
0x1c098:      mov     rdi, qword ptr [rsp + 0x10]
0x1c09d:      call    0x49b0
0x1c0a2:      lea     r12d, [rbx + rax]
0x1c0a6:      mov     eax, dword ptr [rsp + 4]
0x1c0aa:      cmp     r12d, eax
0x1c0ad:      jl      0x1c0e0
0x1c0af:      movsxd  r14, eax
0x1c0b2:      mov     rdi, r15
0x1c0b5:      mov     r15d, eax
0x1c0b8:      sub     r14, -0x80
0x1c0bc:      nop     dword ptr [rax]
0x1c0c0:      mov     rsi, r14
0x1c0c3:      sub     r15d, -0x80
0x1c0c7:      sub     r14, -0x80

```

```

0x1c0cb:      call    0x1d690
0x1c0d0:      mov     rdi, rax
0x1c0d3:      cmp     r12d, r15d
0x1c0d6:      jge     0x1c0c0
0x1c0d8:      mov     dword ptr [rsp + 4], r15d
0x1c0dd:      mov     r15, rax
0x1c0e0:      movsxd  rdi, ebx
0x1c0e3:      mov     rsi, qword ptr [rsp + 0x10]
0x1c0e8:      movsxd  rbx, r12d
0x1c0eb:      add     rdi, r15
0x1c0ee:      call    0x48b0
0x1c0f3:      jmp     0x1bb18
0x1bb18
0x1c0f8:      nop     dword ptr [rax + rax]
0x1c100:      xor     edi, edi
0x1c102:      lea     r8d, [rdx + 1]
0x1c106:      movsxd  rax, r8d
0x1c109:      movzx   eax, byte ptr [rbp + rax]
0x1c10e:      lea     ecx, [rax - 0x30]
0x1c111:      cmp     cl, 9
0x1c114:      jbe     0x1c283
0x1c11a:      cmp     al, 0x24
0x1c11c:      je      0x1c3e5
0x1c122:      mov     dword ptr [rsp + 0x6c], r8d
0x1c127:      test    al, al
0x1c129:      je      0x1c3f6
0x1c12f:      cmp     al, 0x3a
0x1c131:      je      0x1c3f6
0x1c137:      test    edi, edi
0x1c139:      jne     0x1bdb3
0x1c13f:      xor     esi, esi
0x1c141:      jmp     0x1bd98
0x1bd98
0x1c146:      nop     word ptr cs:[rax + rax]
0x1c150:      test    esi, esi
0x1c152:      je      0x1bd54
0x1c158:      add     edx, 1
0x1c15b:      xor     edi, edi
0x1c15d:      movsxd  rcx, edx
0x1c160:      sub     eax, 0x30
0x1c163:      lea     edx, [rdi + rdi*4]
0x1c166:      movsx   eax, al
0x1c169:      lea     edi, [rax + rdx*2]
0x1c16c:      movzx   eax, byte ptr [rbp + rcx]
0x1c171:      mov     edx, ecx
0x1c173:      add     rcx, 1
0x1c177:      lea     r8d, [rax - 0x30]
0x1c17b:      cmp     r8b, 9

```

```

0x1c17f:    jbe     0x1c160
0x1c181:    cmp     al, 0x5e
0x1c183:    je      0x1c478
0x1c189:    cmp     al, 0x2a
0x1c18b:    je      0x1c250
0x1c191:    cmp     al, 0x2d
0x1c193:    je      0x1c102
0x1c199:    mov     dword ptr [rsp + 0x6c], edx
0x1c19d:    mov     esi, edi
0x1c19f:    jmp     0x1bd98
0x1bd98:
0x1c1a4:    nop     dword ptr [rax]
0x1c1a8:    mov     rdi, r13
0x1c1ab:    mov     rbp, r13
0x1c1ae:    add     r14d, dword ptr [rip + 0xf953]
0x1c1b5:    call    0x49b0
0x1c1ba:    add     dword ptr [rsp + 0x18], 1
0x1c1bf:    mov     r13d, dword ptr [rip + 0xf946]
0x1c1c6:    mov     dword ptr [rsp + 0x50], 0
0x1c1ce:    mov     r15d, eax
0x1c1d1:    jmp     0x1bea8
0x1bea8:
0x1c1d6:    nop     word ptr cs:[rax + rax]
0x1c1e0:    lea     edi, [r12 + 5]
0x1c1e5:    movsxd  rdi, edi
0x1c1e8:    call    0x1d670
0x1c1ed:    mov     r11d, 0x733a
0x1c1f3:    mov     rsi, qword ptr [rsp + 0x20]
0x1c1f8:    mov     rbp, rax
0x1c1fb:    movzx   eax, byte ptr [rip + 0xe5eb]
0x1c202:    mov     word ptr [rbp + 2], r11w
0x1c207:    lea     rdi, [rbp + 4]
0x1c20b:    mov     byte ptr [rbp + 1], al
0x1c20e:    mov     byte ptr [rbp], al
0x1c211:    call    0x48b0
0x1c216:    lea     eax, [r12 + 4]
0x1c21b:    mov     dword ptr [rsp + 0xc], eax
0x1c21f:    test    eax, eax
0x1c221:    jg      0x1b6e6
0x1c227:    mov     rax, qword ptr [rsp + 0x30]
0x1c22c:    mov     qword ptr [rax], r15
0x1c22f:    cmp     qword ptr [rsp + 0x20], rbp
0x1c234:    je      0x1b67a
0x1c23a:    mov     rdi, rbp
0x1c23d:    call    0x18d10
0x1c242:    mov     dword ptr [rsp + 0x3c], 0
0x1c24a:    jmp     0x1b7c0
0x1b7c0:

```

```

0x1c24f:    nop
0x1c250:    cmp     al, 0x5e
0x1c252:    je      0x1c478
0x1c258:    add     edx, 1
0x1c25b:    mov     esi, 0x24
0x1c260:    mov     dword ptr [rsp + 0x6c], edx
0x1c264:    jmp     0x1bd98
0x1bd98:
0x1c269:    add     eax, 1
0x1c26c:    mov     rdi, qword ptr [rsp + 0x10]
0x1c271:    movsxd  rsi, eax
0x1c274:    call    0x1d690
0x1c279:    mov     qword ptr [rsp + 0x10], rax
0x1c27e:    jmp     0x1baeb
0x1baeb:
0x1c283:    add     edx, 2
0x1c286:    xor     esi, esi
0x1c288:    movsxd  rdx, edx
0x1c28b:    nop     dword ptr [rax + rax]
0x1c290:    sub     eax, 0x30
0x1c293:    lea     ecx, [rsi + rsi*4]
0x1c296:    mov     r8d, edx
0x1c299:    movsx   eax, al
0x1c29c:    lea     esi, [rax + rcx*2]
0x1c29f:    movzx   eax, byte ptr [rbp + rdx]
0x1c2a4:    add     rdx, 1
0x1c2a8:    lea     ecx, [rax - 0x30]
0x1c2ab:    cmp     cl, 9
0x1c2ae:    jbe     0x1c290
0x1c2b0:    cmp     esi, 0x24
0x1c2b3:    sete    al
0x1c2b6:    cmp     esi, edi
0x1c2b8:    setge   dl
0x1c2bb:    or      eax, edx
0x1c2bd:    mov     dword ptr [rsp + 0x6c], r8d
0x1c2c2:    test    al, al
0x1c2c4:    je      0x1bdb3
0x1c2ca:    jmp     0x1bd98
0x1bd98:
0x1c2cf:    nop
0x1c2d0:    xor     ebx, ebx
0x1c2d2:    mov     byte ptr [rax], 0
0x1c2d5:    mov     rdi, r10
0x1c2d8:    mov     qword ptr [rsp + 0x48], r8
0x1c2dd:    call    0x18d10
0x1c2e2:    mov     r8, qword ptr [rsp + 0x48]
0x1c2e7:    mov     dword ptr [rip + 0xf81b], ebx
0x1c2ed:    mov     qword ptr [rip + 0xf81c], r8

```

```

0x1c2f4:      jmp      0x1be75
0x1be75
0x1c2f9:      nop      dword ptr [rax]
0x1c300:      cmp      byte ptr [r10 + r14 + 1], 0x26
0x1c306:      lea      rax, [r14 + 1]
0x1c30a:      jne      0x1bcd8
0x1c310:      add      r13d, 1
0x1c314:      mov      r14, rax
0x1c317:      jmp      0x1bcd8
0x1bcd8
0x1c31c:      nop      dword ptr [rax]
0x1c320:      add      r15d, r15d
0x1c323:      mov      rdi, rbp
0x1c326:      movsxd   rsi, r15d
0x1c329:      call     0x1d690
0x1c32e:      mov      r10, qword ptr [rip + 0xf7db]
0x1c335:      mov      rbp, rax
0x1c338:      jmp      0x1bce1
0x1bce1
0x1c33d:      lea      r15d, [rax + r15*2]
0x1c341:      mov      rdi, rbp
0x1c344:      movsxd   rsi, r15d
0x1c347:      call     0x1d690
0x1c34c:      mov      r10, qword ptr [rip + 0xf7bd]
0x1c353:      mov      rbp, rax
0x1c356:      jmp      0x1bd1f
0x1bd1f
0x1c35b:      xor      eax, eax
0x1c35d:      mov      rdi, r15
0x1c360:      call     0x1a380
0x1c365:      mov      eax, 2
0x1c36a:      jmp      0x1b67c
0x1b67c
0x1c36f:      mov      r8, rbp
0x1c372:      movsxd   rax, ebx
0x1c375:      mov      r15, qword ptr [rsp + 0x58]
0x1c37a:      mov      rbp, qword ptr [rsp + 0x50]
0x1c37f:      add      rax, r8
0x1c382:      jmp      0x1c2d2
0x1c2d2
0x1c387:      cmp      qword ptr [rip + 0xf789], 0
0x1c38f:      mov      eax, dword ptr [rip + 0xf777]
0x1c395:      jne      0x1bc29
0x1c39b:      mov      rbx, qword ptr [rip + 0xf74e]
0x1c3a2:      test     rbx, rbx
0x1c3a5:      je       0x1c3b0
0x1c3a7:      cmp      byte ptr [rbx], 0
0x1c3aa:      jne      0x1c442

```

```

0x1c3b0:    mov     dword ptr [rip + 0xf752], 0
0x1c3ba:    mov     r14, r15
0x1c3bd:    mov     r15, qword ptr [rsp + 0x28]
0x1c3c2:    mov     edx, dword ptr [rsp + 0x6c]
0x1c3c6:    mov     ecx, 2
0x1c3cb:    mov     esi, dword ptr [rsp + 0x44]
0x1c3cf:    jmp     0x1b9f7
0x1b9f7
0x1c3d4:    mov     dword ptr [rsp + 0x28], 0
0x1c3dc:    mov     edx, dword ptr [rsp + 0x44]
0x1c3e0:    jmp     0x1bad1
0x1bad1
0x1c3e5:    add     edx, 2
0x1c3e8:    mov     esi, 0x24
0x1c3ed:    mov     dword ptr [rsp + 0x6c], edx
0x1c3f1:    jmp     0x1bd98
0x1bd98
0x1c3f6:    mov     esi, 0xffffffff
0x1c3fb:    jmp     0x1bd98
0x1bd98
0x1c400:    mov     r14, r15
0x1c403:    mov     r15, qword ptr [rsp + 0x28]
0x1c408:    jmp     0x1c3c2
0x1c3c2
0x1c40a:    mov     rax, qword ptr [rsp + 0x30]
0x1c40f:    mov     qword ptr [rax], r15
0x1c412:    jmp     0x1b67a
0x1b67a
0x1c417:    mov     esi, dword ptr [rsp + 0x50]
0x1c41b:    test    esi, esi
0x1c41d:    je      0x1baa5
0x1c423:    mov     edx, r14d
0x1c426:    mov     ecx, 2
0x1c42b:    mov     r14, r15
0x1c42e:    mov     r15, qword ptr [rsp + 0x28]
0x1c433:    jmp     0x1c3cb
0x1c3cb
0x1c435:    xor     ecx, ecx
0x1c437:    mov     edx, r12d
0x1c43a:    mov     esi, r13d
0x1c43d:    jmp     0x1bdbbe
0x1bdbbe
0x1c442:    mov     rdi, rbx
0x1c445:    call    0x49b0
0x1c44a:    lea     rdi, [rax + 1]
0x1c44e:    call    0x1d670
0x1c453:    mov     rsi, rbx
0x1c456:    mov     rdi, rax

```

```

0x1c459:      call    0x48b0
0x1c45e:      mov     rdi, rax
0x1c461:      mov     qword ptr [rip + 0xf6b0], rax
0x1c468:      call    0x49b0
0x1c46d:      mov     dword ptr [rip + 0xf699], eax
0x1c473:      jmp     0x1bc29
0x1bc29:
0x1c478:      cmp     edi, 1
0x1c47b:      lea     r8d, [rdx + 1]
0x1c47f:      setle   al
0x1c482:      jmp     0x1c2bd
0x1c2bd:
0x1c487:      call    0x49d0
0x1c48c:      nop     dword ptr [rax]
0x1c490:      endbr64
0x1c494:      push    rbx
0x1c495:      mov     rbx, rdx
0x1c498:      add     rsp, -0x80
0x1c49c:      mov     rax, qword ptr fs:[0x28]
0x1c4a5:      mov     qword ptr [rsp + 0x78], rax
0x1c4aa:      xor     eax, eax
0x1c4ac:      mov     rsi, rsp
0x1c4af:      call    0x4cb0
0x1c4b4:      test    eax, eax
0x1c4b6:      js      0x1c514
0x1c4b8:      mov     rax, qword ptr [rsp + 8]
0x1c4bd:      test    rax, rax
0x1c4c0:      cmov     rax, qword ptr [rsp]
0x1c4c5:      mov     qword ptr [rbx], rax
0x1c4c8:      mov     rax, qword ptr [rsp + 0x10]
0x1c4cd:      mov     qword ptr [rbx + 8], rax
0x1c4d1:      mov     rax, qword ptr [rsp + 0x18]
0x1c4d6:      mov     qword ptr [rbx + 0x10], rax
0x1c4da:      mov     rax, qword ptr [rsp + 0x20]
0x1c4df:      mov     qword ptr [rbx + 0x18], rax
0x1c4e3:      shr     rax, 0x3f
0x1c4e7:      mov     byte ptr [rbx + 0x20], al
0x1c4ea:      mov     rax, qword ptr [rsp + 0x28]
0x1c4ef:      mov     qword ptr [rbx + 0x28], rax
0x1c4f3:      mov     rax, qword ptr [rsp + 0x30]
0x1c4f8:      mov     qword ptr [rbx + 0x30], rax
0x1c4fc:      xor     eax, eax
0x1c4fe:      mov     rdx, qword ptr [rsp + 0x78]
0x1c503:      sub     rdx, qword ptr fs:[0x28]
0x1c50c:      jne     0x1c51b
0x1c50e:      sub     rsp, -0x80
0x1c512:      pop     rbx
0x1c513:      ret

```

```

0x1c514:      mov     eax, 0xffffffff
0x1c519:      jmp     0x1c4fe
0x1c4fe
0x1c51b:      call    0x49d0
0x1c520:      endbr64
0x1c524:      push    r15
0x1c526:      push    r14
0x1c528:      push    r13
0x1c52a:      mov     r13, r8
0x1c52d:      push    r12
0x1c52f:      mov     r12, rcx
0x1c532:      push    rbp
0x1c533:      push    rbx
0x1c534:      mov     rbx, rdi
0x1c537:      sub     rsp, 0x98
0x1c53e:      mov     qword ptr [rsp], rsi
0x1c542:      mov     qword ptr [rsp + 0x18], r8
0x1c547:      mov     dword ptr [rsp + 0x48], edx
0x1c54b:      mov     rax, qword ptr fs:[0x28]
0x1c554:      mov     qword ptr [rsp + 0x88], rax
0x1c55c:      mov     eax, edx
0x1c55e:      and     edx, 0x20
0x1c561:      mov     dword ptr [rsp + 0x30], edx
0x1c565:      and     eax, 3
0x1c568:      cmp     edx, 1
0x1c56b:      mov     dword ptr [rsp + 0x10], eax
0x1c56f:      sbb     eax, eax
0x1c571:      and     eax, 0xffffffffe8
0x1c574:      add     eax, 0x400
0x1c579:      mov     dword ptr [rsp + 0x34], eax
0x1c57d:      call    0x4900
0x1c582:      mov     r15, qword ptr [rax]
0x1c585:      mov     r14, rax
0x1c588:      mov     rdi, r15
0x1c58b:      call    0x49b0
0x1c590:      mov     rcx, qword ptr [r14 + 0x10]
0x1c594:      mov     r14, qword ptr [r14 + 8]
0x1c598:      mov     rbp, rax
0x1c59b:      sub     rax, 1
0x1c59f:      cmp     rax, 0x10
0x1c5a3:      mov     eax, 1
0x1c5a8:      mov     rdi, r14
0x1c5ab:      mov     qword ptr [rsp + 0x38], rcx
0x1c5b0:      cmovae  rbp, rax
0x1c5b4:      lea     rax, [rip + 0x5123]
0x1c5bb:      cmovae  r15, rax
0x1c5bf:      call    0x49b0
0x1c5c4:      cmp     rax, 0x10

```



```

0x1c5c8:    lea     rax, [rip + 0x5e16]
0x1c5cf:    cmova   r14, rax
0x1c5d3:    mov     rax, qword ptr [rsp]
0x1c5d7:    add     rax, 0x288
0x1c5dd:    mov     qword ptr [rsp + 8], rax
0x1c5e2:    cmp     r13, r12
0x1c5e5:    ja      0x1c740
0x1c5eb:    mov     rax, r12
0x1c5ee:    xor     edx, edx
0x1c5f0:    div     r13
0x1c5f3:    test    rdx, rdx
0x1c5f6:    jne     0x1c604
0x1c5f8:    mul     rbx
0x1c5fb:    mov     rcx, rax
0x1c5fe:    jno     0x1cb98
0x1c604:    mov     qword ptr [rsp + 0x20], r12
0x1c609:    fild    qword ptr [rsp + 0x20]
0x1c60d:    test    r12, r12
0x1c610:    js      0x1c928
0x1c616:    mov     rax, qword ptr [rsp + 0x18]
0x1c61b:    mov     qword ptr [rsp + 0x20], rax
0x1c620:    fild    qword ptr [rsp + 0x20]
0x1c624:    test    rax, rax
0x1c627:    js      0x1c908
0x1c62d:    mov     qword ptr [rsp + 0x20], rbx
0x1c632:    fdivp   st(1)
0x1c634:    fild    qword ptr [rsp + 0x20]
0x1c638:    test    rbx, rbx
0x1c63b:    js      0x1c8f0
0x1c641:    fmulp   st(1)
0x1c643:    test    byte ptr [rsp + 0x48], 0x10
0x1c648:    je      0x1c938
0x1c64e:    fild    dword ptr [rsp + 0x34]
0x1c652:    xor     ebx, ebx
0x1c654:    fld     st(0)
0x1c656:    jmp     0x1c664
0x1c664:
0x1c658:    nop     dword ptr [rax + rax]
0x1c660:    fstp    st(1)
0x1c662:    fxch    st(2)
0x1c664:    fld     st(0)
0x1c666:    add     ebx, 1
0x1c669:    fmul    st(2)
0x1c66b:    fxch    st(3)
0x1c66d:    fcomi   st(3)
0x1c66f:    jb      0x1c680
0x1c671:    cmp     ebx, 8
0x1c674:    jne     0x1c660

```

```

0x1c676:      fstp    st(2)
0x1c678:      fstp    st(2)
0x1c67a:      jmp     0x1c684
0x1c684:
0x1c67c:      nop     dword ptr [rax]
0x1c680:      fstp    st(2)
0x1c682:      fstp    st(2)
0x1c684:      lea     r15, [rbp + 1]
0x1c688:      fdivrp   st(1)
0x1c68a:      cmp     dword ptr [rsp + 0x30], 1
0x1c68f:      adc     rbp, 2
0x1c693:      cmp     dword ptr [rsp + 0x10], 1
0x1c698:      je      0x1cc70
0x1c69e:      fld     xword ptr [rip + 0x5e9c]
0x1c6a4:      fcomip   st(1)
0x1c6a6:      jbe     0x1cb10
0x1c6ac:      fld     dword ptr [rip + 0x5e7e]
0x1c6b2:      fxch     st(1)
0x1c6b4:      fcomi    st(1)
0x1c6b6:      jae     0x1ce00
0x1c6bc:      fstp    st(1)
0x1c6be:      fnstcw   word ptr [rsp + 0x4e]
0x1c6c2:      movzx    eax, word ptr [rsp + 0x4e]
0x1c6c7:      or      ah, 0xc
0x1c6ca:      mov     word ptr [rsp + 0x4c], ax
0x1c6cf:      fld     st(0)
0x1c6d1:      fldcw    word ptr [rsp + 0x4c]
0x1c6d5:      fistp    qword ptr [rsp + 0x20]
0x1c6d9:      fldcw    word ptr [rsp + 0x4e]
0x1c6dd:      mov     rax, qword ptr [rsp + 0x20]
0x1c6e2:      mov     qword ptr [rsp + 0x20], rax
0x1c6e7:      fild     qword ptr [rsp + 0x20]
0x1c6eb:      test     rax, rax
0x1c6ee:      jns     0x1c6f6
0x1c6f0:      fadd     dword ptr [rip + 0x5e36]
0x1c6f6:      mov     ecx, dword ptr [rsp + 0x10]
0x1c6fa:      test     ecx, ecx
0x1c6fc:      jne     0x1cb18
0x1c702:      fxch     st(1)
0x1c704:      fucomi   st(1)
0x1c706:      jp      0x1c712
0x1c708:      je      0x1cb20
0x1c70e:      fstp    st(1)
0x1c710:      jmp     0x1c718
0x1c718:
0x1c712:      fstp    st(1)
0x1c714:      nop     dword ptr [rax]
0x1c718:      add     rax, 1

```

```

0x1c71c:      mov     qword ptr [rsp + 0x20], rax
0x1c721:      fild   qword ptr [rsp + 0x20]
0x1c725:      jns     0x1cb1c
0x1c72b:      fadd   dword ptr [rip + 0x5dfb]
0x1c731:      fxch   st(1)
0x1c733:      jmp     0x1cb20
0x1cb20:
0x1c738:      nop     dword ptr [rax + rax]
0x1c740:      test   r12, r12
0x1c743:      je      0x1c604
0x1c749:      mov     rax, qword ptr [rsp + 0x18]
0x1c74e:      xor     edx, edx
0x1c750:      div     r12
0x1c753:      mov     r8, rax
0x1c756:      test   rdx, rdx
0x1c759:      jne     0x1c604
0x1c75f:      mov     rax, rbx
0x1c762:      xor     edx, edx
0x1c764:      div     r8
0x1c767:      mov     rcx, rax
0x1c76a:      lea     rax, [rdx + rdx*4]
0x1c76e:      xor     edx, edx
0x1c770:      add     rax, rax
0x1c773:      div     r8
0x1c776:      add     rdx, rdx
0x1c779:      mov     edi, eax
0x1c77b:      cmp     r8, rdx
0x1c77e:      jbe     0x1cde8
0x1c784:      xor     esi, esi
0x1c786:      test   rdx, rdx
0x1c789:      setne   sil
0x1c78d:      mov     r10d, dword ptr [rsp + 0x48]
0x1c792:      and     r10d, 0x10
0x1c796:      je      0x1cbab
0x1c79c:      mov     r8d, dword ptr [rsp + 0x34]
0x1c7a1:      xor     ebx, ebx
0x1c7a3:      mov     r11, r8
0x1c7a6:      cmp     r8, rcx
0x1c7a9:      jbe     0x1cacf
0x1c7af:      cmp     dword ptr [rsp + 0x10], 1
0x1c7b4:      mov     r8, qword ptr [rsp + 8]
0x1c7b9:      je      0x1cdc0
0x1c7bf:      mov     r11d, dword ptr [rsp + 0x10]
0x1c7c4:      test   r11d, r11d
0x1c7c7:      jne     0x1c7e8
0x1c7c9:      add     esi, edi
0x1c7cb:      test   esi, esi
0x1c7cd:      jle     0x1c7e8

```

```

0x1c7cf:      add     rcx, 1
0x1c7d3:      test    r10d, r10d
0x1c7d6:      je      0x1c7e8
0x1c7d8:      mov     eax, dword ptr [rsp + 0x34]
0x1c7dc:      cmp     rax, rcx
0x1c7df:      je      0x1cec2
0x1c7e5:      nop     dword ptr [rax]
0x1c7e8:      movabs   rsi, 0xffffffffcccccd
0x1c7f2:      mov     rbp, r8
0x1c7f5:      nop     dword ptr [rax]
0x1c7f8:      mov     rax, rcx
0x1c7fb:      sub     rbp, 1
0x1c7ff:      mul     rsi
0x1c802:      mov     rax, rcx
0x1c805:      shr     rdx, 3
0x1c809:      lea     rdi, [rdx + rdx*4]
0x1c80d:      add     rdi, rdi
0x1c810:      sub     rax, rdi
0x1c813:      add     eax, 0x30
0x1c816:      mov     byte ptr [rbp], al
0x1c819:      mov     rax, rcx
0x1c81c:      mov     rcx, rdx
0x1c81f:      cmp     rax, 9
0x1c823:      ja      0x1c7f8
0x1c825:      test    byte ptr [rsp + 0x48], 4
0x1c82a:      je      0x1ca2c
0x1c830:      mov     rdi, r14
0x1c833:      mov     r12, r8
0x1c836:      mov     qword ptr [rsp + 0x40], r8
0x1c83b:      mov     r13, -1
0x1c842:      call    0x49b0
0x1c847:      sub     r12, rbp
0x1c84a:      mov     rsi, rbp
0x1c84d:      mov     ecx, 0x29
0x1c852:      mov     r15, rax
0x1c855:      lea     rax, [rsp + 0x50]
0x1c85a:      mov     rdx, r12
0x1c85d:      mov     rdi, rax
0x1c860:      mov     qword ptr [rsp + 0x10], rax
0x1c865:      call    0x4b70
0x1c86a:      mov     dword ptr [rsp + 0x20], ebx
0x1c86e:      mov     rbp, qword ptr [rsp + 0x40]
0x1c873:      mov     rbx, r12
0x1c876:      mov     r12, qword ptr [rsp + 0x38]
0x1c87b:      jmp     0x1c8bb
0x1c8bb:
0x1c87d:      nop     dword ptr [rax]
0x1c880:      cmp     r13, rbx

```

```

0x1c883:    mov     rax, qword ptr [rsp + 0x10]
0x1c888:    cmova  r13, rbx
0x1c88c:    sub     rbx, r13
0x1c88f:    lea     rsi, [rax + rbx]
0x1c893:    sub     rbp, r13
0x1c896:    mov     rdx, r13
0x1c899:    mov     rdi, rbp
0x1c89c:    call    0x4bb0
0x1c8a1:    test    rbx, rbx
0x1c8a4:    je      0x1caa0
0x1c8aa:    sub     rbp, r15
0x1c8ad:    mov     rdx, r15
0x1c8b0:    mov     rsi, r14
0x1c8b3:    mov     rdi, rbp
0x1c8b6:    call    0x4bb0
0x1c8bb:    movzx   eax, byte ptr [r12]
0x1c8c0:    test    al, al
0x1c8c2:    je      0x1c880
0x1c8c4:    cmp     al, 0x7e
0x1c8c6:    ja      0x1ca88
0x1c8cc:    cmp     rax, rbx
0x1c8cf:    cmova  rax, rbx
0x1c8d3:    sub     rbx, rax
0x1c8d6:    mov     r13, rax
0x1c8d9:    mov     rax, qword ptr [rsp + 0x10]
0x1c8de:    lea     rsi, [rax + rbx]
0x1c8e2:    add     r12, 1
0x1c8e6:    jmp     0x1c893
0x1c893:
0x1c8e8:    nop     dword ptr [rax + rax]
0x1c8f0:    fadd    dword ptr [rip + 0x5c36]
0x1c8f6:    fmulp   st(1)
0x1c8f8:    test    byte ptr [rsp + 0x48], 0x10
0x1c8fd:    jne     0x1c64e
0x1c903:    jmp     0x1c938
0x1c938:
0x1c905:    nop     dword ptr [rax]
0x1c908:    fadd    dword ptr [rip + 0x5c1e]
0x1c90e:    mov     qword ptr [rsp + 0x20], rbx
0x1c913:    fdivp   st(1)
0x1c915:    fild    qword ptr [rsp + 0x20]
0x1c919:    test    rbx, rbx
0x1c91c:    jns     0x1c641
0x1c922:    jmp     0x1c8f0
0x1c8f0:
0x1c924:    nop     dword ptr [rax]
0x1c928:    fadd    dword ptr [rip + 0x5bfe]
0x1c92e:    jmp     0x1c616

```

```

0x1c616
0x1c933:    nop        dword ptr [rax + rax]
0x1c938:    cmp        dword ptr [rsp + 0x10], 1
0x1c93d:    je         0x1c9ca
0x1c943:    fld        xword ptr [rip + 0x5bf7]
0x1c949:    fcomip     st(1)
0x1c94b:    jbe        0x1c9ca
0x1c94d:    fld        dword ptr [rip + 0x5bdd]
0x1c953:    fxch       st(1)
0x1c955:    fcomi      st(1)
0x1c957:    jae        0x1ce60
0x1c95d:    fstp       st(1)
0x1c95f:    fnstcw     word ptr [rsp + 0x4e]
0x1c963:    movzx      eax, word ptr [rsp + 0x4e]
0x1c968:    or         ah, 0xc
0x1c96b:    mov        word ptr [rsp + 0x4c], ax
0x1c970:    fld        st(0)
0x1c972:    fldcw      word ptr [rsp + 0x4c]
0x1c976:    fistp      qword ptr [rsp + 0x20]
0x1c97a:    fldcw      word ptr [rsp + 0x4e]
0x1c97e:    mov        rax, qword ptr [rsp + 0x20]
0x1c983:    mov        qword ptr [rsp + 0x20], rax
0x1c988:    fild       qword ptr [rsp + 0x20]
0x1c98c:    test       rax, rax
0x1c98f:    jns        0x1c997
0x1c991:    fadd       dword ptr [rip + 0x5b95]
0x1c997:    mov        r8d, dword ptr [rsp + 0x10]
0x1c99c:    test       r8d, r8d
0x1c99f:    jne        0x1c9c8
0x1c9a1:    fxch       st(1)
0x1c9a3:    fucomip    st(1)
0x1c9a5:    jp         0x1c9ad
0x1c9a7:    je         0x1c9ca
0x1c9a9:    fstp       st(0)
0x1c9ab:    jmp        0x1c9af
0x1c9af
0x1c9ad:    fstp       st(0)
0x1c9af:    add        rax, 1
0x1c9b3:    mov        qword ptr [rsp + 0x10], rax
0x1c9b8:    fild       qword ptr [rsp + 0x10]
0x1c9bc:    js         0x1cee7
0x1c9c2:    jmp        0x1c9ca
0x1c9ca
0x1c9c4:    nop        dword ptr [rax]
0x1c9c8:    fstp       st(1)
0x1c9ca:    sub        rsp, 0x10
0x1c9ce:    mov        rdx, -1
0x1c9d5:    mov        esi, 1

```

```

0x1c9da:    xor     eax, eax
0x1c9dc:    mov     rbx, qword ptr [rsp + 0x10]
0x1c9e1:    lea     rcx, [rip + 0x5adc]
0x1c9e8:    fstp    xword ptr [rsp]
0x1c9eb:    mov     rdi, rbx
0x1c9ee:    call    0x4f10
0x1c9f3:    mov     rdi, rbx
0x1c9f6:    mov     ebx, 0xffffffff
0x1c9fb:    call    0x49b0
0x1ca00:    pop     rsi
0x1ca01:    pop     rdi
0x1ca02:    mov     rdx, rax
0x1ca05:    mov     r12, rax
0x1ca08:    mov     rbp, qword ptr [rsp + 8]
0x1ca0d:    mov     rsi, qword ptr [rsp]
0x1ca11:    sub     rbp, rdx
0x1ca14:    mov     rdi, rbp
0x1ca17:    call    0x4d40
0x1ca1c:    lea     r8, [rbp + r12]
0x1ca21:    test    byte ptr [rsp + 0x48], 4
0x1ca26:    jne     0x1c830
0x1ca2c:    test    byte ptr [rsp + 0x48], 0x80
0x1ca31:    je      0x1ca4f
0x1ca33:    cmp     ebx, -1
0x1ca36:    je      0x1cbb8
0x1ca3c:    mov     eax, dword ptr [rsp + 0x48]
0x1ca40:    and     eax, 0x100
0x1ca45:    mov     esi, eax
0x1ca47:    or      esi, ebx
0x1ca49:    jne     0x1cd08
0x1ca4f:    mov     rax, qword ptr [rsp + 8]
0x1ca54:    mov     byte ptr [rax], 0
0x1ca57:    mov     rax, qword ptr [rsp + 0x88]
0x1ca5f:    sub     rax, qword ptr fs:[0x28]
0x1ca68:    jne     0x1d0eb
0x1ca6e:    add     rsp, 0x98
0x1ca75:    mov     rax, rbp
0x1ca78:    pop     rbx
0x1ca79:    pop     rbp
0x1ca7a:    pop     r12
0x1ca7c:    pop     r13
0x1ca7e:    pop     r14
0x1ca80:    pop     r15
0x1ca82:    ret
0x1ca83:    nop     dword ptr [rax + rax]
0x1ca88:    mov     r13, rbx
0x1ca8b:    mov     rsi, qword ptr [rsp + 0x10]
0x1ca90:    xor     ebx, ebx

```

```

0x1ca92:      jmp      0x1c8e2
0x1c8e2
0x1ca97:      nop      word ptr [rax + rax]
0x1caa0:      mov      ebx, dword ptr [rsp + 0x20]
0x1caa4:      jmp      0x1ca2c
0x1ca2c
0x1caa6:      nop      word ptr cs:[rax + rax]
0x1cab0:      test     esi, esi
0x1cab2:      setne    sil
0x1cab6:      movzx    esi, sil
0x1caba:      add      ebx, 1
0x1cabd:      cmp      r8, r9
0x1cac0:      ja       0x1cf18
0x1cac6:      cmp      ebx, 8
0x1cac9:      je       0x1c7af
0x1cacf:      mov      rax, rcx
0x1cad2:      xor      edx, edx
0x1cad4:      mov      ecx, esi
0x1cad6:      div      r8
0x1cad9:      sar      ecx, 1
0x1cadb:      mov      r9, rax
0x1cade:      lea      eax, [rdx + rdx*4]
0x1cae1:      xor      edx, edx
0x1cae3:      lea      eax, [rdi + rax*2]
0x1cae6:      div      r11d
0x1cae9:      lea      edx, [rcx + rdx*2]
0x1caec:      mov      edi, eax
0x1caee:      mov      rcx, r9
0x1caf1:      add      esi, edx
0x1caf3:      cmp      r11d, edx
0x1caf6:      ja       0x1cab0
0x1caf8:      cmp      r11d, esi
0x1cafb:      setb     sil
0x1caff:      movzx    esi, sil
0x1cb03:      add      esi, 2
0x1cb06:      jmp      0x1caba
0x1caba
0x1cb08:      nop      dword ptr [rax + rax]
0x1cb10:      fld      st(0)
0x1cb12:      jmp      0x1cb20
0x1cb20
0x1cb14:      nop      dword ptr [rax]
0x1cb18:      fxch     st(1)
0x1cb1a:      jmp      0x1cb20
0x1cb20
0x1cb1c:      fxch     st(1)
0x1cb1e:      nop
0x1cb20:      fstp     xword ptr [rsp + 0x20]

```



```

0x1cb24:    sub     rsp, 0x10
0x1cb28:    mov     esi, 1
0x1cb2d:    xor     eax, eax
0x1cb2f:    mov     r12, qword ptr [rsp + 0x10]
0x1cb34:    lea     rcx, [rip + 0x598f]
0x1cb3b:    mov     rdx, -1
0x1cb42:    mov     rdi, r12
0x1cb45:    fstp    xword ptr [rsp]
0x1cb48:    call    0x4f10
0x1cb4d:    mov     rdi, r12
0x1cb50:    call    0x49b0
0x1cb55:    mov     rdx, rax
0x1cb58:    pop     rax
0x1cb59:    pop     rcx
0x1cb5a:    fld     xword ptr [rsp + 0x20]
0x1cb5e:    cmp     rdx, rbp
0x1cb61:    ja      0x1cd20
0x1cb67:    test    byte ptr [rsp + 0x48], 8
0x1cb6c:    je      0x1cb88
0x1cb6e:    mov     rax, qword ptr [rsp]
0x1cb72:    cmp     byte ptr [rax + rdx - 1], 0x30
0x1cb77:    je      0x1ce40
0x1cb7d:    fstp    st(0)
0x1cb7f:    jmp     0x1cb8a
0x1cb8a:
0x1cb81:    nop     dword ptr [rax]
0x1cb88:    fstp    st(0)
0x1cb8a:    mov     r12, rdx
0x1cb8d:    sub     r12, r15
0x1cb90:    jmp     0x1ca08
0x1ca08:
0x1cb95:    nop     dword ptr [rax]
0x1cb98:    mov     r10d, dword ptr [rsp + 0x48]
0x1cb9d:    xor     esi, esi
0x1cb9f:    xor     edi, edi
0x1cba1:    and     r10d, 0x10
0x1cba5:    jne     0x1c79c
0x1cbab:    mov     ebx, 0xffffffff
0x1cbb0:    jmp     0x1c7af
0x1c7af:
0x1cbb5:    nop     dword ptr [rax]
0x1cbb8:    mov     rcx, qword ptr [rsp + 0x18]
0x1cbbd:    cmp     rcx, 1
0x1cbc1:    jbe     0x1cfca
0x1cbc7:    mov     edx, dword ptr [rsp + 0x34]
0x1cbcb:    mov     ebx, 1
0x1cbd0:    mov     eax, 1
0x1cbd5:    nop     dword ptr [rax]

```

```

0x1cbd8:    imul    rax, rdx
0x1cbdc:    cmp     rcx, rax
0x1cbdf:    jbe     0x1cbe9
0x1cbe1:    add     ebx, 1
0x1cbe4:    cmp     ebx, 8
0x1cbe7:    jne     0x1cbd8
0x1cbe9:    mov     esi, dword ptr [rsp + 0x48]
0x1cbec:    mov     eax, esi
0x1cbef:    and     eax, 0x100
0x1cbf4:    and     esi, 0x40
0x1cbf7:    je      0x1cc18
0x1cbf9:    mov     rsi, qword ptr [rsp]
0x1cbfd:    lea     rdi, [rsi + 0x289]
0x1cc04:    mov     byte ptr [rsi + 0x288], 0x20
0x1cc0b:    mov     qword ptr [rsp + 8], rdi
0x1cc10:    test    ebx, ebx
0x1cc12:    je      0x1d105
0x1cc18:    mov     rsi, qword ptr [rsp + 8]
0x1cc1d:    mov     r9d, dword ptr [rsp + 0x30]
0x1cc22:    lea     rdx, [rsi + 1]
0x1cc26:    test    r9d, r9d
0x1cc29:    jne     0x1cc34
0x1cc2b:    cmp     ebx, 1
0x1cc2e:    je      0x1cf00
0x1cc34:    movsxd  rbx, ebx
0x1cc37:    lea     rcx, [rip + 0x58e2]
0x1cc3e:    movzx   ecx, byte ptr [rcx + rbx]
0x1cc42:    mov     byte ptr [rsi], cl
0x1cc44:    test    eax, eax
0x1cc46:    je      0x1cf0b
0x1cc4c:    mov     r8d, dword ptr [rsp + 0x30]
0x1cc51:    test    r8d, r8d
0x1cc54:    jne     0x1ce30
0x1cc5a:    lea     rax, [rdx + 1]
0x1cc5e:    mov     byte ptr [rdx], 0x42
0x1cc61:    mov     qword ptr [rsp + 8], rax
0x1cc66:    jmp     0x1ca4f
0x1ca4f:
0x1cc6b:    nop     dword ptr [rax + rax]
0x1cc70:    sub     rsp, 0x10
0x1cc74:    mov     rdx, -1
0x1cc7b:    mov     esi, 1
0x1cc80:    xor     eax, eax
0x1cc82:    fld     st(0)
0x1cc84:    fstp    xword ptr [rsp]
0x1cc87:    mov     r12, qword ptr [rsp + 0x10]
0x1cc8c:    lea     rcx, [rip + 0x5837]
0x1cc93:    mov     rdi, r12

```

```

0x1cc96:      fstp    xword ptr [rsp + 0x30]
0x1cc9a:      call   0x4f10
0x1cc9f:      mov     rdi, r12
0x1cca2:      call   0x49b0
0x1cca7:      pop     rsi
0x1cca8:      pop     rdi
0x1cca9:      fld     xword ptr [rsp + 0x20]
0x1ccad:      cmp     rax, rbp
0x1ccb0:      mov     rdx, rax
0x1ccb3:      jbe     0x1cb67
0x1ccb9:      fld     dword ptr [rip + 0x5875]
0x1ccbfb:     fmul    st(1), st(0)
0x1ccc1:     fdivp   st(1)
0x1ccc3:     sub     rsp, 0x10
0x1ccc7:     mov     rdx, -1
0x1ccce:     xor     eax, eax
0x1ccd0:     mov     r15, qword ptr [rsp + 0x10]
0x1ccd5:     lea     rcx, [rip + 0x57e8]
0x1ccdc:     mov     esi, 1
0x1cce1:     mov     rdi, r15
0x1cce4:     fstp    xword ptr [rsp]
0x1cce7:     call   0x4f10
0x1ccecb:     mov     rdi, r15
0x1ccf4:     call   0x49b0
0x1ccf6:     pop     r13
0x1ccf8:     pop     r15
0x1ccfb:     mov     rdx, rax
0x1ccfb:     mov     r12, rax
0x1ccfe:     jmp     0x1ca08
0x1ca08:
0x1cd03:     nop     dword ptr [rax + rax]
0x1cd08:     test    byte ptr [rsp + 0x48], 0x40
0x1cd0d:     je      0x1cc10
0x1cd13:     jmp     0x1cbf9
0x1cbf9:
0x1cd18:     nop     dword ptr [rax + rax]
0x1cd20:     fld     dword ptr [rip + 0x580e]
0x1cd26:     fmul    st(1), st(0)
0x1cd28:     fld     xword ptr [rip + 0x5812]
0x1cd2e:     fcomip  st(2)
0x1cd30:     jbe     0x1ccc1
0x1cd32:     fld     dword ptr [rip + 0x57f8]
0x1cd38:     fxch    st(2)
0x1cd3a:     fcomi   st(2)
0x1cd3c:     jae     0x1ce90
0x1cd42:     fstp    st(2)
0x1cd44:     fxch    st(1)
0x1cd46:     fnstcw  word ptr [rsp + 0x4e]

```

```

0x1cd4a:    movzx    eax, word ptr [rsp + 0x4e]
0x1cd4f:    or      ah, 0xc
0x1cd52:    mov     word ptr [rsp + 0x4c], ax
0x1cd57:    fld     st(0)
0x1cd59:    fldcw   word ptr [rsp + 0x4c]
0x1cd5d:    fistp   qword ptr [rsp + 0x20]
0x1cd61:    fldcw   word ptr [rsp + 0x4e]
0x1cd65:    mov     rax, qword ptr [rsp + 0x20]
0x1cd6a:    mov     qword ptr [rsp + 0x20], rax
0x1cd6f:    fild    qword ptr [rsp + 0x20]
0x1cd73:    test    rax, rax
0x1cd76:    jns     0x1cd7e
0x1cd78:    fadd    dword ptr [rip + 0x57ae]
0x1cd7e:    mov     edx, dword ptr [rsp + 0x10]
0x1cd82:    test    edx, edx
0x1cd84:    jne     0x1cdb0
0x1cd86:    fucomi  st(1)
0x1cd88:    fstp    st(1)
0x1cd8a:    jp      0x1cd92
0x1cd8c:    je      0x1cdb2
0x1cd8e:    fstp    st(0)
0x1cd90:    jmp     0x1cd94
0x1cd94:
0x1cd92:    fstp    st(0)
0x1cd94:    add     rax, 1
0x1cd98:    mov     qword ptr [rsp + 0x10], rax
0x1cd9d:    fild    qword ptr [rsp + 0x10]
0x1cda1:    js      0x1cef2
0x1cda7:    jmp     0x1cdb2
0x1cdb2:
0x1cda9:    nop     dword ptr [rax]
0x1cdb0:    fstp    st(1)
0x1cdb2:    fxch    st(1)
0x1cdb4:    jmp     0x1ccc1
0x1ccc1:
0x1cdb9:    nop     dword ptr [rax]
0x1cdc0:    mov     rax, rcx
0x1cdc3:    movsxd  rsi, esi
0x1cdc6:    and     eax, 1
0x1cdc9:    add     rax, rsi
0x1cdcc:    setne   al
0x1cdcf:    movzx   eax, al
0x1cdd2:    add     eax, edi
0x1cdd4:    cmp     eax, 5
0x1cdd7:    jg      0x1c7cf
0x1cddd:    jmp     0x1c7e8
0x1c7e8:
0x1cde2:    nop     word ptr [rax + rax]

```

```

0x1cde8:      mov     esi, 2
0x1cded:      mov     eax, 3
0x1cdf2:      cmovb   esi, eax
0x1cdf5:      jmp     0x1c78d
0x1c78d
0x1cdfa:      nop     word ptr [rax + rax]
0x1ce00:      fnstcw  word ptr [rsp + 0x4e]
0x1ce04:      fsubr    st(1), st(0)
0x1ce06:      fxch     st(1)
0x1ce08:      movzx    eax, word ptr [rsp + 0x4e]
0x1ce0d:      or       ah, 0xc
0x1ce10:      mov     word ptr [rsp + 0x4c], ax
0x1ce15:      fldcw    word ptr [rsp + 0x4c]
0x1ce19:      fistp    qword ptr [rsp + 0x20]
0x1ce1d:      fldcw    word ptr [rsp + 0x4e]
0x1ce21:      mov     rax, qword ptr [rsp + 0x20]
0x1ce26:      btc      rax, 0x3f
0x1ce2b:      jmp     0x1c6e2
0x1c6e2
0x1ce30:      mov     byte ptr [rsi + 1], 0x69
0x1ce34:      lea     rdx, [rsi + 2]
0x1ce38:      jmp     0x1cc5a
0x1cc5a
0x1ce3d:      nop     dword ptr [rax]
0x1ce40:      fld     dword ptr [rip + 0x56ee]
0x1ce46:      cmp     dword ptr [rsp + 0x10], 1
0x1ce4b:      fmul     st(1), st(0)
0x1ce4d:      jne     0x1cd28
0x1ce53:      jmp     0x1ccc1
0x1ccc1
0x1ce58:      nop     dword ptr [rax + rax]
0x1ce60:      fnstcw  word ptr [rsp + 0x4e]
0x1ce64:      fsubr    st(1), st(0)
0x1ce66:      fxch     st(1)
0x1ce68:      movzx    eax, word ptr [rsp + 0x4e]
0x1ce6d:      or       ah, 0xc
0x1ce70:      mov     word ptr [rsp + 0x4c], ax
0x1ce75:      fldcw    word ptr [rsp + 0x4c]
0x1ce79:      fistp    qword ptr [rsp + 0x20]
0x1ce7d:      fldcw    word ptr [rsp + 0x4e]
0x1ce81:      mov     rax, qword ptr [rsp + 0x20]
0x1ce86:      btc      rax, 0x3f
0x1ce8b:      jmp     0x1c983
0x1c983
0x1ce90:      fnstcw  word ptr [rsp + 0x4e]
0x1ce94:      fsubr    st(2), st(0)
0x1ce96:      fxch     st(2)
0x1ce98:      movzx    eax, word ptr [rsp + 0x4e]

```

```

0x1ce9d:      or      ah, 0xc
0x1cea0:      mov     word ptr [rsp + 0x4c], ax
0x1cea5:      fldcw   word ptr [rsp + 0x4c]
0x1cea9:      fistp   qword ptr [rsp + 0x20]
0x1cead:      fldcw   word ptr [rsp + 0x4e]
0x1ceb1:      fxch    st(1)
0x1ceb3:      mov     rax, qword ptr [rsp + 0x20]
0x1ceb8:      btc     rax, 0x3f
0x1cebd:      jmp     0x1cd6a
0x1cd6a
0x1cec2:      cmp     ebx, 8
0x1cec5:      je      0x1c7e8
0x1cecb:      add     ebx, 1
0x1cece:      test    byte ptr [rsp + 0x48], 8
0x1ced3:      je      0x1d039
0x1ced9:      mov     byte ptr [r8 - 1], 0x31
0x1cede:      lea     rbp, [r8 - 1]
0x1cee2:      jmp     0x1ca21
0x1ca21
0x1cee7:      fadd     dword ptr [rip + 0x563f]
0x1ceed:      jmp     0x1c9ca
0x1c9ca
0x1cef2:      fadd     dword ptr [rip + 0x5634]
0x1cef8:      fxch     st(1)
0x1cefa:      jmp     0x1ccc1
0x1ccc1
0x1ceff:      nop
0x1cf00:      mov     byte ptr [rsi], 0x6b
0x1cf03:      test    eax, eax
0x1cf05:      jne     0x1cc5a
0x1cf0b:      mov     qword ptr [rsp + 8], rdx
0x1cf10:      jmp     0x1ca4f
0x1ca4f
0x1cf15:      nop     dword ptr [rax]
0x1cf18:      cmp     r9, 9
0x1cf1c:      ja      0x1c7af
0x1cf22:      cmp     dword ptr [rsp + 0x10], 1
0x1cf27:      je      0x1cfa0
0x1cf29:      mov     r12d, dword ptr [rsp + 0x10]
0x1cf2e:      test    r12d, r12d
0x1cf31:      jne     0x1cfac
0x1cf33:      test    esi, esi
0x1cf35:      je      0x1cfac
0x1cf37:      lea     edx, [rax + 1]
0x1cf3a:      cmp     eax, 9
0x1cf3d:      je      0x1d024
0x1cf43:      lea     eax, [rdx + 0x30]
0x1cf46:      mov     rdi, qword ptr [rsp]

```

```

0x1cf4a:    lea     r8, [rdi + 0x287]
0x1cf51:    mov     byte ptr [rdi + 0x287], al
0x1cf57:    mov     eax, ebp
0x1cf59:    sub     r8, rbp
0x1cf5c:    cmp     ebp, 8
0x1cf5f:    jae     0x1cfd1
0x1cf61:    test    bpl, 4
0x1cf65:    jne     0x1d0d6
0x1cf6b:    test    eax, eax
0x1cf6d:    je      0x1cf7a
0x1cf6f:    movzx   edx, byte ptr [r15]
0x1cf73:    mov     byte ptr [r8], dl
0x1cf76:    test    al, 2
0x1cf78:    jne     0x1cf8e
0x1cf7a:    xor     esi, esi
0x1cf7c:    xor     edi, edi
0x1cf7e:    cmp     dword ptr [rsp + 0x10], 1
0x1cf83:    jne     0x1c7bf
0x1cf89:    jmp     0x1c7e8
0x1c7e8
0x1cf8e:    movzx   edx, word ptr [r15 + rax - 2]
0x1cf94:    mov     word ptr [r8 + rax - 2], dx
0x1cf9a:    jmp     0x1cf7a
0x1cf7a
0x1cf9c:    nop     dword ptr [rax]
0x1cfa0:    mov     edx, eax
0x1cfa2:    and     edx, 1
0x1cfa5:    add     edx, esi
0x1cfa7:    cmp     edx, 2
0x1cfaa:    jg      0x1cf37
0x1cfac:    test    eax, eax
0x1cfae:    jne     0x1d117
0x1cfb4:    mov     r8, qword ptr [rsp + 8]
0x1cfb9:    test    byte ptr [rsp + 0x48], 8
0x1cfbe:    jne     0x1cf7c
0x1cfc0:    mov     eax, 0x30
0x1cfc5:    jmp     0x1cf46
0x1cf46
0x1cfca:    xor     ebx, ebx
0x1cfcc:    jmp     0x1ca3c
0x1ca3c
0x1cfd1:    mov     rax, qword ptr [r15]
0x1cfd4:    lea     rsi, [r8 + 8]
0x1cfd8:    mov     rdi, r15
0x1cfdb:    and     rsi, 0xfffffffffffffff8
0x1cfd1:    mov     qword ptr [r8], rax
0x1cfe2:    mov     eax, ebp
0x1cfe4:    mov     rdx, qword ptr [r15 + rax - 8]

```

```

0x1cfe9:    mov     qword ptr [r8 + rax - 8], rdx
0x1cfee:    mov     rax, r8
0x1cff1:    sub     rax, rsi
0x1cff4:    sub     rdi, rax
0x1cff7:    add     eax, ebp
0x1cff9:    and     eax, 0xffffffff8
0x1cffc:    cmp     eax, 8
0x1cfff:    jb      0x1cf7a
0x1d005:    and     eax, 0xffffffff8
0x1d008:    mov     r9d, eax
0x1d00b:    xor     eax, eax
0x1d00d:    mov     edx, eax
0x1d00f:    add     eax, 8
0x1d012:    mov     r11, qword ptr [rdi + rdx]
0x1d016:    mov     qword ptr [rsi + rdx], r11
0x1d01a:    cmp     eax, r9d
0x1d01d:    jb      0x1d00d
0x1d01f:    jmp     0x1cf7a
0x1cf7a:
0x1d024:    lea     rcx, [r9 + 1]
0x1d028:    cmp     r9, 9
0x1d02c:    je      0x1d0cc
0x1d032:    xor     esi, esi
0x1d034:    jmp     0x1cfb4
0x1cfb4:
0x1d039:    mov     rax, rbp
0x1d03c:    mov     byte ptr [r8 - 1], 0x30
0x1d041:    not     rax
0x1d044:    add     r8, rax
0x1d047:    mov     eax, ebp
0x1d049:    cmp     ebp, 8
0x1d04c:    jae     0x1d080
0x1d04e:    and     ebp, 4
0x1d051:    jne     0x1d0f0
0x1d057:    test    eax, eax
0x1d059:    je      0x1ced9
0x1d05f:    movzx   edx, byte ptr [r15]
0x1d063:    mov     byte ptr [r8], dl
0x1d066:    test    al, 2
0x1d068:    je      0x1ced9
0x1d06e:    movzx   edx, word ptr [r15 + rax - 2]
0x1d074:    mov     word ptr [r8 + rax - 2], dx
0x1d07a:    jmp     0x1ced9
0x1ced9:
0x1d07f:    nop
0x1d080:    mov     rax, qword ptr [r15]
0x1d083:    lea     rcx, [r8 + 8]
0x1d087:    and     rcx, 0xffffffffffffffff8

```



```

0x1d08b:    mov     qword ptr [r8], rax
0x1d08e:    mov     eax, ebp
0x1d090:    mov     rdx, qword ptr [r15 + rax - 8]
0x1d095:    mov     qword ptr [r8 + rax - 8], rdx
0x1d09a:    mov     rax, r8
0x1d09d:    sub     rax, rcx
0x1d0a0:    sub     r15, rax
0x1d0a3:    add     eax, ebp
0x1d0a5:    and     eax, 0xffffffff8
0x1d0a8:    cmp     eax, 8
0x1d0ab:    jb      0x1ced9
0x1d0b1:    and     eax, 0xffffffff8
0x1d0b4:    xor     edx, edx
0x1d0b6:    mov     esi, edx
0x1d0b8:    add     edx, 8
0x1d0bb:    mov     rdi, qword ptr [r15 + rsi]
0x1d0bf:    mov     qword ptr [rcx + rsi], rdi
0x1d0c3:    cmp     edx, eax
0x1d0c5:    jb      0x1d0b6
0x1d0c7:    jmp     0x1ced9
0x1ced9
0x1d0cc:    mov     r8, qword ptr [rsp + 8]
0x1d0d1:    jmp     0x1cf7a
0x1cf7a
0x1d0d6:    mov     edx, dword ptr [r15]
0x1d0d9:    mov     dword ptr [r8], edx
0x1d0dc:    mov     edx, dword ptr [r15 + rax - 4]
0x1d0e1:    mov     dword ptr [r8 + rax - 4], edx
0x1d0e6:    jmp     0x1cf7a
0x1cf7a
0x1d0eb:    call    0x49d0
0x1d0f0:    mov     edx, dword ptr [r15]
0x1d0f3:    mov     dword ptr [r8], edx
0x1d0f6:    mov     edx, dword ptr [r15 + rax - 4]
0x1d0fb:    mov     dword ptr [r8 + rax - 4], edx
0x1d100:    jmp     0x1ced9
0x1ced9
0x1d105:    mov     rdx, qword ptr [rsp + 8]
0x1d10a:    test    eax, eax
0x1d10c:    jne     0x1cc5a
0x1d112:    jmp     0x1ca4f
0x1ca4f
0x1d117:    add     eax, 0x30
0x1d11a:    jmp     0x1cf46
0x1cf46
0x1d11f:    nop
0x1d120:    endbr64
0x1d124:    push    r14

```

```

0x1d126:    push    r13
0x1d128:    mov     r13, rsi
0x1d12b:    push    r12
0x1d12d:    push    rbp
0x1d12e:    mov     rbp, rdx
0x1d131:    push    rbx
0x1d132:    mov     rbx, rdi
0x1d135:    sub     rsp, 0x10
0x1d139:    mov     rax, qword ptr fs:[0x28]
0x1d142:    mov     qword ptr [rsp + 8], rax
0x1d147:    xor     eax, eax
0x1d149:    test    rdi, rdi
0x1d14c:    je      0x1d270
0x1d152:    xor     r12d, r12d
0x1d155:    cmp     byte ptr [rbx], 0x27
0x1d158:    jne     0x1d164
0x1d15a:    add     rbx, 1
0x1d15e:    mov     r12d, 4
0x1d164:    lea     r14, [rip + 0x53ad]
0x1d16b:    mov     ecx, 4
0x1d170:    lea     rsi, [rip + 0xb819]
0x1d177:    mov     rdi, rbx
0x1d17a:    mov     rdx, r14
0x1d17d:    call    0x1dc80
0x1d182:    test    eax, eax
0x1d184:    js      0x1d1c0
0x1d186:    cdqe
0x1d188:    mov     qword ptr [rbp], 1
0x1d190:    or      r12d, dword ptr [r14 + rax*4]
0x1d194:    xor     eax, eax
0x1d196:    mov     dword ptr [r13], r12d
0x1d19a:    mov     rdx, qword ptr [rsp + 8]
0x1d19f:    sub     rdx, qword ptr fs:[0x28]
0x1d1a8:    jne     0x1d2fd
0x1d1ae:    add     rsp, 0x10
0x1d1b2:    pop     rbx
0x1d1b3:    pop     rbp
0x1d1b4:    pop     r12
0x1d1b6:    pop     r13
0x1d1b8:    pop     r14
0x1d1ba:    ret
0x1d1bb:    nop     dword ptr [rax + rax]
0x1d1c0:    xor     edx, edx
0x1d1c2:    mov     rsi, rsp
0x1d1c5:    lea     r8, [rip + 0x5329]
0x1d1cc:    mov     rcx, rbp
0x1d1cf:    mov     rdi, rbx
0x1d1d2:    call    0x1d7e0

```

```

0x1d1d7:    test    eax, eax
0x1d1d9:    jne     0x1d228
0x1d1db:    movzx   ecx, byte ptr [rbx]
0x1d1de:    lea     edx, [rcx - 0x30]
0x1d1e1:    mov     rcx, qword ptr [rsp]
0x1d1e5:    cmp     dl, 9
0x1d1e8:    ja      0x1d200
0x1d1ea:    jmp     0x1d217
0x1d217:
0x1d1ec:    nop     dword ptr [rax]
0x1d1f0:    movzx   edi, byte ptr [rbx + 1]
0x1d1f4:    add     rbx, 1
0x1d1f8:    lea     edx, [rdi - 0x30]
0x1d1fb:    cmp     dl, 9
0x1d1fe:    jbe     0x1d217
0x1d200:    cmp     rcx, rbx
0x1d203:    jne     0x1d1f0
0x1d205:    cmp     byte ptr [rcx - 1], 0x42
0x1d209:    je      0x1d2d0
0x1d20f:    or      r12b, 0x80
0x1d213:    or      r12d, 0x20
0x1d217:    mov     rdx, qword ptr [rbp]
0x1d21b:    mov     dword ptr [r13], r12d
0x1d21f:    jmp     0x1d234
0x1d234:
0x1d221:    nop     dword ptr [rax]
0x1d228:    mov     dword ptr [r13], 0
0x1d230:    mov     rdx, qword ptr [rbp]
0x1d234:    test    rdx, rdx
0x1d237:    jne     0x1d19a
0x1d23d:    lea     rdi, [rip + 0x52a1]
0x1d244:    call    0x47e0
0x1d249:    cmp     rax, 1
0x1d24d:    sbb     rax, rax
0x1d250:    and     eax, 0x200
0x1d255:    add     rax, 0x200
0x1d25b:    mov     qword ptr [rbp], rax
0x1d25f:    mov     eax, 4
0x1d264:    jmp     0x1d19a
0x1d19a:
0x1d269:    nop     dword ptr [rax]
0x1d270:    lea     rdi, [rip + 0x5259]
0x1d277:    call    0x47e0
0x1d27c:    mov     rbx, rax
0x1d27f:    test    rax, rax
0x1d282:    jne     0x1d152
0x1d288:    lea     rdi, [rip + 0x524c]
0x1d28f:    call    0x47e0

```

```

0x1d294:      mov     rbx, rax
0x1d297:      test    rax, rax
0x1d29a:      jne     0x1d152
0x1d2a0:      lea     rdi, [rip + 0x523e]
0x1d2a7:      call    0x47e0
0x1d2ac:      test    rax, rax
0x1d2af:      je      0x1d2e6
0x1d2b1:      mov     qword ptr [rbp], 0x200
0x1d2b9:      xor     eax, eax
0x1d2bb:      mov     dword ptr [r13], 0
0x1d2c3:      jmp     0x1d19a
0x1d19a:
0x1d2c8:      nop     dword ptr [rax + rax]
0x1d2d0:      or      r12d, 0x180
0x1d2d7:      cmp     byte ptr [rcx - 2], 0x69
0x1d2db:      jne     0x1d217
0x1d2e1:      jmp     0x1d213
0x1d213:
0x1d2e6:      mov     qword ptr [rbp], 0x400
0x1d2ee:      xor     eax, eax
0x1d2f0:      mov     dword ptr [r13], 0
0x1d2f8:      jmp     0x1d19a
0x1d19a:
0x1d2fd:      call    0x49d0
0x1d302:      nop     word ptr cs:[rax + rax]
0x1d30c:      nop     dword ptr [rax]
0x1d310:      endbr64
0x1d314:      push    r12
0x1d316:      push    rbp
0x1d317:      mov     ebp, edi
0x1d319:      push    rbx
0x1d31a:      mov     rbx, qword ptr [rip + 0xe81f]
0x1d321:      test    rbx, rbx
0x1d324:      jne     0x1d339
0x1d326:      jmp     0x1d350
0x1d350:
0x1d328:      nop     dword ptr [rax + rax]
0x1d330:      mov     rbx, qword ptr [rbx + 8]
0x1d334:      test    rbx, rbx
0x1d337:      je      0x1d350
0x1d339:      cmp     dword ptr [rbx], ebp
0x1d33b:      jne     0x1d330
0x1d33d:      xor     eax, eax
0x1d33f:      cmp     byte ptr [rbx + 0x10], 0
0x1d343:      je      0x1d349
0x1d345:      lea     rax, [rbx + 0x10]
0x1d349:      pop     rbx
0x1d34a:      pop     rbp

```

```

0x1d34b:    pop     r12
0x1d34d:    ret
0x1d34e:    nop
0x1d350:    mov     edi, ebp
0x1d352:    lea     r12, [rip + 0x508c]
0x1d359:    call    0x4970
0x1d35e:    mov     edi, 0x11
0x1d363:    test    rax, rax
0x1d366:    je      0x1d377
0x1d368:    mov     r12, qword ptr [rax]
0x1d36b:    mov     rdi, r12
0x1d36e:    call    0x49b0
0x1d373:    lea     rdi, [rax + 0x11]
0x1d377:    call    0x1d670
0x1d37c:    mov     rsi, r12
0x1d37f:    mov     dword ptr [rax], ebp
0x1d381:    lea     rdi, [rax + 0x10]
0x1d385:    mov     rbx, rax
0x1d388:    call    0x48b0
0x1d38d:    mov     rax, qword ptr [rip + 0xe7ac]
0x1d394:    mov     qword ptr [rip + 0xe7a5], rbx
0x1d39b:    mov     qword ptr [rbx + 8], rax
0x1d39f:    jmp     0x1d33d
0x1d33d
0x1d3a1:    nop     word ptr cs:[rax + rax]
0x1d3ac:    nop     dword ptr [rax]
0x1d3b0:    endbr64
0x1d3b4:    push    r12
0x1d3b6:    push    rbp
0x1d3b7:    mov     rbp, rdi
0x1d3ba:    push    rbx
0x1d3bb:    mov     rbx, qword ptr [rip + 0xe77e]
0x1d3c2:    test    rbx, rbx
0x1d3c5:    je      0x1d400
0x1d3c7:    movzx   r12d, byte ptr [rdi]
0x1d3cb:    jmp     0x1d3d9
0x1d3d9
0x1d3cd:    nop     dword ptr [rax]
0x1d3d0:    mov     rbx, qword ptr [rbx + 8]
0x1d3d4:    test    rbx, rbx
0x1d3d7:    je      0x1d400
0x1d3d9:    cmp     byte ptr [rbx + 0x10], r12b
0x1d3dd:    jne     0x1d3d0
0x1d3df:    lea     rdi, [rbx + 0x10]
0x1d3e3:    mov     rsi, rbp
0x1d3e6:    call    0x4b30
0x1d3eb:    test    eax, eax
0x1d3ed:    jne     0x1d3d0

```

```

0x1d3ef:    mov     r12, rbx
0x1d3f2:    pop     rbx
0x1d3f3:    pop     rbp
0x1d3f4:    mov     rax, r12
0x1d3f7:    pop     r12
0x1d3f9:    ret
0x1d3fa:    nop     word ptr [rax + rax]
0x1d400:    mov     rbx, qword ptr [rip + 0xe731]
0x1d407:    test    rbx, rbx
0x1d40a:    je      0x1d448
0x1d40c:    movzx   r12d, byte ptr [rbp]
0x1d411:    jmp     0x1d421
0x1d421:
0x1d413:    nop     dword ptr [rax + rax]
0x1d418:    mov     rbx, qword ptr [rbx + 8]
0x1d41c:    test    rbx, rbx
0x1d41f:    je      0x1d448
0x1d421:    cmp     byte ptr [rbx + 0x10], r12b
0x1d425:    jne     0x1d418
0x1d427:    lea     rdi, [rbx + 0x10]
0x1d42b:    mov     rsi, rbp
0x1d42e:    call    0x4b30
0x1d433:    test    eax, eax
0x1d435:    jne     0x1d418
0x1d437:    xor     r12d, r12d
0x1d43a:    pop     rbx
0x1d43b:    pop     rbp
0x1d43c:    mov     rax, r12
0x1d43f:    pop     r12
0x1d441:    ret
0x1d442:    nop     word ptr [rax + rax]
0x1d448:    mov     rdi, rbp
0x1d44b:    call    0x4b60
0x1d450:    mov     rdi, rbp
0x1d453:    mov     r12, rax
0x1d456:    call    0x49b0
0x1d45b:    lea     rdi, [rax + 0x11]
0x1d45f:    call    0x1d670
0x1d464:    mov     rsi, rbp
0x1d467:    lea     rdi, [rax + 0x10]
0x1d46b:    mov     rbx, rax
0x1d46e:    call    0x48b0
0x1d473:    test    r12, r12
0x1d476:    je      0x1d4a0
0x1d478:    mov     eax, dword ptr [r12 + 0x10]
0x1d47d:    mov     dword ptr [rbx], eax
0x1d47f:    mov     rax, qword ptr [rip + 0xe6ba]
0x1d486:    mov     qword ptr [rip + 0xe6b3], rbx

```

```

0x1d48d:      mov     qword ptr [rbx + 8], rax
0x1d491:      jmp     0x1d3ef
0x1d3ef:
0x1d496:      nop     word ptr cs:[rax + rax]
0x1d4a0:      mov     rax, qword ptr [rip + 0xe691]
0x1d4a7:      mov     qword ptr [rip + 0xe68a], rbx
0x1d4ae:      mov     qword ptr [rbx + 8], rax
0x1d4b2:      mov     rax, r12
0x1d4b5:      pop     rbx
0x1d4b6:      pop     rbp
0x1d4b7:      pop     r12
0x1d4b9:      ret
0x1d4ba:      nop     word ptr [rax + rax]
0x1d4c0:      endbr64
0x1d4c4:      push    r12
0x1d4c6:      push    rbp
0x1d4c7:      mov     ebp, edi
0x1d4c9:      push    rbx
0x1d4ca:      mov     rbx, qword ptr [rip + 0xe65f]
0x1d4d1:      test    rbx, rbx
0x1d4d4:      jne     0x1d4e9
0x1d4d6:      jmp     0x1d500
0x1d500:
0x1d4d8:      nop     dword ptr [rax + rax]
0x1d4e0:      mov     rbx, qword ptr [rbx + 8]
0x1d4e4:      test    rbx, rbx
0x1d4e7:      je      0x1d500
0x1d4e9:      cmp     dword ptr [rbx], ebp
0x1d4eb:      jne     0x1d4e0
0x1d4ed:      xor     eax, eax
0x1d4ef:      cmp     byte ptr [rbx + 0x10], 0
0x1d4f3:      je      0x1d4f9
0x1d4f5:      lea     rax, [rbx + 0x10]
0x1d4f9:      pop     rbx
0x1d4fa:      pop     rbp
0x1d4fb:      pop     r12
0x1d4fd:      ret
0x1d4fe:      nop
0x1d500:      mov     edi, ebp
0x1d502:      lea     r12, [rip + 0x4edc]
0x1d509:      call    0x4a10
0x1d50e:      mov     edi, 0x11
0x1d513:      test    rax, rax
0x1d516:      je      0x1d527
0x1d518:      mov     r12, qword ptr [rax]
0x1d51b:      mov     rdi, r12
0x1d51e:      call    0x49b0
0x1d523:      lea     rdi, [rax + 0x11]

```

```

0x1d527:      call    0x1d670
0x1d52c:      mov     rsi, r12
0x1d52f:      mov     dword ptr [rax], ebp
0x1d531:      lea     rdi, [rax + 0x10]
0x1d535:      mov     rbx, rax
0x1d538:      call    0x48b0
0x1d53d:      mov     rax, qword ptr [rip + 0xe5ec]
0x1d544:      mov     qword ptr [rip + 0xe5e5], rbx
0x1d54b:      mov     qword ptr [rbx + 8], rax
0x1d54f:      jmp     0x1d4ed
0x1d4ed
0x1d551:      nop     word ptr cs:[rax + rax]
0x1d55c:      nop     dword ptr [rax]
0x1d560:      endbr64
0x1d564:      push    r12
0x1d566:      push    rbp
0x1d567:      mov     rbp, rdi
0x1d56a:      push    rbx
0x1d56b:      mov     rbx, qword ptr [rip + 0xe5be]
0x1d572:      test    rbx, rbx
0x1d575:      je      0x1d5b0
0x1d577:      movzx   r12d, byte ptr [rdi]
0x1d57b:      jmp     0x1d589
0x1d589
0x1d57d:      nop     dword ptr [rax]
0x1d580:      mov     rbx, qword ptr [rbx + 8]
0x1d584:      test    rbx, rbx
0x1d587:      je      0x1d5b0
0x1d589:      cmp     byte ptr [rbx + 0x10], r12b
0x1d58d:      jne     0x1d580
0x1d58f:      lea     rdi, [rbx + 0x10]
0x1d593:      mov     rsi, rbp
0x1d596:      call    0x4b30
0x1d59b:      test    eax, eax
0x1d59d:      jne     0x1d580
0x1d59f:      mov     r12, rbx
0x1d5a2:      pop     rbx
0x1d5a3:      pop     rbp
0x1d5a4:      mov     rax, r12
0x1d5a7:      pop     r12
0x1d5a9:      ret
0x1d5aa:      nop     word ptr [rax + rax]
0x1d5b0:      mov     rbx, qword ptr [rip + 0xe571]
0x1d5b7:      test    rbx, rbx
0x1d5ba:      je      0x1d5f8
0x1d5bc:      movzx   r12d, byte ptr [rbp]
0x1d5c1:      jmp     0x1d5d1
0x1d5d1

```



```

0x1d5c3:      nop      dword ptr [rax + rax]
0x1d5c8:      mov      rbx, qword ptr [rbx + 8]
0x1d5cc:      test     rbx, rbx
0x1d5cf:      je        0x1d5f8
0x1d5d1:      cmp      byte ptr [rbx + 0x10], r12b
0x1d5d5:      jne       0x1d5c8
0x1d5d7:      lea      rdi, [rbx + 0x10]
0x1d5db:      mov      rsi, rbp
0x1d5de:      call     0x4b30
0x1d5e3:      test     eax, eax
0x1d5e5:      jne       0x1d5c8
0x1d5e7:      xor      r12d, r12d
0x1d5ea:      pop      rbx
0x1d5eb:      pop      rbp
0x1d5ec:      mov      rax, r12
0x1d5ef:      pop      r12
0x1d5f1:      ret
0x1d5f2:      nop      word ptr [rax + rax]
0x1d5f8:      mov      rdi, rbp
0x1d5fb:      call     0x4bc0
0x1d600:      mov      rdi, rbp
0x1d603:      mov      r12, rax
0x1d606:      call     0x49b0
0x1d60b:      lea      rdi, [rax + 0x11]
0x1d60f:      call     0x1d670
0x1d614:      mov      rsi, rbp
0x1d617:      lea      rdi, [rax + 0x10]
0x1d61b:      mov      rbx, rax
0x1d61e:      call     0x48b0
0x1d623:      test     r12, r12
0x1d626:      je        0x1d650
0x1d628:      mov      eax, dword ptr [r12 + 0x10]
0x1d62d:      mov      dword ptr [rbx], eax
0x1d62f:      mov      rax, qword ptr [rip + 0xe4fa]
0x1d636:      mov      qword ptr [rip + 0xe4f3], rbx
0x1d63d:      mov      qword ptr [rbx + 8], rax
0x1d641:      jmp      0x1d59f
0x1d59f:
0x1d646:      nop      word ptr cs:[rax + rax]
0x1d650:      mov      rax, qword ptr [rip + 0xe4d1]
0x1d657:      mov      qword ptr [rip + 0xe4ca], rbx
0x1d65e:      mov      qword ptr [rbx + 8], rax
0x1d662:      mov      rax, r12
0x1d665:      pop      rbx
0x1d666:      pop      rbp
0x1d667:      pop      r12
0x1d669:      ret
0x1d66a:      nop      word ptr [rax + rax]

```

```

0x1d670:      endbr64
0x1d674:      push    rbx
0x1d675:      mov     rbx, rdi
0x1d678:      call    0x4c20
0x1d67d:      test    rax, rax
0x1d680:      jne     0x1d687
0x1d682:      test    rbx, rbx
0x1d685:      jne     0x1d689
0x1d687:      pop     rbx
0x1d688:      ret
0x1d689:      call    0x18cf0
0x1d68e:      nop
0x1d690:      endbr64
0x1d694:      push    rbx
0x1d695:      mov     rbx, rsi
0x1d698:      call    0x4ce0
0x1d69d:      test    rax, rax
0x1d6a0:      jne     0x1d6a7
0x1d6a2:      test    rbx, rbx
0x1d6a5:      jne     0x1d6a9
0x1d6a7:      pop     rbx
0x1d6a8:      ret
0x1d6a9:      call    0x18cf0
0x1d6ae:      nop
0x1d6b0:      endbr64
0x1d6b4:      push    rbx
0x1d6b5:      mov     rax, rsi
0x1d6b8:      mov     rsi, qword ptr [rsi]
0x1d6bb:      test    rdi, rdi
0x1d6be:      je      0x1d6f0
0x1d6c0:      movabs  rdx, 0xaaaaaaaaaaaaa9
0x1d6ca:      cmp     rsi, rdx
0x1d6cd:      ja      0x1d701
0x1d6cf:      lea     rdx, [rsi + 1]
0x1d6d3:      shr     rdx, 1
0x1d6d6:      add     rsi, rdx
0x1d6d9:      setne   bl
0x1d6dc:      mov     qword ptr [rax], rsi
0x1d6df:      call    0x4ce0
0x1d6e4:      test    rax, rax
0x1d6e7:      jne     0x1d6ed
0x1d6e9:      test    bl, bl
0x1d6eb:      jne     0x1d701
0x1d6ed:      pop     rbx
0x1d6ee:      ret
0x1d6ef:      nop
0x1d6f0:      mov     ebx, 1
0x1d6f5:      test    rsi, rsi

```

```

0x1d6f8:    jne     0x1d6dc
0x1d6fa:    mov     esi, 0x40
0x1d6ff:    jmp     0x1d6dc
0x1d6dc:
0x1d701:    call    0x18cf0
0x1d706:    nop     word ptr cs:[rax + rax]
0x1d710:    endbr64
0x1d714:    push    rbx
0x1d715:    mov     esi, 1
0x1d71a:    mov     rbx, rdi
0x1d71d:    call    0x4b10
0x1d722:    test    rax, rax
0x1d725:    jne     0x1d72c
0x1d727:    test    rbx, rbx
0x1d72a:    jne     0x1d72e
0x1d72c:    pop     rbx
0x1d72d:    ret
0x1d72e:    call    0x18cf0
0x1d733:    nop     word ptr cs:[rax + rax]
0x1d73e:    nop
0x1d740:    endbr64
0x1d744:    sub     rsp, 8
0x1d748:    call    0x4b10
0x1d74d:    test    rax, rax
0x1d750:    je      0x1d757
0x1d752:    add     rsp, 8
0x1d756:    ret
0x1d757:    call    0x18cf0
0x1d75c:    nop     dword ptr [rax]
0x1d760:    endbr64
0x1d764:    push    r12
0x1d766:    mov     r12, rsi
0x1d769:    push    rbp
0x1d76a:    mov     rbp, rdi
0x1d76d:    mov     rdi, rsi
0x1d770:    sub     rsp, 8
0x1d774:    call    0x4c20
0x1d779:    mov     rdi, rax
0x1d77c:    test    rax, rax
0x1d77f:    jne     0x1d786
0x1d781:    test    r12, r12
0x1d784:    jne     0x1d798
0x1d786:    add     rsp, 8
0x1d78a:    mov     rdx, r12
0x1d78d:    mov     rsi, rbp
0x1d790:    pop     rbp
0x1d791:    pop     r12
0x1d793:    jmp     0x4bb0

```

```

0x4bb0
0x1d798:      call    0x18cf0
0x1d79d:      nop     dword ptr [rax]
0x1d7a0:      endbr64
0x1d7a4:      push    r12
0x1d7a6:      push    rbp
0x1d7a7:      mov     rbp, rdi
0x1d7aa:      sub     rsp, 8
0x1d7ae:      call    0x49b0
0x1d7b3:      lea     r12, [rax + 1]
0x1d7b7:      mov     rdi, r12
0x1d7ba:      call    0x4c20
0x1d7bf:      test    rax, rax
0x1d7c2:      je      0x1d7d9
0x1d7c4:      add     rsp, 8
0x1d7c8:      mov     rdx, r12
0x1d7cb:      mov     rsi, rbp
0x1d7ce:      mov     rdi, rax
0x1d7d1:      pop     rbp
0x1d7d2:      pop     r12
0x1d7d4:      jmp     0x4bb0
0x4bb0
0x1d7d9:      call    0x18cf0
0x1d7de:      nop
0x1d7e0:      endbr64
0x1d7e4:      push    r15
0x1d7e6:      mov     r15, rsi
0x1d7e9:      push    r14
0x1d7eb:      mov     r14, rcx
0x1d7ee:      push    r13
0x1d7f0:      mov     r13d, edx
0x1d7f3:      push    r12
0x1d7f5:      mov     r12, r8
0x1d7f8:      push    rbp
0x1d7f9:      mov     rbp, rdi
0x1d7fc:      push    rbx
0x1d7fd:      sub     rsp, 0x28
0x1d801:      movzx   ebx, byte ptr [rdi]
0x1d804:      mov     rax, qword ptr fs:[0x28]
0x1d80d:      mov     qword ptr [rsp + 0x18], rax
0x1d812:      xor     eax, eax
0x1d814:      lea     rax, [rsp + 0x10]
0x1d819:      test    rsi, rsi
0x1d81c:      cmove   r15, rax
0x1d820:      call    0x4f00
0x1d825:      mov     rcx, qword ptr [rax]
0x1d828:      mov     rax, rbp
0x1d82b:      jmp     0x1d838

```

```

0x1d838
0x1d82d:    nop        dword ptr [rax]
0x1d830:    movzx     ebx, byte ptr [rax + 1]
0x1d834:    add       rax, 1
0x1d838:    movzx     edx, bl
0x1d83b:    test      byte ptr [rcx + rdx*2 + 1], 0x20
0x1d840:    jne       0x1d830
0x1d842:    cmp       bl, 0x2d
0x1d845:    je        0x1d8ca
0x1d84b:    call      0x4860
0x1d850:    mov       edx, r13d
0x1d853:    mov       rsi, r15
0x1d856:    mov       rdi, rbp
0x1d859:    mov       dword ptr [rax], 0
0x1d85f:    mov       qword ptr [rsp + 8], rax
0x1d864:    call      0x4dd0
0x1d869:    mov       r13, qword ptr [r15]
0x1d86c:    mov       rcx, qword ptr [rsp + 8]
0x1d871:    mov       rbx, rax
0x1d874:    cmp       r13, rbp
0x1d877:    je        0x1d8d8
0x1d879:    mov       eax, dword ptr [rcx]
0x1d87b:    test      eax, eax
0x1d87d:    jne       0x1d8c0
0x1d87f:    xor       ebp, ebp
0x1d881:    test      r12, r12
0x1d884:    je        0x1d893
0x1d886:    movzx     edx, byte ptr [r13]
0x1d88b:    test      dl, dl
0x1d88d:    jne       0x1d958
0x1d893:    mov       qword ptr [r14], rbx
0x1d896:    mov       rax, qword ptr [rsp + 0x18]
0x1d89b:    sub       rax, qword ptr fs:[0x28]
0x1d8a4:    jne       0x1dbf9
0x1d8aa:    add       rsp, 0x28
0x1d8ae:    mov       eax, ebp
0x1d8b0:    pop       rbx
0x1d8b1:    pop       rbp
0x1d8b2:    pop       r12
0x1d8b4:    pop       r13
0x1d8b6:    pop       r14
0x1d8b8:    pop       r15
0x1d8ba:    ret
0x1d8bb:    nop        dword ptr [rax + rax]
0x1d8c0:    mov       ebp, 1
0x1d8c5:    cmp       eax, 0x22
0x1d8c8:    je        0x1d881
0x1d8ca:    mov       ebp, 4

```

```

0x1d8cf:      jmp      0x1d896
0x1d896:
0x1d8d1:      nop      dword ptr [rax]
0x1d8d8:      test     r12, r12
0x1d8db:      je       0x1d8ca
0x1d8dd:      movzx     edx, byte ptr [rbp]
0x1d8e1:      test     dl, dl
0x1d8e3:      je       0x1d8ca
0x1d8e5:      movsx     esi, dl
0x1d8e8:      mov      rdi, r12
0x1d8eb:      mov      byte ptr [rsp + 8], dl
0x1d8ef:      xor      ebp, ebp
0x1d8f1:      call     0x4a00
0x1d8f6:      mov      ebx, 1
0x1d8fb:      movzx     edx, byte ptr [rsp + 8]
0x1d900:      test     rax, rax
0x1d903:      je       0x1d8ca
0x1d905:      mov      esi, 0x30
0x1d90a:      mov      rdi, r12
0x1d90d:      mov      byte ptr [rsp + 8], dl
0x1d911:      call     0x4a00
0x1d916:      movzx     edx, byte ptr [rsp + 8]
0x1d91b:      test     rax, rax
0x1d91e:      je       0x1d97c
0x1d920:      movzx     eax, byte ptr [r13 + 1]
0x1d925:      cmp      al, 0x44
0x1d927:      je       0x1d9a2
0x1d929:      cmp      al, 0x69
0x1d92b:      je       0x1dbe0
0x1d931:      cmp      al, 0x42
0x1d933:      je       0x1d9a2
0x1d935:      sub      edx, 0x42
0x1d938:      cmp      dl, 0x35
0x1d93b:      ja       0x1d971
0x1d93d:      lea      rcx, [rip + 0x4c0c]
0x1d944:      movzx     edx, dl
0x1d947:      movsxd    rax, dword ptr [rcx + rdx*4]
0x1d94b:      add      rax, rcx
0x1d94e:      jmp      rax
rax
->->->                                indirect JMP
0x1b9e0:      mov      esi, dword ptr [rsp + 0x6c]
0x1b9e4:      mov      r14, r15
0x1b9e7:      mov      r15, qword ptr [rsp + 0x28]
0x1b9ec:      mov      ecx, 3
0x1b9f1:      lea      edx, [rsi + 2]
0x1b9f4:      add      esi, 1
0x1b9f7:      mov      rdi, rbp

```

```

0x1b9fa:    xor     eax, eax
0x1b9fc:    call   0x19fe0
0x1ba01:    mov     rdi, qword ptr [rsp + 0x10]
0x1ba06:    mov     rbx, rax
0x1ba09:    call   0x18d10
0x1ba0e:    mov     rdi, r14
0x1ba11:    call   0x18d10
0x1ba16:    mov     rax, qword ptr [rsp + 0x30]
0x1ba1b:    mov     rdi, r15
0x1ba1e:    mov     qword ptr [rax], rbx
0x1ba21:    call   0x18d10
0x1ba26:    mov     eax, 0xffffffff
0x1ba2b:    cmp     qword ptr [rsp + 0x20], rbp
0x1ba30:    je      0x1b67c
0x1ba36:    mov     rdi, rbp
0x1ba39:    mov     dword ptr [rsp + 4], eax
0x1ba3d:    call   0x18d10
0x1ba42:    mov     eax, dword ptr [rsp + 4]
0x1ba46:    jmp     0x1b67c
0x1b67c:
0x1ba4b:    nop     dword ptr [rax + rax]
0x1ba50:    mov     esi, 0x2f
0x1ba55:    mov     rdi, r15
0x1ba58:    call   0x4a20
0x1ba5d:    lea     r13, [rax + 1]
0x1ba61:    test    rax, rax
0x1ba64:    je      0x1bb70
0x1ba6a:    mov     rdi, r13
0x1ba6d:    call   0x49b0
0x1ba72:    lea     rdi, [rax + 1]
0x1ba76:    call   0x1d670
0x1ba7b:    mov     rsi, r13
0x1ba7e:    mov     rdi, rax
0x1ba81:    call   0x48b0
0x1ba86:    mov     rdi, r15
0x1ba89:    mov     rbx, rax
0x1ba8c:    call   0x18d10
0x1ba91:    mov     eax, dword ptr [rsp + 0x6c]
0x1ba95:    mov     r15, rbx
0x1ba98:    lea     r14d, [rax + 2]
0x1ba9c:    nop     dword ptr [rax]
0x1baa0:    mov     dword ptr [rsp + 0x6c], r14d
0x1baa5:    movsxd  rcx, r14d
0x1baa8:    cmp     byte ptr [rbp + rcx], 0x3a
0x1baad:    je      0x1b9a0
0x1bab3:    xor     eax, eax
0x1bab5:    cmp     dword ptr [rsp + 0x38], 1
0x1baba:    mov     edx, r14d

```

```

0x1babd:    mov     r14, r15
0x1bac0:    sete   al
0x1bac3:    mov     r15, qword ptr [rsp + 0x28]
0x1bac8:    movsxd  rbx, dword ptr [rsp + 0x40]
0x1bacd:    mov     dword ptr [rsp + 0x28], eax
0x1bad1:    sub     edx, 1
0x1bad4:    mov     rdi, r14
0x1bad7:    mov     dword ptr [rsp + 0x6c], edx
0x1badb:    call    0x49b0
0x1bae0:    cmp     eax, 0x80
0x1bae5:    jg      0x1c269
0x1baeb:    mov     r12, qword ptr [rsp + 0x10]
0x1baf0:    mov     rsi, r14
0x1baf3:    mov     rdi, r12
0x1baf6:    call    0x48b0
0x1bafb:    mov     rdi, r14
0x1bafe:    call    0x18d10
0x1bb03:    add     dword ptr [rsp + 0x3c], 1
0x1bb08:    cmp     byte ptr [r12], 0
0x1bb0d:    mov     r13d, dword ptr [rsp + 0x6c]
0x1bb12:    jne     0x1c098
0x1bb18:    mov     rdi, qword ptr [rsp + 0x10]
0x1bb1d:    call    0x18d10
0x1bb22:    jmp     0x1b74d
0x1b74d:
0x1bb27:    nop     word ptr [rax + rax]
0x1bb30:    mov     esi, 0x2e
0x1bb35:    mov     rdi, r15
0x1bb38:    call    0x4a20
0x1bb3d:    test    rax, rax
0x1bb40:    je      0x1bb70
0x1bb42:    mov     byte ptr [rax], 0
0x1bb45:    mov     eax, dword ptr [rsp + 0x6c]
0x1bb49:    lea     r14d, [rax + 2]
0x1bb4d:    jmp     0x1baa0
0x1baa0:
0x1bb52:    nop     word ptr [rax + rax]
0x1bb58:    mov     esi, 0x2f
0x1bb5d:    mov     rdi, r15
0x1bb60:    call    0x4a20
0x1bb65:    test    rax, rax
0x1bb68:    jne     0x1bb42
0x1bb6a:    nop     word ptr [rax + rax]
0x1bb70:    mov     eax, dword ptr [rsp + 0x6c]
0x1bb74:    lea     r14d, [rax + 2]
0x1bb78:    jmp     0x1baa0
0x1baa0:
0x1bb7d:    nop     dword ptr [rax]

```



```

0x1bb80:    mov     esi, 0x2e
0x1bb85:    mov     rdi, r15
0x1bb88:    call    0x4a20
0x1bb8d:    mov     r13, rax
0x1bb90:    test    rax, rax
0x1bb93:    jne     0x1ba6a
0x1bb99:    jmp     0x1bb70
0x1bb70:
0x1bb9b:    nop     dword ptr [rax + rax]
0x1bba0:    mov     eax, dword ptr [rsp + 0x6c]
0x1bba4:    add     dword ptr [rsp + 0x38], 1
0x1bba9:    lea     r14d, [rax + 2]
0x1bbad:    jmp     0x1baa0
0x1baa0:
0x1bbb2:    nop     word ptr [rax + rax]
0x1bbb8:    mov     ebx, dword ptr [rsp + 0x6c]
0x1bbbc:    mov     rdi, rbp
0x1bbbf:    lea     r14d, [rbx + 2]
0x1bbc3:    call    0x49b0
0x1bbc8:    cmp     r14d, eax
0x1bbcb:    jge     0x1baa0
0x1bbd1:    movsxd  rax, ebx
0x1bbd4:    lea     r14, [rsp + 0x6c]
0x1bbd9:    add     ebx, 3
0x1bbdc:    xor     ecx, ecx
0x1bbde:    movsx   r13d, byte ptr [rbp + rax + 2]
0x1bbe4:    lea     r8, [rip + 0xff21]
0x1bbeb:    mov     rsi, r14
0x1bbee:    mov     rdi, rbp
0x1bbf1:    xor     eax, eax
0x1bbf3:    mov     dword ptr [rsp + 0x6c], ebx
0x1bbf7:    mov     edx, r13d
0x1bbfa:    call    0x19c70
0x1bbff:    mov     rbx, rax
0x1bc02:    test    rax, rax
0x1bc05:    je      0x1c387
0x1bc0b:    mov     rdi, qword ptr [rip + 0xff06]
0x1bc12:    test    rdi, rdi
0x1bc15:    je      0x1bc1c
0x1bc17:    call    0x18d10
0x1bc1c:    mov     qword ptr [rip + 0xfef5], rbx
0x1bc23:    mov     eax, dword ptr [rip + 0xfef3]
0x1bc29:    test    eax, eax
0x1bc2b:    je      0x1c400
0x1bc31:    mov     rdi, qword ptr [rip + 0xfed8]
0x1bc38:    test    rdi, rdi
0x1bc3b:    je      0x1bc42
0x1bc3d:    call    0x18d10

```

```

0x1bc42:    mov     rsi, r14
0x1bc45:    mov     rdi, rbp
0x1bc48:    xor     eax, eax
0x1bc4a:    mov     ecx, 1
0x1bc4f:    lea     r8, [rip + 0xfeb2]
0x1bc56:    mov     edx, r13d
0x1bc59:    call    0x19c70
0x1bc5e:    mov     esi, 0x26
0x1bc63:    mov     rdi, rax
0x1bc66:    mov     qword ptr [rip + 0xfea3], rax
0x1bc6d:    call    0x4a00
0x1bc72:    test    rax, rax
0x1bc75:    je      0x1be75
0x1bc7b:    mov     edx, dword ptr [rip + 0xfe8b]
0x1bc81:    add     edx, dword ptr [rip + 0xfe81]
0x1bc87:    movsxd  rdi, edx
0x1bc8a:    mov     dword ptr [rsp + 0x48], edx
0x1bc8e:    call    0x1d670
0x1bc93:    mov     r9d, dword ptr [rip + 0xfe6e]
0x1bc9a:    mov     r10, qword ptr [rip + 0xfe6f]
0x1bca1:    mov     r8, rax
0x1bca4:    test    r9d, r9d
0x1bca7:    jle     0x1c2d0
0x1bcad:    mov     qword ptr [rsp + 0x50], rbp
0x1bcb2:    xor     ebx, ebx
0x1bcb4:    xor     r13d, r13d
0x1bcb7:    mov     rbp, rax
0x1bcba:    mov     qword ptr [rsp + 0x58], r15
0x1bcbf:    mov     r15d, dword ptr [rsp + 0x48]
0x1bcc4:    jmp     0x1bd01
0x1bd01:
0x1bcc6:    nop     word ptr cs:[rax + rax]
0x1bcd0:    cmp     al, 0x5c
0x1bcd2:    je      0x1c300
0x1bcd8:    cmp     r15d, ebx
0x1bcdb:    jle     0x1c320
0x1bce1:    movzx   ecx, byte ptr [r10 + r14]
0x1bce6:    movsxd  rax, ebx
0x1bce9:    add     ebx, 1
0x1bcec:    mov     byte ptr [rbp + rax], cl
0x1bcf0:    add     r13d, 1
0x1bcf4:    cmp     r13d, dword ptr [rip + 0xfe0d]
0x1bcfb:    jge     0x1c36f
0x1bd01:    movsxd  r14, r13d
0x1bd04:    movzx   eax, byte ptr [r10 + r14]
0x1bd09:    cmp     al, 0x26
0x1bd0b:    jne     0x1bcd0
0x1bd0d:    mov     eax, dword ptr [rip + 0xfdf9]

```

```

0x1bd13:    lea     ecx, [rax + rbx]
0x1bd16:    cmp     ecx, r15d
0x1bd19:    jge     0x1c33d
0x1bd1f:    movsxd  rdi, ebx
0x1bd22:    mov     rsi, qword ptr [rip + 0xfdef]
0x1bd29:    mov     qword ptr [rsp + 0x48], r10
0x1bd2e:    add     rdi, rbp
0x1bd31:    call    0x48b0
0x1bd36:    mov     r10, qword ptr [rsp + 0x48]
0x1bd3b:    add     ebx, dword ptr [rip + 0xfdc3]
0x1bd41:    jmp     0x1bcf0
0x1bcf0:
0x1bd43:    nop     dword ptr [rax + rax]
0x1bd48:    cmp     al, 0x39
0x1bd4a:    jg      0x1bd80
0x1bd4c:    cmp     al, 0x2f
0x1bd4e:    jg      0x1c150
0x1bd54:    mov     rdi, r14
0x1bd57:    call    0x49b0
0x1bd5c:    lea     rdi, [rax + 1]
0x1bd60:    call    0x1d670
0x1bd65:    mov     rsi, r14
0x1bd68:    mov     rdi, rax
0x1bd6b:    call    0x48b0
0x1bd70:    mov     r14, rax
0x1bd73:    jmp     0x1b963
0x1b963:
0x1bd78:    nop     dword ptr [rax + rax]
0x1bd80:    cmp     al, 0x5e
0x1bd82:    jne     0x1bd54
0x1bd84:    add     edx, 1
0x1bd87:    mov     edi, 1
0x1bd8c:    mov     esi, 1
0x1bd91:    mov     dword ptr [rsp + 0x6c], edx
0x1bd95:    nop     dword ptr [rax]
0x1bd98:    mov     rdx, r14
0x1bd9b:    xor     eax, eax
0x1bd9d:    call    0x1b420
0x1bda2:    mov     r13, rax
0x1bda5:    test    rax, rax
0x1bda8:    jne     0x1b92c
0x1bdae:    mov     r8d, dword ptr [rsp + 0x6c]
0x1bdb3:    mov     ecx, 1
0x1bdb8:    mov     edx, r8d
0x1bdbb:    mov     esi, r12d
0x1bdbe:    mov     rdi, rbp
0x1bdc1:    xor     eax, eax
0x1bdc3:    call    0x19fe0

```

```

0x1bdc8:      mov     rdi, qword ptr [rsp + 0x10]
0x1bdcd:      mov     rbx, rax
0x1bdd0:      call    0x18d10
0x1bdd5:      jmp     0x1ba16
0x1ba16
0x1bdda:      nop     word ptr [rax + rax]
0x1bde0:      cmp     ebx, dword ptr [rsp + 4]
0x1bde4:      jge     0x1bff0
0x1bdea:      lea     eax, [rbx + 1]
0x1bded:      mov     byte ptr [r15 + rbx], 0x5c
0x1bdf2:      movsxd  rdx, eax
0x1bdf5:      mov     dword ptr [rsp + 8], 1
0x1bdfd:      movsxd  rbx, eax
0x1be00:      mov     byte ptr [r15 + rdx], 0
0x1be05:      jmp     0x1b74d
0x1b74d
0x1be0a:      nop     word ptr [rax + rax]
0x1be10:      add     dword ptr [rsp + 4], 0x40
0x1be15:      movsxd  rsi, dword ptr [rsp + 4]
0x1be1a:      mov     rdi, r15
0x1be1d:      call    0x1d690
0x1be22:      mov     r15, rax
0x1be25:      jmp     0x1b778
0x1b778
0x1be2a:      nop     word ptr [rax + rax]
0x1be30:      xor     edx, edx
0x1be32:      test    r13d, r13d
0x1be35:      je      0x1be4d
0x1be37:      movzx   eax, byte ptr [rbp + r14 - 1]
0x1be3d:      cmp     al, 0x27
0x1be3f:      je      0x1c078
0x1be45:      cmp     al, 0x22
0x1be47:      je      0x1c078
0x1be4d:      lea     rsi, [rsp + 0x6c]
0x1be52:      mov     rdi, rbp
0x1be55:      xor     eax, eax
0x1be57:      call    0x1ac70
0x1be5c:      mov     r12d, dword ptr [rsp + 0x6c]
0x1be61:      mov     r14, rax
0x1be64:      jmp     0x1b8b3
0x1b8b3
0x1be69:      nop     dword ptr [rax]
0x1be70:      add     dword ptr [rsp + 0x6c], 2
0x1be75:      mov     rdi, r15
0x1be78:      call    0x49b0
0x1be7d:      mov     r13d, dword ptr [rip + 0xfc88]
0x1be84:      cmp     eax, r13d
0x1be87:      jl      0x1c400

```

```

0x1be8d:    mov     qword ptr [rsp + 0x48], rbp
0x1be92:    xor     r14d, r14d
0x1be95:    mov     rbp, r15
0x1be98:    mov     r15d, eax
0x1be9b:    mov     dword ptr [rsp + 0x50], 1
0x1bea3:    jmp     0x1beba
0x1beba:
0x1bea5:    nop     dword ptr [rax]
0x1bea8:    add     r14d, 1
0x1beac:    lea     eax, [r13 + r14]
0x1beb1:    cmp     eax, r15d
0x1beb4:    jg      0x1c050
0x1beba:    mov     rsi, qword ptr [rip + 0xfc57]
0x1bec1:    movsxd  rbx, r14d
0x1bec4:    lea     rdi, [rbp + rbx]
0x1bec9:    movzx   eax, byte ptr [rsi]
0x1becc:    cmp     byte ptr [rdi], al
0x1bece:    jne     0x1bea8
0x1bed0:    movsxd  rdx, r13d
0x1bed3:    call    0x4890
0x1bed8:    test    eax, eax
0x1beda:    jne     0x1bea8
0x1bedc:    mov     ecx, dword ptr [rip + 0xfc26]
0x1bee2:    add     ecx, r15d
0x1bee5:    sub     ecx, r13d
0x1bee8:    lea     edi, [rcx + 1]
0x1beeb:    mov     dword ptr [rsp + 0x50], ecx
0x1beef:    movsxd  rdi, edi
0x1bef2:    call    0x1d670
0x1bef7:    mov     rdx, rbx
0x1befa:    mov     rsi, rbp
0x1befd:    mov     rdi, rax
0x1bf00:    mov     r13, rax
0x1bf03:    call    0x4880
0x1bf08:    movsxd  rdx, dword ptr [rip + 0xfbf9]
0x1bf0f:    lea     rdi, [r13 + rbx]
0x1bf14:    mov     rsi, qword ptr [rip + 0xfbf5]
0x1bf1b:    call    0x4880
0x1bf20:    movsxd  rsi, dword ptr [rip + 0xfbe5]
0x1bf27:    movsxd  rdi, dword ptr [rip + 0xfbda]
0x1bf2e:    lea     edx, [rsi + r14]
0x1bf32:    add     rdi, rbx
0x1bf35:    add     rsi, rbx
0x1bf38:    sub     r15d, edx
0x1bf3b:    add     rsi, rbp
0x1bf3e:    add     rdi, r13
0x1bf41:    movsxd  rdx, r15d
0x1bf44:    call    0x4880

```

```

0x1bf49:      movsxd  rcx, dword ptr [rsp + 0x50]
0x1bf4e:      mov     rdi, rbp
0x1bf51:      mov     byte ptr [r13 + rcx], 0
0x1bf57:      call    0x18d10
0x1bf5c:      mov     edi, dword ptr [rsp + 0x18]
0x1bf60:      test    edi, edi
0x1bf62:      jne     0x1c1a8
0x1bf68:      mov     rbp, qword ptr [rsp + 0x48]
0x1bf6d:      mov     r14d, dword ptr [rsp + 0x6c]
0x1bf72:      mov     r15, r13
0x1bf75:      jmp     0x1baa5
0x1baa5:
0x1bf7a:      nop     word ptr [rax + rax]
0x1bf80:      add     edx, 1
0x1bf83:      mov     esi, 0x24
0x1bf88:      mov     edi, 0x24
0x1bf8d:      xor     eax, eax
0x1bf8f:      mov     dword ptr [rsp + 0x6c], edx
0x1bf93:      mov     rdx, r14
0x1bf96:      call    0x1b420
0x1bf9b:      mov     r13, rax
0x1bf9e:      lea     rax, [rip + 0xf7b]
0x1bfa5:      cmp     r13, rax
0x1bfa8:      je      0x1bdae
0x1bfae:      test    r13, r13
0x1bfb1:      jne     0x1b93c
0x1bfb7:      jmp     0x1bd54
0x1bd54:
0x1bfbc:      nop     dword ptr [rax]
0x1bfc0:      cmp     r12b, 0x23
0x1bfc4:      jne     0x1be30
0x1bfca:      lea     r12d, [r13 + 2]
0x1bfce:      mov     r14, r15
0x1bfd1:      mov     dword ptr [rsp + 0x6c], r12d
0x1bfd6:      jmp     0x1b8b3
0x1b8b3:
0x1bfdb:      nop     dword ptr [rax + rax]
0x1bfe0:      mov     qword ptr [rsp + 0x10], rdx
0x1bfe5:      jmp     0x1b806
0x1b806:
0x1bfea:      nop     word ptr [rax + rax]
0x1bff0:      add     dword ptr [rsp + 4], 0x40
0x1bff5:      movsxd  rsi, dword ptr [rsp + 4]
0x1bffa:      mov     rdi, r15
0x1bffd:      call    0x1d690
0x1c002:      mov     r15, rax
0x1c005:      jmp     0x1bdea
0x1bdea:

```

```

0x1c00a:      nop      word ptr [rax + rax]
0x1c010:      mov      r13, qword ptr [rip + 0xfad9]
0x1c017:      add      edx, 1
0x1c01a:      mov      dword ptr [rsp + 0x6c], edx
0x1c01e:      test     r13, r13
0x1c021:      je       0x1c080
0x1c023:      mov      rdi, r13
0x1c026:      call     0x49b0
0x1c02b:      lea      rdi, [rax + 1]
0x1c02f:      call     0x1d670
0x1c034:      mov      rsi, r13
0x1c037:      mov      rdi, rax
0x1c03a:      call     0x48b0
0x1c03f:      mov      r13, rax
0x1c042:      jmp      0x1b92c
0x1b92c
0x1c047:      nop      word ptr [rax + rax]
0x1c050:      cmp      dword ptr [rsp + 0x18], 1
0x1c055:      mov      r15, rbp
0x1c058:      mov      r14d, dword ptr [rsp + 0x6c]
0x1c05d:      mov      rbp, qword ptr [rsp + 0x48]
0x1c062:      jle      0x1c417
0x1c068:      mov      dword ptr [rsp + 0x18], 0
0x1c070:      jmp      0x1baa5
0x1baa5
0x1c075:      nop      dword ptr [rax]
0x1c078:      movsx    edx, al
0x1c07b:      jmp      0x1be4d
0x1be4d
0x1c080:      mov      edi, 1
0x1c085:      call     0x1d670
0x1c08a:      mov      byte ptr [rax], 0
0x1c08d:      mov      r13, rax
0x1c090:      jmp      0x1b92c
0x1b92c
0x1c095:      nop      dword ptr [rax]
0x1c098:      mov      rdi, qword ptr [rsp + 0x10]
0x1c09d:      call     0x49b0
0x1c0a2:      lea      r12d, [rbx + rax]
0x1c0a6:      mov      eax, dword ptr [rsp + 4]
0x1c0aa:      cmp      r12d, eax
0x1c0ad:      jl       0x1c0e0
0x1c0af:      movsxd   r14, eax
0x1c0b2:      mov      rdi, r15
0x1c0b5:      mov      r15d, eax
0x1c0b8:      sub      r14, -0x80
0x1c0bc:      nop      dword ptr [rax]
0x1c0c0:      mov      rsi, r14

```

```

0x1c0c3:      sub     r15d, -0x80
0x1c0c7:      sub     r14, -0x80
0x1c0cb:      call    0x1d690
0x1c0d0:      mov     rdi, rax
0x1c0d3:      cmp     r12d, r15d
0x1c0d6:      jge     0x1c0c0
0x1c0d8:      mov     dword ptr [rsp + 4], r15d
0x1c0dd:      mov     r15, rax
0x1c0e0:      movsxd  rdi, ebx
0x1c0e3:      mov     rsi, qword ptr [rsp + 0x10]
0x1c0e8:      movsxd  rbx, r12d
0x1c0eb:      add     rdi, r15
0x1c0ee:      call    0x48b0
0x1c0f3:      jmp     0x1bb18
0x1bb18
0x1c0f8:      nop     dword ptr [rax + rax]
0x1c100:      xor     edi, edi
0x1c102:      lea     r8d, [rdx + 1]
0x1c106:      movsxd  rax, r8d
0x1c109:      movzx   eax, byte ptr [rbp + rax]
0x1c10e:      lea     ecx, [rax - 0x30]
0x1c111:      cmp     cl, 9
0x1c114:      jbe     0x1c283
0x1c11a:      cmp     al, 0x24
0x1c11c:      je      0x1c3e5
0x1c122:      mov     dword ptr [rsp + 0x6c], r8d
0x1c127:      test    al, al
0x1c129:      je      0x1c3f6
0x1c12f:      cmp     al, 0x3a
0x1c131:      je      0x1c3f6
0x1c137:      test    edi, edi
0x1c139:      jne     0x1bdb3
0x1c13f:      xor     esi, esi
0x1c141:      jmp     0x1bd98
0x1bd98
0x1c146:      nop     word ptr cs:[rax + rax]
0x1c150:      test    esi, esi
0x1c152:      je      0x1bd54
0x1c158:      add     edx, 1
0x1c15b:      xor     edi, edi
0x1c15d:      movsxd  rcx, edx
0x1c160:      sub     eax, 0x30
0x1c163:      lea     edx, [rdi + rdi*4]
0x1c166:      movsx   eax, al
0x1c169:      lea     edi, [rax + rdx*2]
0x1c16c:      movzx   eax, byte ptr [rbp + rcx]
0x1c171:      mov     edx, ecx
0x1c173:      add     rcx, 1

```



```

0x1c177:    lea     r8d, [rax - 0x30]
0x1c17b:    cmp     r8b, 9
0x1c17f:    jbe     0x1c160
0x1c181:    cmp     al, 0x5e
0x1c183:    je      0x1c478
0x1c189:    cmp     al, 0x2a
0x1c18b:    je      0x1c250
0x1c191:    cmp     al, 0x2d
0x1c193:    je      0x1c102
0x1c199:    mov     dword ptr [rsp + 0x6c], edx
0x1c19d:    mov     esi, edi
0x1c19f:    jmp     0x1bd98
0x1bd98:
0x1c1a4:    nop     dword ptr [rax]
0x1c1a8:    mov     rdi, r13
0x1c1ab:    mov     rbp, r13
0x1c1ae:    add     r14d, dword ptr [rip + 0xf953]
0x1c1b5:    call    0x49b0
0x1c1ba:    add     dword ptr [rsp + 0x18], 1
0x1c1bf:    mov     r13d, dword ptr [rip + 0xf946]
0x1c1c6:    mov     dword ptr [rsp + 0x50], 0
0x1c1ce:    mov     r15d, eax
0x1c1d1:    jmp     0x1bea8
0x1bea8:
0x1c1d6:    nop     word ptr cs:[rax + rax]
0x1c1e0:    lea     edi, [r12 + 5]
0x1c1e5:    movsxd  rdi, edi
0x1c1e8:    call    0x1d670
0x1c1ed:    mov     r11d, 0x733a
0x1c1f3:    mov     rsi, qword ptr [rsp + 0x20]
0x1c1f8:    mov     rbp, rax
0x1c1fb:    movzx   eax, byte ptr [rip + 0xe5eb]
0x1c202:    mov     word ptr [rbp + 2], r11w
0x1c207:    lea     rdi, [rbp + 4]
0x1c20b:    mov     byte ptr [rbp + 1], al
0x1c20e:    mov     byte ptr [rbp], al
0x1c211:    call    0x48b0
0x1c216:    lea     eax, [r12 + 4]
0x1c21b:    mov     dword ptr [rsp + 0xc], eax
0x1c21f:    test    eax, eax
0x1c221:    jg      0x1b6e6
0x1c227:    mov     rax, qword ptr [rsp + 0x30]
0x1c22c:    mov     qword ptr [rax], r15
0x1c22f:    cmp     qword ptr [rsp + 0x20], rbp
0x1c234:    je      0x1b67a
0x1c23a:    mov     rdi, rbp
0x1c23d:    call    0x18d10
0x1c242:    mov     dword ptr [rsp + 0x3c], 0

```

```

0x1c24a:      jmp      0x1b7c0
0x1b7c0
0x1c24f:      nop
0x1c250:      cmp      al, 0x5e
0x1c252:      je       0x1c478
0x1c258:      add      edx, 1
0x1c25b:      mov      esi, 0x24
0x1c260:      mov      dword ptr [rsp + 0x6c], edx
0x1c264:      jmp      0x1bd98
0x1bd98
0x1c269:      add      eax, 1
0x1c26c:      mov      rdi, qword ptr [rsp + 0x10]
0x1c271:      movsxd   rsi, eax
0x1c274:      call     0x1d690
0x1c279:      mov      qword ptr [rsp + 0x10], rax
0x1c27e:      jmp      0x1baeb
0x1baeb
0x1c283:      add      edx, 2
0x1c286:      xor      esi, esi
0x1c288:      movsxd   rdx, edx
0x1c28b:      nop      dword ptr [rax + rax]
0x1c290:      sub      eax, 0x30
0x1c293:      lea      ecx, [rsi + rsi*4]
0x1c296:      mov      r8d, edx
0x1c299:      movsx    eax, al
0x1c29c:      lea      esi, [rax + rcx*2]
0x1c29f:      movzx    eax, byte ptr [rbp + rdx]
0x1c2a4:      add      rdx, 1
0x1c2a8:      lea      ecx, [rax - 0x30]
0x1c2ab:      cmp      cl, 9
0x1c2ae:      jbe      0x1c290
0x1c2b0:      cmp      esi, 0x24
0x1c2b3:      sete     al
0x1c2b6:      cmp      esi, edi
0x1c2b8:      setge    dl
0x1c2bb:      or       eax, edx
0x1c2bd:      mov      dword ptr [rsp + 0x6c], r8d
0x1c2c2:      test     al, al
0x1c2c4:      je       0x1bdb3
0x1c2ca:      jmp      0x1bd98
0x1bd98
0x1c2cf:      nop
0x1c2d0:      xor      ebx, ebx
0x1c2d2:      mov      byte ptr [rax], 0
0x1c2d5:      mov      rdi, r10
0x1c2d8:      mov      qword ptr [rsp + 0x48], r8
0x1c2dd:      call     0x18d10
0x1c2e2:      mov      r8, qword ptr [rsp + 0x48]

```

```

0x1c2e7:    mov     dword ptr [rip + 0xf81b], ebx
0x1c2ed:    mov     qword ptr [rip + 0xf81c], r8
0x1c2f4:    jmp     0x1be75
0x1be75:
0x1c2f9:    nop     dword ptr [rax]
0x1c300:    cmp     byte ptr [r10 + r14 + 1], 0x26
0x1c306:    lea     rax, [r14 + 1]
0x1c30a:    jne     0x1bcd8
0x1c310:    add     r13d, 1
0x1c314:    mov     r14, rax
0x1c317:    jmp     0x1bcd8
0x1bcd8:
0x1c31c:    nop     dword ptr [rax]
0x1c320:    add     r15d, r15d
0x1c323:    mov     rdi, rbp
0x1c326:    movsxd  rsi, r15d
0x1c329:    call    0x1d690
0x1c32e:    mov     r10, qword ptr [rip + 0xf7db]
0x1c335:    mov     rbp, rax
0x1c338:    jmp     0x1bce1
0x1bce1:
0x1c33d:    lea     r15d, [rax + r15*2]
0x1c341:    mov     rdi, rbp
0x1c344:    movsxd  rsi, r15d
0x1c347:    call    0x1d690
0x1c34c:    mov     r10, qword ptr [rip + 0xf7bd]
0x1c353:    mov     rbp, rax
0x1c356:    jmp     0x1bd1f
0x1bd1f:
0x1c35b:    xor     eax, eax
0x1c35d:    mov     rdi, r15
0x1c360:    call    0x1a380
0x1c365:    mov     eax, 2
0x1c36a:    jmp     0x1b67c
0x1b67c:
0x1c36f:    mov     r8, rbp
0x1c372:    movsxd  rax, ebx
0x1c375:    mov     r15, qword ptr [rsp + 0x58]
0x1c37a:    mov     rbp, qword ptr [rsp + 0x50]
0x1c37f:    add     rax, r8
0x1c382:    jmp     0x1c2d2
0x1c2d2:
0x1c387:    cmp     qword ptr [rip + 0xf789], 0
0x1c38f:    mov     eax, dword ptr [rip + 0xf777]
0x1c395:    jne     0x1bc29
0x1c39b:    mov     rbx, qword ptr [rip + 0xf74e]
0x1c3a2:    test    rbx, rbx
0x1c3a5:    je      0x1c3b0

```

```

0x1c3a7:    cmp     byte ptr [rbx], 0
0x1c3aa:    jne     0x1c442
0x1c3b0:    mov     dword ptr [rip + 0xf752], 0
0x1c3ba:    mov     r14, r15
0x1c3bd:    mov     r15, qword ptr [rsp + 0x28]
0x1c3c2:    mov     edx, dword ptr [rsp + 0x6c]
0x1c3c6:    mov     ecx, 2
0x1c3cb:    mov     esi, dword ptr [rsp + 0x44]
0x1c3cf:    jmp     0x1b9f7
0x1b9f7
0x1c3d4:    mov     dword ptr [rsp + 0x28], 0
0x1c3dc:    mov     edx, dword ptr [rsp + 0x44]
0x1c3e0:    jmp     0x1bad1
0x1bad1
0x1c3e5:    add     edx, 2
0x1c3e8:    mov     esi, 0x24
0x1c3ed:    mov     dword ptr [rsp + 0x6c], edx
0x1c3f1:    jmp     0x1bd98
0x1bd98
0x1c3f6:    mov     esi, 0xffffffff
0x1c3fb:    jmp     0x1bd98
0x1bd98
0x1c400:    mov     r14, r15
0x1c403:    mov     r15, qword ptr [rsp + 0x28]
0x1c408:    jmp     0x1c3c2
0x1c3c2
0x1c40a:    mov     rax, qword ptr [rsp + 0x30]
0x1c40f:    mov     qword ptr [rax], r15
0x1c412:    jmp     0x1b67a
0x1b67a
0x1c417:    mov     esi, dword ptr [rsp + 0x50]
0x1c41b:    test    esi, esi
0x1c41d:    je      0x1baa5
0x1c423:    mov     edx, r14d
0x1c426:    mov     ecx, 2
0x1c42b:    mov     r14, r15
0x1c42e:    mov     r15, qword ptr [rsp + 0x28]
0x1c433:    jmp     0x1c3cb
0x1c3cb
0x1c435:    xor     ecx, ecx
0x1c437:    mov     edx, r12d
0x1c43a:    mov     esi, r13d
0x1c43d:    jmp     0x1bdbbe
0x1bdbbe
0x1c442:    mov     rdi, rbx
0x1c445:    call    0x49b0
0x1c44a:    lea     rdi, [rax + 1]
0x1c44e:    call    0x1d670

```

```

0x1c453:    mov     rsi, rbx
0x1c456:    mov     rdi, rax
0x1c459:    call    0x48b0
0x1c45e:    mov     rdi, rax
0x1c461:    mov     qword ptr [rip + 0xf6b0], rax
0x1c468:    call    0x49b0
0x1c46d:    mov     dword ptr [rip + 0xf699], eax
0x1c473:    jmp     0x1bc29
0x1bc29:
0x1c478:    cmp     edi, 1
0x1c47b:    lea     r8d, [rdx + 1]
0x1c47f:    setle   al
0x1c482:    jmp     0x1c2bd
0x1c2bd:
0x1c487:    call    0x49d0
0x1c48c:    nop     dword ptr [rax]
0x1c490:    endbr64
0x1c494:    push    rbx
0x1c495:    mov     rbx, rdx
0x1c498:    add     rsp, -0x80
0x1c49c:    mov     rax, qword ptr fs:[0x28]
0x1c4a5:    mov     qword ptr [rsp + 0x78], rax
0x1c4aa:    xor     eax, eax
0x1c4ac:    mov     rsi, rsp
0x1c4af:    call    0x4cb0
0x1c4b4:    test    eax, eax
0x1c4b6:    js      0x1c514
0x1c4b8:    mov     rax, qword ptr [rsp + 8]
0x1c4bd:    test    rax, rax
0x1c4c0:    cmov     rax, qword ptr [rsp]
0x1c4c5:    mov     qword ptr [rbx], rax
0x1c4c8:    mov     rax, qword ptr [rsp + 0x10]
0x1c4cd:    mov     qword ptr [rbx + 8], rax
0x1c4d1:    mov     rax, qword ptr [rsp + 0x18]
0x1c4d6:    mov     qword ptr [rbx + 0x10], rax
0x1c4da:    mov     rax, qword ptr [rsp + 0x20]
0x1c4df:    mov     qword ptr [rbx + 0x18], rax
0x1c4e3:    shr     rax, 0x3f
0x1c4e7:    mov     byte ptr [rbx + 0x20], al
0x1c4ea:    mov     rax, qword ptr [rsp + 0x28]
0x1c4ef:    mov     qword ptr [rbx + 0x28], rax
0x1c4f3:    mov     rax, qword ptr [rsp + 0x30]
0x1c4f8:    mov     qword ptr [rbx + 0x30], rax
0x1c4fc:    xor     eax, eax
0x1c4fe:    mov     rdx, qword ptr [rsp + 0x78]
0x1c503:    sub     rdx, qword ptr fs:[0x28]
0x1c50c:    jne     0x1c51b
0x1c50e:    sub     rsp, -0x80

```

```

0x1c512:    pop     rbx
0x1c513:    ret
0x1c514:    mov     eax, 0xffffffff
0x1c519:    jmp     0x1c4fe
0x1c4fe
0x1c51b:    call    0x49d0
0x1c520:    endbr64
0x1c524:    push    r15
0x1c526:    push    r14
0x1c528:    push    r13
0x1c52a:    mov     r13, r8
0x1c52d:    push    r12
0x1c52f:    mov     r12, rcx
0x1c532:    push    rbp
0x1c533:    push    rbx
0x1c534:    mov     rbx, rdi
0x1c537:    sub     rsp, 0x98
0x1c53e:    mov     qword ptr [rsp], rsi
0x1c542:    mov     qword ptr [rsp + 0x18], r8
0x1c547:    mov     dword ptr [rsp + 0x48], edx
0x1c54b:    mov     rax, qword ptr fs:[0x28]
0x1c554:    mov     qword ptr [rsp + 0x88], rax
0x1c55c:    mov     eax, edx
0x1c55e:    and     edx, 0x20
0x1c561:    mov     dword ptr [rsp + 0x30], edx
0x1c565:    and     eax, 3
0x1c568:    cmp     edx, 1
0x1c56b:    mov     dword ptr [rsp + 0x10], eax
0x1c56f:    sbb     eax, eax
0x1c571:    and     eax, 0xffffffffe8
0x1c574:    add     eax, 0x400
0x1c579:    mov     dword ptr [rsp + 0x34], eax
0x1c57d:    call    0x4900
0x1c582:    mov     r15, qword ptr [rax]
0x1c585:    mov     r14, rax
0x1c588:    mov     rdi, r15
0x1c58b:    call    0x49b0
0x1c590:    mov     rcx, qword ptr [r14 + 0x10]
0x1c594:    mov     r14, qword ptr [r14 + 8]
0x1c598:    mov     rbp, rax
0x1c59b:    sub     rax, 1
0x1c59f:    cmp     rax, 0x10
0x1c5a3:    mov     eax, 1
0x1c5a8:    mov     rdi, r14
0x1c5ab:    mov     qword ptr [rsp + 0x38], rcx
0x1c5b0:    cmovae  rbp, rax
0x1c5b4:    lea     rax, [rip + 0x5123]
0x1c5bb:    cmovae  r15, rax

```

```

0x1c5bf:      call    0x49b0
0x1c5c4:      cmp     rax, 0x10
0x1c5c8:      lea     rax, [rip + 0x5e16]
0x1c5cf:      cmova   r14, rax
0x1c5d3:      mov     rax, qword ptr [rsp]
0x1c5d7:      add     rax, 0x288
0x1c5dd:      mov     qword ptr [rsp + 8], rax
0x1c5e2:      cmp     r13, r12
0x1c5e5:      ja      0x1c740
0x1c5eb:      mov     rax, r12
0x1c5ee:      xor     edx, edx
0x1c5f0:      div     r13
0x1c5f3:      test    rdx, rdx
0x1c5f6:      jne     0x1c604
0x1c5f8:      mul     rbx
0x1c5fb:      mov     rcx, rax
0x1c5fe:      jno     0x1cb98
0x1c604:      mov     qword ptr [rsp + 0x20], r12
0x1c609:      fild    qword ptr [rsp + 0x20]
0x1c60d:      test    r12, r12
0x1c610:      js      0x1c928
0x1c616:      mov     rax, qword ptr [rsp + 0x18]
0x1c61b:      mov     qword ptr [rsp + 0x20], rax
0x1c620:      fild    qword ptr [rsp + 0x20]
0x1c624:      test    rax, rax
0x1c627:      js      0x1c908
0x1c62d:      mov     qword ptr [rsp + 0x20], rbx
0x1c632:      fdivp   st(1)
0x1c634:      fild    qword ptr [rsp + 0x20]
0x1c638:      test    rbx, rbx
0x1c63b:      js      0x1c8f0
0x1c641:      fmulp   st(1)
0x1c643:      test    byte ptr [rsp + 0x48], 0x10
0x1c648:      je      0x1c938
0x1c64e:      fild    dword ptr [rsp + 0x34]
0x1c652:      xor     ebx, ebx
0x1c654:      fld     st(0)
0x1c656:      jmp     0x1c664
0x1c664:
0x1c658:      nop     dword ptr [rax + rax]
0x1c660:      fstp    st(1)
0x1c662:      fxch    st(2)
0x1c664:      fld     st(0)
0x1c666:      add     ebx, 1
0x1c669:      fmul    st(2)
0x1c66b:      fxch    st(3)
0x1c66d:      fcomi   st(3)
0x1c66f:      jnb     0x1c680

```

```

0x1c671:      cmp     ebx, 8
0x1c674:      jne     0x1c660
0x1c676:      fstp    st(2)
0x1c678:      fstp    st(2)
0x1c67a:      jmp     0x1c684
0x1c684:
0x1c67c:      nop     dword ptr [rax]
0x1c680:      fstp    st(2)
0x1c682:      fstp    st(2)
0x1c684:      lea     r15, [rbp + 1]
0x1c688:      fdivrp   st(1)
0x1c68a:      cmp     dword ptr [rsp + 0x30], 1
0x1c68f:      adc     rbp, 2
0x1c693:      cmp     dword ptr [rsp + 0x10], 1
0x1c698:      je      0x1cc70
0x1c69e:      fld     xword ptr [rip + 0x5e9c]
0x1c6a4:      fcomip   st(1)
0x1c6a6:      jbe     0x1cb10
0x1c6ac:      fld     dword ptr [rip + 0x5e7e]
0x1c6b2:      fxch     st(1)
0x1c6b4:      fcomi    st(1)
0x1c6b6:      jae     0x1ce00
0x1c6bc:      fstp    st(1)
0x1c6be:      fnstcw   word ptr [rsp + 0x4e]
0x1c6c2:      movzx    eax, word ptr [rsp + 0x4e]
0x1c6c7:      or      ah, 0xc
0x1c6ca:      mov     word ptr [rsp + 0x4c], ax
0x1c6cf:      fld     st(0)
0x1c6d1:      fldcw    word ptr [rsp + 0x4c]
0x1c6d5:      fistp    qword ptr [rsp + 0x20]
0x1c6d9:      fldcw    word ptr [rsp + 0x4e]
0x1c6dd:      mov     rax, qword ptr [rsp + 0x20]
0x1c6e2:      mov     qword ptr [rsp + 0x20], rax
0x1c6e7:      fild     qword ptr [rsp + 0x20]
0x1c6eb:      test     rax, rax
0x1c6ee:      jns     0x1c6f6
0x1c6f0:      fadd     dword ptr [rip + 0x5e36]
0x1c6f6:      mov     ecx, dword ptr [rsp + 0x10]
0x1c6fa:      test     ecx, ecx
0x1c6fc:      jne     0x1cb18
0x1c702:      fxch     st(1)
0x1c704:      fucomi    st(1)
0x1c706:      jp      0x1c712
0x1c708:      je      0x1cb20
0x1c70e:      fstp    st(1)
0x1c710:      jmp     0x1c718
0x1c718:
0x1c712:      fstp    st(1)

```



```

0x1c714:      nop      dword ptr [rax]
0x1c718:      add      rax, 1
0x1c71c:      mov      qword ptr [rsp + 0x20], rax
0x1c721:      fild     qword ptr [rsp + 0x20]
0x1c725:      jns      0x1cb1c
0x1c72b:      fadd     dword ptr [rip + 0x5dfb]
0x1c731:      fxch     st(1)
0x1c733:      jmp      0x1cb20
0x1cb20
0x1c738:      nop      dword ptr [rax + rax]
0x1c740:      test     r12, r12
0x1c743:      je       0x1c604
0x1c749:      mov      rax, qword ptr [rsp + 0x18]
0x1c74e:      xor      edx, edx
0x1c750:      div      r12
0x1c753:      mov      r8, rax
0x1c756:      test     rdx, rdx
0x1c759:      jne      0x1c604
0x1c75f:      mov      rax, rbx
0x1c762:      xor      edx, edx
0x1c764:      div      r8
0x1c767:      mov      rcx, rax
0x1c76a:      lea      rax, [rdx + rdx*4]
0x1c76e:      xor      edx, edx
0x1c770:      add      rax, rax
0x1c773:      div      r8
0x1c776:      add      rdx, rdx
0x1c779:      mov      edi, eax
0x1c77b:      cmp      r8, rdx
0x1c77e:      jbe      0x1cde8
0x1c784:      xor      esi, esi
0x1c786:      test     rdx, rdx
0x1c789:      setne    sil
0x1c78d:      mov      r10d, dword ptr [rsp + 0x48]
0x1c792:      and      r10d, 0x10
0x1c796:      je       0x1cbab
0x1c79c:      mov      r8d, dword ptr [rsp + 0x34]
0x1c7a1:      xor      ebx, ebx
0x1c7a3:      mov      r11, r8
0x1c7a6:      cmp      r8, rcx
0x1c7a9:      jbe      0x1cacf
0x1c7af:      cmp      dword ptr [rsp + 0x10], 1
0x1c7b4:      mov      r8, qword ptr [rsp + 8]
0x1c7b9:      je       0x1cdc0
0x1c7bf:      mov      r11d, dword ptr [rsp + 0x10]
0x1c7c4:      test     r11d, r11d
0x1c7c7:      jne      0x1c7e8
0x1c7c9:      add      esi, edi

```

```

0x1c7cb:    test    esi, esi
0x1c7cd:    jle     0x1c7e8
0x1c7cf:    add     rcx, 1
0x1c7d3:    test    r10d, r10d
0x1c7d6:    je      0x1c7e8
0x1c7d8:    mov     eax, dword ptr [rsp + 0x34]
0x1c7dc:    cmp     rax, rcx
0x1c7df:    je      0x1cec2
0x1c7e5:    nop     dword ptr [rax]
0x1c7e8:    movabs  rsi, 0xffffffffcccccd
0x1c7f2:    mov     rbp, r8
0x1c7f5:    nop     dword ptr [rax]
0x1c7f8:    mov     rax, rcx
0x1c7fb:    sub     rbp, 1
0x1c7ff:    mul     rsi
0x1c802:    mov     rax, rcx
0x1c805:    shr     rdx, 3
0x1c809:    lea     rdi, [rdx + rdx*4]
0x1c80d:    add     rdi, rdi
0x1c810:    sub     rax, rdi
0x1c813:    add     eax, 0x30
0x1c816:    mov     byte ptr [rbp], al
0x1c819:    mov     rax, rcx
0x1c81c:    mov     rcx, rdx
0x1c81f:    cmp     rax, 9
0x1c823:    ja      0x1c7f8
0x1c825:    test    byte ptr [rsp + 0x48], 4
0x1c82a:    je      0x1ca2c
0x1c830:    mov     rdi, r14
0x1c833:    mov     r12, r8
0x1c836:    mov     qword ptr [rsp + 0x40], r8
0x1c83b:    mov     r13, -1
0x1c842:    call    0x49b0
0x1c847:    sub     r12, rbp
0x1c84a:    mov     rsi, rbp
0x1c84d:    mov     ecx, 0x29
0x1c852:    mov     r15, rax
0x1c855:    lea     rax, [rsp + 0x50]
0x1c85a:    mov     rdx, r12
0x1c85d:    mov     rdi, rax
0x1c860:    mov     qword ptr [rsp + 0x10], rax
0x1c865:    call    0x4b70
0x1c86a:    mov     dword ptr [rsp + 0x20], ebx
0x1c86e:    mov     rbp, qword ptr [rsp + 0x40]
0x1c873:    mov     rbx, r12
0x1c876:    mov     r12, qword ptr [rsp + 0x38]
0x1c87b:    jmp     0x1c8bb
0x1c8bb:

```

```

0x1c87d:      nop      dword ptr [rax]
0x1c880:      cmp      r13, rbx
0x1c883:      mov      rax, qword ptr [rsp + 0x10]
0x1c888:      cmova    r13, rbx
0x1c88c:      sub      rbx, r13
0x1c88f:      lea      rsi, [rax + rbx]
0x1c893:      sub      rbp, r13
0x1c896:      mov      rdx, r13
0x1c899:      mov      rdi, rbp
0x1c89c:      call     0x4bb0
0x1c8a1:      test     rbx, rbx
0x1c8a4:      je       0x1caa0
0x1c8aa:      sub      rbp, r15
0x1c8ad:      mov      rdx, r15
0x1c8b0:      mov      rsi, r14
0x1c8b3:      mov      rdi, rbp
0x1c8b6:      call     0x4bb0
0x1c8bb:      movzx    eax, byte ptr [r12]
0x1c8c0:      test     al, al
0x1c8c2:      je       0x1c880
0x1c8c4:      cmp      al, 0x7e
0x1c8c6:      ja       0x1ca88
0x1c8cc:      cmp      rax, rbx
0x1c8cf:      cmova    rax, rbx
0x1c8d3:      sub      rbx, rax
0x1c8d6:      mov      r13, rax
0x1c8d9:      mov      rax, qword ptr [rsp + 0x10]
0x1c8de:      lea      rsi, [rax + rbx]
0x1c8e2:      add      r12, 1
0x1c8e6:      jmp      0x1c893
0x1c893:
0x1c8e8:      nop      dword ptr [rax + rax]
0x1c8f0:      fadd     dword ptr [rip + 0x5c36]
0x1c8f6:      fmulp    st(1)
0x1c8f8:      test     byte ptr [rsp + 0x48], 0x10
0x1c8fd:      jne      0x1c64e
0x1c903:      jmp      0x1c938
0x1c938:
0x1c905:      nop      dword ptr [rax]
0x1c908:      fadd     dword ptr [rip + 0x5c1e]
0x1c90e:      mov      qword ptr [rsp + 0x20], rbx
0x1c913:      fdivp    st(1)
0x1c915:      fild     qword ptr [rsp + 0x20]
0x1c919:      test     rbx, rbx
0x1c91c:      jns      0x1c641
0x1c922:      jmp      0x1c8f0
0x1c8f0:
0x1c924:      nop      dword ptr [rax]

```

```

0x1c928:      fadd    dword ptr [rip + 0x5bfe]
0x1c92e:      jmp     0x1c616
0x1c616
0x1c933:      nop     dword ptr [rax + rax]
0x1c938:      cmp     dword ptr [rsp + 0x10], 1
0x1c93d:      je      0x1c9ca
0x1c943:      fld     xword ptr [rip + 0x5bf7]
0x1c949:      fcomip   st(1)
0x1c94b:      jbe     0x1c9ca
0x1c94d:      fld     dword ptr [rip + 0x5bdd]
0x1c953:      fxch     st(1)
0x1c955:      fcomi    st(1)
0x1c957:      jae     0x1ce60
0x1c95d:      fstp     st(1)
0x1c95f:      fnstcw   word ptr [rsp + 0x4e]
0x1c963:      movzx    eax, word ptr [rsp + 0x4e]
0x1c968:      or       ah, 0xc
0x1c96b:      mov      word ptr [rsp + 0x4c], ax
0x1c970:      fld     st(0)
0x1c972:      fldcw    word ptr [rsp + 0x4c]
0x1c976:      fistp    qword ptr [rsp + 0x20]
0x1c97a:      fldcw    word ptr [rsp + 0x4e]
0x1c97e:      mov      rax, qword ptr [rsp + 0x20]
0x1c983:      mov      qword ptr [rsp + 0x20], rax
0x1c988:      fild     qword ptr [rsp + 0x20]
0x1c98c:      test     rax, rax
0x1c98f:      jns      0x1c997
0x1c991:      fadd     dword ptr [rip + 0x5b95]
0x1c997:      mov      r8d, dword ptr [rsp + 0x10]
0x1c99c:      test     r8d, r8d
0x1c99f:      jne      0x1c9c8
0x1c9a1:      fxch     st(1)
0x1c9a3:      fucomip  st(1)
0x1c9a5:      jp       0x1c9ad
0x1c9a7:      je       0x1c9ca
0x1c9a9:      fstp     st(0)
0x1c9ab:      jmp      0x1c9af
0x1c9af
0x1c9ad:      fstp     st(0)
0x1c9af:      add      rax, 1
0x1c9b3:      mov      qword ptr [rsp + 0x10], rax
0x1c9b8:      fild     qword ptr [rsp + 0x10]
0x1c9bc:      js       0x1cee7
0x1c9c2:      jmp      0x1c9ca
0x1c9ca
0x1c9c4:      nop     dword ptr [rax]
0x1c9c8:      fstp     st(1)
0x1c9ca:      sub      rsp, 0x10

```

```

0x1c9ce:    mov     rdx, -1
0x1c9d5:    mov     esi, 1
0x1c9da:    xor     eax, eax
0x1c9dc:    mov     rbx, qword ptr [rsp + 0x10]
0x1c9e1:    lea     rcx, [rip + 0x5adc]
0x1c9e8:    fstp    xword ptr [rsp]
0x1c9eb:    mov     rdi, rbx
0x1c9ee:    call    0x4f10
0x1c9f3:    mov     rdi, rbx
0x1c9f6:    mov     ebx, 0xffffffff
0x1c9fb:    call    0x49b0
0x1ca00:    pop     rsi
0x1ca01:    pop     rdi
0x1ca02:    mov     rdx, rax
0x1ca05:    mov     r12, rax
0x1ca08:    mov     rbp, qword ptr [rsp + 8]
0x1ca0d:    mov     rsi, qword ptr [rsp]
0x1ca11:    sub     rbp, rdx
0x1ca14:    mov     rdi, rbp
0x1ca17:    call    0x4d40
0x1ca1c:    lea     r8, [rbp + r12]
0x1ca21:    test    byte ptr [rsp + 0x48], 4
0x1ca26:    jne     0x1c830
0x1ca2c:    test    byte ptr [rsp + 0x48], 0x80
0x1ca31:    je      0x1ca4f
0x1ca33:    cmp     ebx, -1
0x1ca36:    je      0x1cbb8
0x1ca3c:    mov     eax, dword ptr [rsp + 0x48]
0x1ca40:    and     eax, 0x100
0x1ca45:    mov     esi, eax
0x1ca47:    or      esi, ebx
0x1ca49:    jne     0x1cd08
0x1ca4f:    mov     rax, qword ptr [rsp + 8]
0x1ca54:    mov     byte ptr [rax], 0
0x1ca57:    mov     rax, qword ptr [rsp + 0x88]
0x1ca5f:    sub     rax, qword ptr fs:[0x28]
0x1ca68:    jne     0x1d0eb
0x1ca6e:    add     rsp, 0x98
0x1ca75:    mov     rax, rbp
0x1ca78:    pop     rbx
0x1ca79:    pop     rbp
0x1ca7a:    pop     r12
0x1ca7c:    pop     r13
0x1ca7e:    pop     r14
0x1ca80:    pop     r15
0x1ca82:    ret
0x1ca83:    nop     dword ptr [rax + rax]
0x1ca88:    mov     r13, rbx

```

```

0x1ca8b:    mov     rsi, qword ptr [rsp + 0x10]
0x1ca90:    xor     ebx, ebx
0x1ca92:    jmp     0x1c8e2
0x1c8e2:
0x1ca97:    nop     word ptr [rax + rax]
0x1caa0:    mov     ebx, dword ptr [rsp + 0x20]
0x1caa4:    jmp     0x1ca2c
0x1ca2c:
0x1caa6:    nop     word ptr cs:[rax + rax]
0x1cab0:    test    esi, esi
0x1cab2:    setne   sil
0x1cab6:    movzx   esi, sil
0x1caba:    add     ebx, 1
0x1cabd:    cmp     r8, r9
0x1cac0:    ja      0x1cf18
0x1cac6:    cmp     ebx, 8
0x1cac9:    je      0x1c7af
0x1cacf:    mov     rax, rcx
0x1cad2:    xor     edx, edx
0x1cad4:    mov     ecx, esi
0x1cad6:    div     r8
0x1cad9:    sar     ecx, 1
0x1cadb:    mov     r9, rax
0x1cade:    lea     eax, [rdx + rdx*4]
0x1cae1:    xor     edx, edx
0x1cae3:    lea     eax, [rdi + rax*2]
0x1cae6:    div     r11d
0x1cae9:    lea     edx, [rcx + rdx*2]
0x1caec:    mov     edi, eax
0x1caee:    mov     rcx, r9
0x1caf1:    add     esi, edx
0x1caf3:    cmp     r11d, edx
0x1caf6:    ja      0x1cab0
0x1caf8:    cmp     r11d, esi
0x1cafb:    setb    sil
0x1caff:    movzx   esi, sil
0x1cb03:    add     esi, 2
0x1cb06:    jmp     0x1caba
0x1caba:
0x1cb08:    nop     dword ptr [rax + rax]
0x1cb10:    fld     st(0)
0x1cb12:    jmp     0x1cb20
0x1cb20:
0x1cb14:    nop     dword ptr [rax]
0x1cb18:    fxch    st(1)
0x1cb1a:    jmp     0x1cb20
0x1cb20:
0x1cb1c:    fxch    st(1)

```

```

0x1cb1e:      nop
0x1cb20:      fstp    xword ptr [rsp + 0x20]
0x1cb24:      sub     rsp, 0x10
0x1cb28:      mov     esi, 1
0x1cb2d:      xor     eax, eax
0x1cb2f:      mov     r12, qword ptr [rsp + 0x10]
0x1cb34:      lea     rcx, [rip + 0x598f]
0x1cb3b:      mov     rdx, -1
0x1cb42:      mov     rdi, r12
0x1cb45:      fstp    xword ptr [rsp]
0x1cb48:      call    0x4f10
0x1cb4d:      mov     rdi, r12
0x1cb50:      call    0x49b0
0x1cb55:      mov     rdx, rax
0x1cb58:      pop     rax
0x1cb59:      pop     rcx
0x1cb5a:      fld     xword ptr [rsp + 0x20]
0x1cb5e:      cmp     rdx, rbp
0x1cb61:      ja      0x1cd20
0x1cb67:      test    byte ptr [rsp + 0x48], 8
0x1cb6c:      je      0x1cb88
0x1cb6e:      mov     rax, qword ptr [rsp]
0x1cb72:      cmp     byte ptr [rax + rdx - 1], 0x30
0x1cb77:      je      0x1ce40
0x1cb7d:      fstp    st(0)
0x1cb7f:      jmp     0x1cb8a
0x1cb8a:
0x1cb81:      nop     dword ptr [rax]
0x1cb88:      fstp    st(0)
0x1cb8a:      mov     r12, rdx
0x1cb8d:      sub     r12, r15
0x1cb90:      jmp     0x1ca08
0x1ca08:
0x1cb95:      nop     dword ptr [rax]
0x1cb98:      mov     r10d, dword ptr [rsp + 0x48]
0x1cb9d:      xor     esi, esi
0x1cb9f:      xor     edi, edi
0x1cba1:      and     r10d, 0x10
0x1cba5:      jne     0x1c79c
0x1cbab:      mov     ebx, 0xffffffff
0x1cbb0:      jmp     0x1c7af
0x1c7af:
0x1cbb5:      nop     dword ptr [rax]
0x1cbb8:      mov     rcx, qword ptr [rsp + 0x18]
0x1cbbd:      cmp     rcx, 1
0x1cbc1:      jbe     0x1cfca
0x1cbc7:      mov     edx, dword ptr [rsp + 0x34]
0x1cbcb:      mov     ebx, 1

```

```

0x1cbd0:    mov     eax, 1
0x1cbd5:    nop     dword ptr [rax]
0x1cbd8:    imul    rax, rdx
0x1cbdc:    cmp     rcx, rax
0x1cbdf:    jbe     0x1cbe9
0x1cbe1:    add     ebx, 1
0x1cbe4:    cmp     ebx, 8
0x1cbe7:    jne     0x1cbd8
0x1cbe9:    mov     esi, dword ptr [rsp + 0x48]
0x1cbec:    mov     eax, esi
0x1cbef:    and     eax, 0x100
0x1cbf4:    and     esi, 0x40
0x1cbf7:    je      0x1cc18
0x1cbf9:    mov     rsi, qword ptr [rsp]
0x1cbfd:    lea     rdi, [rsi + 0x289]
0x1cc04:    mov     byte ptr [rsi + 0x288], 0x20
0x1cc0b:    mov     qword ptr [rsp + 8], rdi
0x1cc10:    test    ebx, ebx
0x1cc12:    je      0x1d105
0x1cc18:    mov     rsi, qword ptr [rsp + 8]
0x1cc1d:    mov     r9d, dword ptr [rsp + 0x30]
0x1cc22:    lea     rdx, [rsi + 1]
0x1cc26:    test    r9d, r9d
0x1cc29:    jne     0x1cc34
0x1cc2b:    cmp     ebx, 1
0x1cc2e:    je      0x1cf00
0x1cc34:    movsxd  rbx, ebx
0x1cc37:    lea     rcx, [rip + 0x58e2]
0x1cc3e:    movzx   ecx, byte ptr [rcx + rbx]
0x1cc42:    mov     byte ptr [rsi], cl
0x1cc44:    test    eax, eax
0x1cc46:    je      0x1cf0b
0x1cc4c:    mov     r8d, dword ptr [rsp + 0x30]
0x1cc51:    test    r8d, r8d
0x1cc54:    jne     0x1ce30
0x1cc5a:    lea     rax, [rdx + 1]
0x1cc5e:    mov     byte ptr [rdx], 0x42
0x1cc61:    mov     qword ptr [rsp + 8], rax
0x1cc66:    jmp     0x1ca4f
0x1ca4f:
0x1cc6b:    nop     dword ptr [rax + rax]
0x1cc70:    sub     rsp, 0x10
0x1cc74:    mov     rdx, -1
0x1cc7b:    mov     esi, 1
0x1cc80:    xor     eax, eax
0x1cc82:    fld     st(0)
0x1cc84:    fstp    xword ptr [rsp]
0x1cc87:    mov     r12, qword ptr [rsp + 0x10]

```



```

0x1cc8c:    lea     rcx, [rip + 0x5837]
0x1cc93:    mov     rdi, r12
0x1cc96:    fstp    xword ptr [rsp + 0x30]
0x1cc9a:    call    0x4f10
0x1cc9f:    mov     rdi, r12
0x1cca2:    call    0x49b0
0x1cca7:    pop     rsi
0x1cca8:    pop     rdi
0x1cca9:    fld     xword ptr [rsp + 0x20]
0x1ccad:    cmp     rax, rbp
0x1ccb0:    mov     rdx, rax
0x1ccb3:    jbe     0x1cb67
0x1ccb9:    fld     dword ptr [rip + 0x5875]
0x1ccbfb:    fmul    st(1), st(0)
0x1ccc1:    fdivp    st(1)
0x1ccc3:    sub     rsp, 0x10
0x1ccc7:    mov     rdx, -1
0x1ccce:    xor     eax, eax
0x1ccd0:    mov     r15, qword ptr [rsp + 0x10]
0x1ccd5:    lea     rcx, [rip + 0x57e8]
0x1ccdc:    mov     esi, 1
0x1cce1:    mov     rdi, r15
0x1cce4:    fstp    xword ptr [rsp]
0x1cce7:    call    0x4f10
0x1ccecb:    mov     rdi, r15
0x1ccf4:    call    0x49b0
0x1ccf4:    pop     r13
0x1ccf6:    pop     r15
0x1ccf8:    mov     rdx, rax
0x1ccfb:    mov     r12, rax
0x1ccfe:    jmp     0x1ca08
0x1ca08:
0x1cd03:    nop     dword ptr [rax + rax]
0x1cd08:    test    byte ptr [rsp + 0x48], 0x40
0x1cd0d:    je      0x1cc10
0x1cd13:    jmp     0x1cbf9
0x1cbf9:
0x1cd18:    nop     dword ptr [rax + rax]
0x1cd20:    fld     dword ptr [rip + 0x580e]
0x1cd26:    fmul    st(1), st(0)
0x1cd28:    fld     xword ptr [rip + 0x5812]
0x1cd2e:    fcomip  st(2)
0x1cd30:    jbe     0x1ccc1
0x1cd32:    fld     dword ptr [rip + 0x57f8]
0x1cd38:    fxch    st(2)
0x1cd3a:    fcomi   st(2)
0x1cd3c:    jae     0x1ce90
0x1cd42:    fstp    st(2)

```

```

0x1cd44:    fxch    st(1)
0x1cd46:    fnstcw  word ptr [rsp + 0x4e]
0x1cd4a:    movzx   eax, word ptr [rsp + 0x4e]
0x1cd4f:    or      ah, 0xc
0x1cd52:    mov     word ptr [rsp + 0x4c], ax
0x1cd57:    fld     st(0)
0x1cd59:    fldcw   word ptr [rsp + 0x4c]
0x1cd5d:    fistp   qword ptr [rsp + 0x20]
0x1cd61:    fldcw   word ptr [rsp + 0x4e]
0x1cd65:    mov     rax, qword ptr [rsp + 0x20]
0x1cd6a:    mov     qword ptr [rsp + 0x20], rax
0x1cd6f:    fild    qword ptr [rsp + 0x20]
0x1cd73:    test    rax, rax
0x1cd76:    jns     0x1cd7e
0x1cd78:    fadd    dword ptr [rip + 0x57ae]
0x1cd7e:    mov     edx, dword ptr [rsp + 0x10]
0x1cd82:    test    edx, edx
0x1cd84:    jne     0x1cdb0
0x1cd86:    fucomi  st(1)
0x1cd88:    fstp    st(1)
0x1cd8a:    jp      0x1cd92
0x1cd8c:    je      0x1cdb2
0x1cd8e:    fstp    st(0)
0x1cd90:    jmp     0x1cd94
0x1cd94:
0x1cd92:    fstp    st(0)
0x1cd94:    add     rax, 1
0x1cd98:    mov     qword ptr [rsp + 0x10], rax
0x1cd9d:    fild    qword ptr [rsp + 0x10]
0x1cda1:    js      0x1cef2
0x1cda7:    jmp     0x1cdb2
0x1cdb2:
0x1cda9:    nop     dword ptr [rax]
0x1cdb0:    fstp    st(1)
0x1cdb2:    fxch    st(1)
0x1cdb4:    jmp     0x1ccc1
0x1ccc1:
0x1cdb9:    nop     dword ptr [rax]
0x1cdc0:    mov     rax, rcx
0x1cdc3:    movsxd  rsi, esi
0x1cdc6:    and     eax, 1
0x1cdc9:    add     rax, rsi
0x1cdcc:    setne   al
0x1cdcf:    movzx   eax, al
0x1cdd2:    add     eax, edi
0x1cdd4:    cmp     eax, 5
0x1cdd7:    jg      0x1c7cf
0x1cddd:    jmp     0x1c7e8

```

```

0x1c7e8
0x1cde2:      nop      word ptr [rax + rax]
0x1cde8:      mov      esi, 2
0x1cded:      mov      eax, 3
0x1cdf2:      cmovb    esi, eax
0x1cdf5:      jmp      0x1c78d
0x1c78d
0x1cdfa:      nop      word ptr [rax + rax]
0x1ce00:      fnstcw   word ptr [rsp + 0x4e]
0x1ce04:      fsubr     st(1), st(0)
0x1ce06:      fxch      st(1)
0x1ce08:      movzx     eax, word ptr [rsp + 0x4e]
0x1ce0d:      or        ah, 0xc
0x1ce10:      mov      word ptr [rsp + 0x4c], ax
0x1ce15:      fldcw     word ptr [rsp + 0x4c]
0x1ce19:      fistp     qword ptr [rsp + 0x20]
0x1ce1d:      fldcw     word ptr [rsp + 0x4e]
0x1ce21:      mov      rax, qword ptr [rsp + 0x20]
0x1ce26:      btc       rax, 0x3f
0x1ce2b:      jmp      0x1c6e2
0x1c6e2
0x1ce30:      mov      byte ptr [rsi + 1], 0x69
0x1ce34:      lea       rdx, [rsi + 2]
0x1ce38:      jmp      0x1cc5a
0x1cc5a
0x1ce3d:      nop      dword ptr [rax]
0x1ce40:      fld       dword ptr [rip + 0x56ee]
0x1ce46:      cmp      dword ptr [rsp + 0x10], 1
0x1ce4b:      fmul      st(1), st(0)
0x1ce4d:      jne      0x1cd28
0x1ce53:      jmp      0x1ccc1
0x1ccc1
0x1ce58:      nop      dword ptr [rax + rax]
0x1ce60:      fnstcw   word ptr [rsp + 0x4e]
0x1ce64:      fsubr     st(1), st(0)
0x1ce66:      fxch      st(1)
0x1ce68:      movzx     eax, word ptr [rsp + 0x4e]
0x1ce6d:      or        ah, 0xc
0x1ce70:      mov      word ptr [rsp + 0x4c], ax
0x1ce75:      fldcw     word ptr [rsp + 0x4c]
0x1ce79:      fistp     qword ptr [rsp + 0x20]
0x1ce7d:      fldcw     word ptr [rsp + 0x4e]
0x1ce81:      mov      rax, qword ptr [rsp + 0x20]
0x1ce86:      btc       rax, 0x3f
0x1ce8b:      jmp      0x1c983
0x1c983
0x1ce90:      fnstcw   word ptr [rsp + 0x4e]
0x1ce94:      fsubr     st(2), st(0)

```

```

0x1ce96:    fxch    st(2)
0x1ce98:    movzx   eax, word ptr [rsp + 0x4e]
0x1ce9d:    or      ah, 0xc
0x1cea0:    mov     word ptr [rsp + 0x4c], ax
0x1cea5:    fldcw   word ptr [rsp + 0x4c]
0x1cea9:    fistp   qword ptr [rsp + 0x20]
0x1cead:    fldcw   word ptr [rsp + 0x4e]
0x1ceb1:    fxch    st(1)
0x1ceb3:    mov     rax, qword ptr [rsp + 0x20]
0x1ceb8:    btc     rax, 0x3f
0x1cebd:    jmp     0x1cd6a
0x1cd6a:
0x1cec2:    cmp     ebx, 8
0x1cec5:    je      0x1c7e8
0x1cecb:    add     ebx, 1
0x1cece:    test    byte ptr [rsp + 0x48], 8
0x1ced3:    je      0x1d039
0x1ced9:    mov     byte ptr [r8 - 1], 0x31
0x1cede:    lea     rbp, [r8 - 1]
0x1cee2:    jmp     0x1ca21
0x1ca21:
0x1cee7:    fadd    dword ptr [rip + 0x563f]
0x1ceed:    jmp     0x1c9ca
0x1c9ca:
0x1cef2:    fadd    dword ptr [rip + 0x5634]
0x1cef8:    fxch    st(1)
0x1cefa:    jmp     0x1ccc1
0x1ccc1:
0x1ceff:    nop
0x1cf00:    mov     byte ptr [rsi], 0x6b
0x1cf03:    test    eax, eax
0x1cf05:    jne     0x1cc5a
0x1cf0b:    mov     qword ptr [rsp + 8], rdx
0x1cf10:    jmp     0x1ca4f
0x1ca4f:
0x1cf15:    nop     dword ptr [rax]
0x1cf18:    cmp     r9, 9
0x1cf1c:    ja      0x1c7af
0x1cf22:    cmp     dword ptr [rsp + 0x10], 1
0x1cf27:    je      0x1cfa0
0x1cf29:    mov     r12d, dword ptr [rsp + 0x10]
0x1cf2e:    test    r12d, r12d
0x1cf31:    jne     0x1cfac
0x1cf33:    test    esi, esi
0x1cf35:    je      0x1cfac
0x1cf37:    lea     edx, [rax + 1]
0x1cf3a:    cmp     eax, 9
0x1cf3d:    je      0x1d024

```

```

0x1cf43:    lea     eax, [rdx + 0x30]
0x1cf46:    mov     rdi, qword ptr [rsp]
0x1cf4a:    lea     r8, [rdi + 0x287]
0x1cf51:    mov     byte ptr [rdi + 0x287], al
0x1cf57:    mov     eax, ebp
0x1cf59:    sub     r8, rbp
0x1cf5c:    cmp     ebp, 8
0x1cf5f:    jae     0x1cfd1
0x1cf61:    test    bpl, 4
0x1cf65:    jne     0x1d0d6
0x1cf6b:    test    eax, eax
0x1cf6d:    je      0x1cf7a
0x1cf6f:    movzx   edx, byte ptr [r15]
0x1cf73:    mov     byte ptr [r8], dl
0x1cf76:    test    al, 2
0x1cf78:    jne     0x1cf8e
0x1cf7a:    xor     esi, esi
0x1cf7c:    xor     edi, edi
0x1cf7e:    cmp     dword ptr [rsp + 0x10], 1
0x1cf83:    jne     0x1c7bf
0x1cf89:    jmp     0x1c7e8
0x1c7e8
0x1cf8e:    movzx   edx, word ptr [r15 + rax - 2]
0x1cf94:    mov     word ptr [r8 + rax - 2], dx
0x1cf9a:    jmp     0x1cf7a
0x1cf7a
0x1cf9c:    nop     dword ptr [rax]
0x1cfa0:    mov     edx, eax
0x1cfa2:    and     edx, 1
0x1cfa5:    add     edx, esi
0x1cfa7:    cmp     edx, 2
0x1cfaa:    jg      0x1cf37
0x1cfac:    test    eax, eax
0x1cfae:    jne     0x1d117
0x1cfb4:    mov     r8, qword ptr [rsp + 8]
0x1cfb9:    test    byte ptr [rsp + 0x48], 8
0x1cfbe:    jne     0x1cf7c
0x1cfc0:    mov     eax, 0x30
0x1cfc5:    jmp     0x1cf46
0x1cf46
0x1cfca:    xor     ebx, ebx
0x1cfcc:    jmp     0x1ca3c
0x1ca3c
0x1cfd1:    mov     rax, qword ptr [r15]
0x1cfd4:    lea     rsi, [r8 + 8]
0x1cfd8:    mov     rdi, r15
0x1cfdb:    and     rsi, 0xfffffffffffffff8
0x1cfd1:    mov     qword ptr [r8], rax

```

```

0x1cfe2:    mov     eax, ebp
0x1cfe4:    mov     rdx, qword ptr [r15 + rax - 8]
0x1cfe9:    mov     qword ptr [r8 + rax - 8], rdx
0x1cfee:    mov     rax, r8
0x1cff1:    sub     rax, rsi
0x1cff4:    sub     rdi, rax
0x1cff7:    add     eax, ebp
0x1cff9:    and     eax, 0xffffffff8
0x1cffc:    cmp     eax, 8
0x1cfff:    jb      0x1cf7a
0x1d005:    and     eax, 0xffffffff8
0x1d008:    mov     r9d, eax
0x1d00b:    xor     eax, eax
0x1d00d:    mov     edx, eax
0x1d00f:    add     eax, 8
0x1d012:    mov     r11, qword ptr [rdi + rdx]
0x1d016:    mov     qword ptr [rsi + rdx], r11
0x1d01a:    cmp     eax, r9d
0x1d01d:    jb      0x1d00d
0x1d01f:    jmp     0x1cf7a
0x1cf7a:
0x1d024:    lea     rcx, [r9 + 1]
0x1d028:    cmp     r9, 9
0x1d02c:    je      0x1d0cc
0x1d032:    xor     esi, esi
0x1d034:    jmp     0x1cfb4
0x1cfb4:
0x1d039:    mov     rax, rbp
0x1d03c:    mov     byte ptr [r8 - 1], 0x30
0x1d041:    not     rax
0x1d044:    add     r8, rax
0x1d047:    mov     eax, ebp
0x1d049:    cmp     ebp, 8
0x1d04c:    jae     0x1d080
0x1d04e:    and     ebp, 4
0x1d051:    jne     0x1d0f0
0x1d057:    test    eax, eax
0x1d059:    je      0x1ced9
0x1d05f:    movzx   edx, byte ptr [r15]
0x1d063:    mov     byte ptr [r8], dl
0x1d066:    test    al, 2
0x1d068:    je      0x1ced9
0x1d06e:    movzx   edx, word ptr [r15 + rax - 2]
0x1d074:    mov     word ptr [r8 + rax - 2], dx
0x1d07a:    jmp     0x1ced9
0x1ced9:
0x1d07f:    nop
0x1d080:    mov     rax, qword ptr [r15]

```

```

0x1d083:    lea     rcx, [r8 + 8]
0x1d087:    and     rcx, 0xfffffffffffffff8
0x1d08b:    mov     qword ptr [r8], rax
0x1d08e:    mov     eax, ebp
0x1d090:    mov     rdx, qword ptr [r15 + rax - 8]
0x1d095:    mov     qword ptr [r8 + rax - 8], rdx
0x1d09a:    mov     rax, r8
0x1d09d:    sub     rax, rcx
0x1d0a0:    sub     r15, rax
0x1d0a3:    add     eax, ebp
0x1d0a5:    and     eax, 0xffffffff8
0x1d0a8:    cmp     eax, 8
0x1d0ab:    jb      0x1ced9
0x1d0b1:    and     eax, 0xffffffff8
0x1d0b4:    xor     edx, edx
0x1d0b6:    mov     esi, edx
0x1d0b8:    add     edx, 8
0x1d0bb:    mov     rdi, qword ptr [r15 + rsi]
0x1d0bf:    mov     qword ptr [rcx + rsi], rdi
0x1d0c3:    cmp     edx, eax
0x1d0c5:    jb      0x1d0b6
0x1d0c7:    jmp     0x1ced9
0x1ced9
0x1d0cc:    mov     r8, qword ptr [rsp + 8]
0x1d0d1:    jmp     0x1cf7a
0x1cf7a
0x1d0d6:    mov     edx, dword ptr [r15]
0x1d0d9:    mov     dword ptr [r8], edx
0x1d0dc:    mov     edx, dword ptr [r15 + rax - 4]
0x1d0e1:    mov     dword ptr [r8 + rax - 4], edx
0x1d0e6:    jmp     0x1cf7a
0x1cf7a
0x1d0eb:    call    0x49d0
0x1d0f0:    mov     edx, dword ptr [r15]
0x1d0f3:    mov     dword ptr [r8], edx
0x1d0f6:    mov     edx, dword ptr [r15 + rax - 4]
0x1d0fb:    mov     dword ptr [r8 + rax - 4], edx
0x1d100:    jmp     0x1ced9
0x1ced9
0x1d105:    mov     rdx, qword ptr [rsp + 8]
0x1d10a:    test    eax, eax
0x1d10c:    jne     0x1cc5a
0x1d112:    jmp     0x1ca4f
0x1ca4f
0x1d117:    add     eax, 0x30
0x1d11a:    jmp     0x1cf46
0x1cf46
0x1d11f:    nop

```

```

0x1d120:      endbr64
0x1d124:      push    r14
0x1d126:      push    r13
0x1d128:      mov     r13, rsi
0x1d12b:      push    r12
0x1d12d:      push    rbp
0x1d12e:      mov     rbp, rdx
0x1d131:      push    rbx
0x1d132:      mov     rbx, rdi
0x1d135:      sub     rsp, 0x10
0x1d139:      mov     rax, qword ptr fs:[0x28]
0x1d142:      mov     qword ptr [rsp + 8], rax
0x1d147:      xor     eax, eax
0x1d149:      test    rdi, rdi
0x1d14c:      je      0x1d270
0x1d152:      xor     r12d, r12d
0x1d155:      cmp     byte ptr [rbx], 0x27
0x1d158:      jne     0x1d164
0x1d15a:      add     rbx, 1
0x1d15e:      mov     r12d, 4
0x1d164:      lea     r14, [rip + 0x53ad]
0x1d16b:      mov     ecx, 4
0x1d170:      lea     rsi, [rip + 0xb819]
0x1d177:      mov     rdi, rbx
0x1d17a:      mov     rdx, r14
0x1d17d:      call    0x1dc80
0x1d182:      test    eax, eax
0x1d184:      js      0x1d1c0
0x1d186:      cdqe
0x1d188:      mov     qword ptr [rbp], 1
0x1d190:      or      r12d, dword ptr [r14 + rax*4]
0x1d194:      xor     eax, eax
0x1d196:      mov     dword ptr [r13], r12d
0x1d19a:      mov     rdx, qword ptr [rsp + 8]
0x1d19f:      sub     rdx, qword ptr fs:[0x28]
0x1d1a8:      jne     0x1d2fd
0x1d1ae:      add     rsp, 0x10
0x1d1b2:      pop     rbx
0x1d1b3:      pop     rbp
0x1d1b4:      pop     r12
0x1d1b6:      pop     r13
0x1d1b8:      pop     r14
0x1d1ba:      ret
0x1d1bb:      nop     dword ptr [rax + rax]
0x1d1c0:      xor     edx, edx
0x1d1c2:      mov     rsi, rsp
0x1d1c5:      lea     r8, [rip + 0x5329]
0x1d1cc:      mov     rcx, rbp

```



```

0x1d1cf:    mov     rdi, rbx
0x1d1d2:    call   0x1d7e0
0x1d1d7:    test   eax, eax
0x1d1d9:    jne     0x1d228
0x1d1db:    movzx   ecx, byte ptr [rbx]
0x1d1de:    lea     edx, [rcx - 0x30]
0x1d1e1:    mov     rcx, qword ptr [rsp]
0x1d1e5:    cmp     dl, 9
0x1d1e8:    ja      0x1d200
0x1d1ea:    jmp     0x1d217
0x1d217:
0x1d1ec:    nop     dword ptr [rax]
0x1d1f0:    movzx   edi, byte ptr [rbx + 1]
0x1d1f4:    add     rbx, 1
0x1d1f8:    lea     edx, [rdi - 0x30]
0x1d1fb:    cmp     dl, 9
0x1d1fe:    jbe     0x1d217
0x1d200:    cmp     rcx, rbx
0x1d203:    jne     0x1d1f0
0x1d205:    cmp     byte ptr [rcx - 1], 0x42
0x1d209:    je      0x1d2d0
0x1d20f:    or      r12b, 0x80
0x1d213:    or      r12d, 0x20
0x1d217:    mov     rdx, qword ptr [rbp]
0x1d21b:    mov     dword ptr [r13], r12d
0x1d21f:    jmp     0x1d234
0x1d234:
0x1d221:    nop     dword ptr [rax]
0x1d228:    mov     dword ptr [r13], 0
0x1d230:    mov     rdx, qword ptr [rbp]
0x1d234:    test    rdx, rdx
0x1d237:    jne     0x1d19a
0x1d23d:    lea     rdi, [rip + 0x52a1]
0x1d244:    call    0x47e0
0x1d249:    cmp     rax, 1
0x1d24d:    sbb     rax, rax
0x1d250:    and     eax, 0x200
0x1d255:    add     rax, 0x200
0x1d25b:    mov     qword ptr [rbp], rax
0x1d25f:    mov     eax, 4
0x1d264:    jmp     0x1d19a
0x1d19a:
0x1d269:    nop     dword ptr [rax]
0x1d270:    lea     rdi, [rip + 0x5259]
0x1d277:    call    0x47e0
0x1d27c:    mov     rbx, rax
0x1d27f:    test    rax, rax
0x1d282:    jne     0x1d152

```

```

0x1d288:    lea     rdi, [rip + 0x524c]
0x1d28f:    call   0x47e0
0x1d294:    mov     rbx, rax
0x1d297:    test    rax, rax
0x1d29a:    jne     0x1d152
0x1d2a0:    lea     rdi, [rip + 0x523e]
0x1d2a7:    call   0x47e0
0x1d2ac:    test    rax, rax
0x1d2af:    je      0x1d2e6
0x1d2b1:    mov     qword ptr [rbp], 0x200
0x1d2b9:    xor     eax, eax
0x1d2bb:    mov     dword ptr [r13], 0
0x1d2c3:    jmp     0x1d19a
0x1d19a:
0x1d2c8:    nop     dword ptr [rax + rax]
0x1d2d0:    or      r12d, 0x180
0x1d2d7:    cmp     byte ptr [rcx - 2], 0x69
0x1d2db:    jne     0x1d217
0x1d2e1:    jmp     0x1d213
0x1d213:
0x1d2e6:    mov     qword ptr [rbp], 0x400
0x1d2ee:    xor     eax, eax
0x1d2f0:    mov     dword ptr [r13], 0
0x1d2f8:    jmp     0x1d19a
0x1d19a:
0x1d2fd:    call   0x49d0
0x1d302:    nop     word ptr cs:[rax + rax]
0x1d30c:    nop     dword ptr [rax]
0x1d310:    endbr64
0x1d314:    push    r12
0x1d316:    push    rbp
0x1d317:    mov     ebp, edi
0x1d319:    push    rbx
0x1d31a:    mov     rbx, qword ptr [rip + 0xe81f]
0x1d321:    test    rbx, rbx
0x1d324:    jne     0x1d339
0x1d326:    jmp     0x1d350
0x1d350:
0x1d328:    nop     dword ptr [rax + rax]
0x1d330:    mov     rbx, qword ptr [rbx + 8]
0x1d334:    test    rbx, rbx
0x1d337:    je      0x1d350
0x1d339:    cmp     dword ptr [rbx], ebp
0x1d33b:    jne     0x1d330
0x1d33d:    xor     eax, eax
0x1d33f:    cmp     byte ptr [rbx + 0x10], 0
0x1d343:    je      0x1d349
0x1d345:    lea     rax, [rbx + 0x10]

```

```

0x1d349:      pop     rbx
0x1d34a:      pop     rbp
0x1d34b:      pop     r12
0x1d34d:      ret
0x1d34e:      nop
0x1d350:      mov     edi, ebp
0x1d352:      lea     r12, [rip + 0x508c]
0x1d359:      call    0x4970
0x1d35e:      mov     edi, 0x11
0x1d363:      test    rax, rax
0x1d366:      je      0x1d377
0x1d368:      mov     r12, qword ptr [rax]
0x1d36b:      mov     rdi, r12
0x1d36e:      call    0x49b0
0x1d373:      lea     rdi, [rax + 0x11]
0x1d377:      call    0x1d670
0x1d37c:      mov     rsi, r12
0x1d37f:      mov     dword ptr [rax], ebp
0x1d381:      lea     rdi, [rax + 0x10]
0x1d385:      mov     rbx, rax
0x1d388:      call    0x48b0
0x1d38d:      mov     rax, qword ptr [rip + 0xe7ac]
0x1d394:      mov     qword ptr [rip + 0xe7a5], rbx
0x1d39b:      mov     qword ptr [rbx + 8], rax
0x1d39f:      jmp     0x1d33d
0x1d33d:
0x1d3a1:      nop     word ptr cs:[rax + rax]
0x1d3ac:      nop     dword ptr [rax]
0x1d3b0:      endbr64
0x1d3b4:      push    r12
0x1d3b6:      push    rbp
0x1d3b7:      mov     rbp, rdi
0x1d3ba:      push    rbx
0x1d3bb:      mov     rbx, qword ptr [rip + 0xe77e]
0x1d3c2:      test    rbx, rbx
0x1d3c5:      je      0x1d400
0x1d3c7:      movzx   r12d, byte ptr [rdi]
0x1d3cb:      jmp     0x1d3d9
0x1d3d9:
0x1d3cd:      nop     dword ptr [rax]
0x1d3d0:      mov     rbx, qword ptr [rbx + 8]
0x1d3d4:      test    rbx, rbx
0x1d3d7:      je      0x1d400
0x1d3d9:      cmp     byte ptr [rbx + 0x10], r12b
0x1d3dd:      jne     0x1d3d0
0x1d3df:      lea     rdi, [rbx + 0x10]
0x1d3e3:      mov     rsi, rbp
0x1d3e6:      call    0x4b30

```

```

0x1d3eb:    test    eax, eax
0x1d3ed:    jne     0x1d3d0
0x1d3ef:    mov     r12, rbx
0x1d3f2:    pop     rbx
0x1d3f3:    pop     rbp
0x1d3f4:    mov     rax, r12
0x1d3f7:    pop     r12
0x1d3f9:    ret
0x1d3fa:    nop     word ptr [rax + rax]
0x1d400:    mov     rbx, qword ptr [rip + 0xe731]
0x1d407:    test    rbx, rbx
0x1d40a:    je      0x1d448
0x1d40c:    movzx   r12d, byte ptr [rbp]
0x1d411:    jmp     0x1d421
0x1d421:
0x1d413:    nop     dword ptr [rax + rax]
0x1d418:    mov     rbx, qword ptr [rbx + 8]
0x1d41c:    test    rbx, rbx
0x1d41f:    je      0x1d448
0x1d421:    cmp     byte ptr [rbx + 0x10], r12b
0x1d425:    jne     0x1d418
0x1d427:    lea     rdi, [rbx + 0x10]
0x1d42b:    mov     rsi, rbp
0x1d42e:    call    0x4b30
0x1d433:    test    eax, eax
0x1d435:    jne     0x1d418
0x1d437:    xor     r12d, r12d
0x1d43a:    pop     rbx
0x1d43b:    pop     rbp
0x1d43c:    mov     rax, r12
0x1d43f:    pop     r12
0x1d441:    ret
0x1d442:    nop     word ptr [rax + rax]
0x1d448:    mov     rdi, rbp
0x1d44b:    call    0x4b60
0x1d450:    mov     rdi, rbp
0x1d453:    mov     r12, rax
0x1d456:    call    0x49b0
0x1d45b:    lea     rdi, [rax + 0x11]
0x1d45f:    call    0x1d670
0x1d464:    mov     rsi, rbp
0x1d467:    lea     rdi, [rax + 0x10]
0x1d46b:    mov     rbx, rax
0x1d46e:    call    0x48b0
0x1d473:    test    r12, r12
0x1d476:    je      0x1d4a0
0x1d478:    mov     eax, dword ptr [r12 + 0x10]
0x1d47d:    mov     dword ptr [rbx], eax

```

```

0x1d47f:      mov     rax, qword ptr [rip + 0xe6ba]
0x1d486:      mov     qword ptr [rip + 0xe6b3], rbx
0x1d48d:      mov     qword ptr [rbx + 8], rax
0x1d491:      jmp     0x1d3ef
0x1d3ef:
0x1d496:      nop     word ptr cs:[rax + rax]
0x1d4a0:      mov     rax, qword ptr [rip + 0xe691]
0x1d4a7:      mov     qword ptr [rip + 0xe68a], rbx
0x1d4ae:      mov     qword ptr [rbx + 8], rax
0x1d4b2:      mov     rax, r12
0x1d4b5:      pop     rbx
0x1d4b6:      pop     rbp
0x1d4b7:      pop     r12
0x1d4b9:      ret
0x1d4ba:      nop     word ptr [rax + rax]
0x1d4c0:      endbr64
0x1d4c4:      push    r12
0x1d4c6:      push    rbp
0x1d4c7:      mov     ebp, edi
0x1d4c9:      push    rbx
0x1d4ca:      mov     rbx, qword ptr [rip + 0xe65f]
0x1d4d1:      test    rbx, rbx
0x1d4d4:      jne     0x1d4e9
0x1d4d6:      jmp     0x1d500
0x1d500:
0x1d4d8:      nop     dword ptr [rax + rax]
0x1d4e0:      mov     rbx, qword ptr [rbx + 8]
0x1d4e4:      test    rbx, rbx
0x1d4e7:      je      0x1d500
0x1d4e9:      cmp     dword ptr [rbx], ebp
0x1d4eb:      jne     0x1d4e0
0x1d4ed:      xor     eax, eax
0x1d4ef:      cmp     byte ptr [rbx + 0x10], 0
0x1d4f3:      je      0x1d4f9
0x1d4f5:      lea     rax, [rbx + 0x10]
0x1d4f9:      pop     rbx
0x1d4fa:      pop     rbp
0x1d4fb:      pop     r12
0x1d4fd:      ret
0x1d4fe:      nop
0x1d500:      mov     edi, ebp
0x1d502:      lea     r12, [rip + 0x4edc]
0x1d509:      call    0x4a10
0x1d50e:      mov     edi, 0x11
0x1d513:      test    rax, rax
0x1d516:      je      0x1d527
0x1d518:      mov     r12, qword ptr [rax]
0x1d51b:      mov     rdi, r12

```

```

0x1d51e:      call    0x49b0
0x1d523:      lea     rdi, [rax + 0x11]
0x1d527:      call    0x1d670
0x1d52c:      mov     rsi, r12
0x1d52f:      mov     dword ptr [rax], ebp
0x1d531:      lea     rdi, [rax + 0x10]
0x1d535:      mov     rbx, rax
0x1d538:      call    0x48b0
0x1d53d:      mov     rax, qword ptr [rip + 0xe5ec]
0x1d544:      mov     qword ptr [rip + 0xe5e5], rbx
0x1d54b:      mov     qword ptr [rbx + 8], rax
0x1d54f:      jmp     0x1d4ed
0x1d4ed
0x1d551:      nop     word ptr cs:[rax + rax]
0x1d55c:      nop     dword ptr [rax]
0x1d560:      endbr64
0x1d564:      push    r12
0x1d566:      push    rbp
0x1d567:      mov     rbp, rdi
0x1d56a:      push    rbx
0x1d56b:      mov     rbx, qword ptr [rip + 0xe5be]
0x1d572:      test    rbx, rbx
0x1d575:      je      0x1d5b0
0x1d577:      movzx   r12d, byte ptr [rdi]
0x1d57b:      jmp     0x1d589
0x1d589
0x1d57d:      nop     dword ptr [rax]
0x1d580:      mov     rbx, qword ptr [rbx + 8]
0x1d584:      test    rbx, rbx
0x1d587:      je      0x1d5b0
0x1d589:      cmp     byte ptr [rbx + 0x10], r12b
0x1d58d:      jne     0x1d580
0x1d58f:      lea     rdi, [rbx + 0x10]
0x1d593:      mov     rsi, rbp
0x1d596:      call    0x4b30
0x1d59b:      test    eax, eax
0x1d59d:      jne     0x1d580
0x1d59f:      mov     r12, rbx
0x1d5a2:      pop     rbx
0x1d5a3:      pop     rbp
0x1d5a4:      mov     rax, r12
0x1d5a7:      pop     r12
0x1d5a9:      ret
0x1d5aa:      nop     word ptr [rax + rax]
0x1d5b0:      mov     rbx, qword ptr [rip + 0xe571]
0x1d5b7:      test    rbx, rbx
0x1d5ba:      je      0x1d5f8
0x1d5bc:      movzx   r12d, byte ptr [rbp]

```

```

0x1d5c1:      jmp      0x1d5d1
0x1d5d1
0x1d5c3:      nop      dword ptr [rax + rax]
0x1d5c8:      mov      rbx, qword ptr [rbx + 8]
0x1d5cc:      test     rbx, rbx
0x1d5cf:      je        0x1d5f8
0x1d5d1:      cmp      byte ptr [rbx + 0x10], r12b
0x1d5d5:      jne        0x1d5c8
0x1d5d7:      lea      rdi, [rbx + 0x10]
0x1d5db:      mov      rsi, rbp
0x1d5de:      call     0x4b30
0x1d5e3:      test     eax, eax
0x1d5e5:      jne        0x1d5c8
0x1d5e7:      xor      r12d, r12d
0x1d5ea:      pop      rbx
0x1d5eb:      pop      rbp
0x1d5ec:      mov      rax, r12
0x1d5ef:      pop      r12
0x1d5f1:      ret
0x1d5f2:      nop      word ptr [rax + rax]
0x1d5f8:      mov      rdi, rbp
0x1d5fb:      call     0x4bc0
0x1d600:      mov      rdi, rbp
0x1d603:      mov      r12, rax
0x1d606:      call     0x49b0
0x1d60b:      lea      rdi, [rax + 0x11]
0x1d60f:      call     0x1d670
0x1d614:      mov      rsi, rbp
0x1d617:      lea      rdi, [rax + 0x10]
0x1d61b:      mov      rbx, rax
0x1d61e:      call     0x48b0
0x1d623:      test     r12, r12
0x1d626:      je        0x1d650
0x1d628:      mov      eax, dword ptr [r12 + 0x10]
0x1d62d:      mov      dword ptr [rbx], eax
0x1d62f:      mov      rax, qword ptr [rip + 0xe4fa]
0x1d636:      mov      qword ptr [rip + 0xe4f3], rbx
0x1d63d:      mov      qword ptr [rbx + 8], rax
0x1d641:      jmp      0x1d59f
0x1d59f
0x1d646:      nop      word ptr cs:[rax + rax]
0x1d650:      mov      rax, qword ptr [rip + 0xe4d1]
0x1d657:      mov      qword ptr [rip + 0xe4ca], rbx
0x1d65e:      mov      qword ptr [rbx + 8], rax
0x1d662:      mov      rax, r12
0x1d665:      pop      rbx
0x1d666:      pop      rbp
0x1d667:      pop      r12

```

```

0x1d669:      ret
0x1d66a:      nop      word ptr [rax + rax]
0x1d670:      endbr64
0x1d674:      push     rbx
0x1d675:      mov      rbx, rdi
0x1d678:      call     0x4c20
0x1d67d:      test     rax, rax
0x1d680:      jne      0x1d687
0x1d682:      test     rbx, rbx
0x1d685:      jne      0x1d689
0x1d687:      pop      rbx
0x1d688:      ret
0x1d689:      call     0x18cf0
0x1d68e:      nop
0x1d690:      endbr64
0x1d694:      push     rbx
0x1d695:      mov      rbx, rsi
0x1d698:      call     0x4ce0
0x1d69d:      test     rax, rax
0x1d6a0:      jne      0x1d6a7
0x1d6a2:      test     rbx, rbx
0x1d6a5:      jne      0x1d6a9
0x1d6a7:      pop      rbx
0x1d6a8:      ret
0x1d6a9:      call     0x18cf0
0x1d6ae:      nop
0x1d6b0:      endbr64
0x1d6b4:      push     rbx
0x1d6b5:      mov      rax, rsi
0x1d6b8:      mov      rsi, qword ptr [rsi]
0x1d6bb:      test     rdi, rdi
0x1d6be:      je       0x1d6f0
0x1d6c0:      movabs    rdx, 0xaaaaaaaaaaaaaaaa9
0x1d6ca:      cmp      rsi, rdx
0x1d6cd:      ja       0x1d701
0x1d6cf:      lea      rdx, [rsi + 1]
0x1d6d3:      shr      rdx, 1
0x1d6d6:      add      rsi, rdx
0x1d6d9:      setne    bl
0x1d6dc:      mov      qword ptr [rax], rsi
0x1d6df:      call     0x4ce0
0x1d6e4:      test     rax, rax
0x1d6e7:      jne      0x1d6ed
0x1d6e9:      test     bl, bl
0x1d6eb:      jne      0x1d701
0x1d6ed:      pop      rbx
0x1d6ee:      ret
0x1d6ef:      nop

```



```

0x1d6f0:    mov     ebx, 1
0x1d6f5:    test    rsi, rsi
0x1d6f8:    jne     0x1d6dc
0x1d6fa:    mov     esi, 0x40
0x1d6ff:    jmp     0x1d6dc
0x1d6dc:
0x1d701:    call    0x18cf0
0x1d706:    nop     word ptr cs:[rax + rax]
0x1d710:    endbr64
0x1d714:    push    rbx
0x1d715:    mov     esi, 1
0x1d71a:    mov     rbx, rdi
0x1d71d:    call    0x4b10
0x1d722:    test    rax, rax
0x1d725:    jne     0x1d72c
0x1d727:    test    rbx, rbx
0x1d72a:    jne     0x1d72e
0x1d72c:    pop     rbx
0x1d72d:    ret
0x1d72e:    call    0x18cf0
0x1d733:    nop     word ptr cs:[rax + rax]
0x1d73e:    nop
0x1d740:    endbr64
0x1d744:    sub     rsp, 8
0x1d748:    call    0x4b10
0x1d74d:    test    rax, rax
0x1d750:    je      0x1d757
0x1d752:    add     rsp, 8
0x1d756:    ret
0x1d757:    call    0x18cf0
0x1d75c:    nop     dword ptr [rax]
0x1d760:    endbr64
0x1d764:    push    r12
0x1d766:    mov     r12, rsi
0x1d769:    push    rbp
0x1d76a:    mov     rbp, rdi
0x1d76d:    mov     rdi, rsi
0x1d770:    sub     rsp, 8
0x1d774:    call    0x4c20
0x1d779:    mov     rdi, rax
0x1d77c:    test    rax, rax
0x1d77f:    jne     0x1d786
0x1d781:    test    r12, r12
0x1d784:    jne     0x1d798
0x1d786:    add     rsp, 8
0x1d78a:    mov     rdx, r12
0x1d78d:    mov     rsi, rbp
0x1d790:    pop     rbp

```

```

0x1d791:      pop     r12
0x1d793:      jmp     0x4bb0
0x4bb0
0x1d798:      call    0x18cf0
0x1d79d:      nop     dword ptr [rax]
0x1d7a0:      endbr64
0x1d7a4:      push    r12
0x1d7a6:      push    rbp
0x1d7a7:      mov     rbp, rdi
0x1d7aa:      sub     rsp, 8
0x1d7ae:      call    0x49b0
0x1d7b3:      lea     r12, [rax + 1]
0x1d7b7:      mov     rdi, r12
0x1d7ba:      call    0x4c20
0x1d7bf:      test    rax, rax
0x1d7c2:      je      0x1d7d9
0x1d7c4:      add     rsp, 8
0x1d7c8:      mov     rdx, r12
0x1d7cb:      mov     rsi, rbp
0x1d7ce:      mov     rdi, rax
0x1d7d1:      pop     rbp
0x1d7d2:      pop     r12
0x1d7d4:      jmp     0x4bb0
0x4bb0
0x1d7d9:      call    0x18cf0
0x1d7de:      nop
0x1d7e0:      endbr64
0x1d7e4:      push    r15
0x1d7e6:      mov     r15, rsi
0x1d7e9:      push    r14
0x1d7eb:      mov     r14, rcx
0x1d7ee:      push    r13
0x1d7f0:      mov     r13d, edx
0x1d7f3:      push    r12
0x1d7f5:      mov     r12, r8
0x1d7f8:      push    rbp
0x1d7f9:      mov     rbp, rdi
0x1d7fc:      push    rbx
0x1d7fd:      sub     rsp, 0x28
0x1d801:      movzx   ebx, byte ptr [rdi]
0x1d804:      mov     rax, qword ptr fs:[0x28]
0x1d80d:      mov     qword ptr [rsp + 0x18], rax
0x1d812:      xor     eax, eax
0x1d814:      lea     rax, [rsp + 0x10]
0x1d819:      test    rsi, rsi
0x1d81c:      cmovbe  r15, rax
0x1d820:      call    0x4f00
0x1d825:      mov     rcx, qword ptr [rax]

```

```

0x1d828:    mov     rax, rbp
0x1d82b:    jmp     0x1d838
0x1d838:
0x1d82d:    nop     dword ptr [rax]
0x1d830:    movzx   ebx, byte ptr [rax + 1]
0x1d834:    add     rax, 1
0x1d838:    movzx   edx, bl
0x1d83b:    test    byte ptr [rcx + rdx*2 + 1], 0x20
0x1d840:    jne     0x1d830
0x1d842:    cmp     bl, 0x2d
0x1d845:    je      0x1d8ca
0x1d84b:    call    0x4860
0x1d850:    mov     edx, r13d
0x1d853:    mov     rsi, r15
0x1d856:    mov     rdi, rbp
0x1d859:    mov     dword ptr [rax], 0
0x1d85f:    mov     qword ptr [rsp + 8], rax
0x1d864:    call    0x4dd0
0x1d869:    mov     r13, qword ptr [r15]
0x1d86c:    mov     rcx, qword ptr [rsp + 8]
0x1d871:    mov     rbx, rax
0x1d874:    cmp     r13, rbp
0x1d877:    je      0x1d8d8
0x1d879:    mov     eax, dword ptr [rcx]
0x1d87b:    test    eax, eax
0x1d87d:    jne     0x1d8c0
0x1d87f:    xor     ebp, ebp
0x1d881:    test    r12, r12
0x1d884:    je      0x1d893
0x1d886:    movzx   edx, byte ptr [r13]
0x1d88b:    test    dl, dl
0x1d88d:    jne     0x1d958
0x1d893:    mov     qword ptr [r14], rbx
0x1d896:    mov     rax, qword ptr [rsp + 0x18]
0x1d89b:    sub     rax, qword ptr fs:[0x28]
0x1d8a4:    jne     0x1dbf9
0x1d8aa:    add     rsp, 0x28
0x1d8ae:    mov     eax, ebp
0x1d8b0:    pop     rbx
0x1d8b1:    pop     rbp
0x1d8b2:    pop     r12
0x1d8b4:    pop     r13
0x1d8b6:    pop     r14
0x1d8b8:    pop     r15
0x1d8ba:    ret
0x1d8bb:    nop     dword ptr [rax + rax]
0x1d8c0:    mov     ebp, 1
0x1d8c5:    cmp     eax, 0x22

```

```

0x1d8c8:      je      0x1d881
0x1d8ca:      mov     ebp, 4
0x1d8cf:      jmp     0x1d896
0x1d896:
0x1d8d1:      nop     dword ptr [rax]
0x1d8d8:      test    r12, r12
0x1d8db:      je      0x1d8ca
0x1d8dd:      movzx    edx, byte ptr [rbp]
0x1d8e1:      test    dl, dl
0x1d8e3:      je      0x1d8ca
0x1d8e5:      movsx    esi, dl
0x1d8e8:      mov     rdi, r12
0x1d8eb:      mov     byte ptr [rsp + 8], dl
0x1d8ef:      xor     ebp, ebp
0x1d8f1:      call    0x4a00
0x1d8f6:      mov     ebx, 1
0x1d8fb:      movzx    edx, byte ptr [rsp + 8]
0x1d900:      test    rax, rax
0x1d903:      je      0x1d8ca
0x1d905:      mov     esi, 0x30
0x1d90a:      mov     rdi, r12
0x1d90d:      mov     byte ptr [rsp + 8], dl
0x1d911:      call    0x4a00
0x1d916:      movzx    edx, byte ptr [rsp + 8]
0x1d91b:      test    rax, rax
0x1d91e:      je      0x1d97c
0x1d920:      movzx    eax, byte ptr [r13 + 1]
0x1d925:      cmp     al, 0x44
0x1d927:      je      0x1d9a2
0x1d929:      cmp     al, 0x69
0x1d92b:      je      0x1dbe0
0x1d931:      cmp     al, 0x42
0x1d933:      je      0x1d9a2
0x1d935:      sub     edx, 0x42
0x1d938:      cmp     dl, 0x35
0x1d93b:      ja      0x1d971
0x1d93d:      lea     rcx, [rip + 0x4c0c]
0x1d944:      movzx    edx, dl
0x1d947:      movsxd   rax, dword ptr [rcx + rdx*4]
0x1d94b:      add     rax, rcx
0x1d94e:      jmp     rax
rax
-->-->-->                                indirect JMP
0xe6e7: lea     rdi, [rip + 0x12a22]
0xe6ee: call    0x8b00
0xe6f3: mov     rbp, qword ptr [rbp]
0xe6f7: mov     r13, qword ptr [rbx]
0xe6fa: mov     rsi, rbp

```

```

0xe6fd: mov     rdi, r13
0xe700: add     rsp, 8
0xe704: pop     rbx
0xe705: pop     rbp
0xe706: pop     r12
0xe708: pop     r13
0xe70a: jmp     0x4b30
0x4b30
0xe70f: nop
0xe710: mov     eax, dword ptr [rsi + 0x14]
0xe713: cmp     dword ptr [rdi + 0x14], eax
0xe716: je      0xe6f3
0xe718: mov     rsi, qword ptr [rsi + 0x28]
0xe71c: mov     rdi, qword ptr [rdi + 0x28]
0xe720: jmp     0xe700
0xe700
0xe722: nop     word ptr [rax + rax]
0xe728: mov     r12d, dword ptr [rdi + 0x18]
0xe72c: mov     eax, dword ptr [rsi + 0x18]
0xe72f: cmp     r12d, eax
0xe732: je      0xe6f3
0xe734: add     rsp, 8
0xe738: sub     r12d, eax
0xe73b: pop     rbx
0xe73c: mov     eax, r12d
0xe73f: pop     rbp
0xe740: pop     r12
0xe742: pop     r13
0xe744: ret
0xe745: nop     dword ptr [rax]
0xe748: mov     r12d, dword ptr [rdi + 0x14]
0xe74c: mov     eax, dword ptr [rsi + 0x14]
0xe74f: cmp     r12d, eax
0xe752: jne     0xe734
0xe754: jmp     0xe6f3
0xe6f3
0xe756: nop     word ptr cs:[rax + rax]
0xe760: mov     r12d, dword ptr [rdi + 0x10]
0xe764: mov     eax, dword ptr [rsi + 0x10]
0xe767: cmp     r12d, eax
0xe76a: jne     0xe734
0xe76c: jmp     0xe6f3
0xe6f3
0xe76e: nop
0xe770: mov     r12, qword ptr [rdi + 0x20]
0xe774: mov     rax, qword ptr [rsi + 0x20]
0xe778: cmp     r12, rax
0xe77b: jne     0xe734

```

```

0xe77d: jmp      0xe6f3
0xe6f3
0xe782: nop      word ptr [rax + rax]
0xe788: mov      rdx, qword ptr [rdi + 8]
0xe78c: mov      rax, qword ptr [rsi + 8]
0xe790: mov      r12d, edx
0xe793: sub      r12d, eax
0xe796: cmp      rdx, rax
0xe799: jne      0xe6ab
0xe79f: jmp      0xe6f3
0xe6f3
0xe7a4: nop      dword ptr [rax]
0xe7a8: mov      r13, qword ptr [rdi]
0xe7ab: mov      esi, 0x2e
0xe7b0: mov      rdi, r13
0xe7b3: call     0x4a20
0xe7b8: mov      rbp, qword ptr [rbp]
0xe7bc: mov      esi, 0x2e
0xe7c1: mov      rbx, rax
0xe7c4: mov      rdi, rbp
0xe7c7: call     0x4a20
0xe7cc: test     rbx, rbx
0xe7cf: je       0xe818
0xe7d1: test     rax, rax
0xe7d4: je       0xe818
0xe7d6: lea      rsi, [rax + 1]
0xe7da: lea      rdi, [rbx + 1]
0xe7de: call     0x4b30
0xe7e3: mov      r12d, eax
0xe7e6: test     eax, eax
0xe7e8: je       0xe6fa
0xe7ee: jmp      0xe6ab
0xe6ab
0xe7f3: nop      dword ptr [rax + rax]
0xe7f8: mov      eax, dword ptr [rsi + 0x18]
0xe7fb: cmp      dword ptr [rdi + 0x18], eax
0xe7fe: je       0xe6f3
0xe804: mov      rsi, qword ptr [rsi + 0x30]
0xe808: mov      rdi, qword ptr [rdi + 0x30]
0xe80c: jmp      0xe700
0xe700
0xe811: nop      dword ptr [rax]
0xe818: or       rax, rbx
0xe81b: je       0xe6fa
0xe821: test     rbx, rbx
0xe824: mov      eax, 0xffffffff
0xe829: cmovne   r12d, eax
0xe82d: jmp      0xe6ab

```

```

0xe6ab
0xe832: nop      word ptr cs:[rax + rax]
0xe83d: nop      dword ptr [rax]
0xe840: push     r13
0xe842: mov      ecx, 0x12
0xe847: push     r12
0xe849: push     rbp
0xe84a: mov      rbp, rdi
0xe84d: push     rbx
0xe84e: sub      rsp, 0x138
0xe855: mov      rax, qword ptr fs:[0x28]
0xe85e: mov      qword ptr [rsp + 0x128], rax
0xe866: xor      eax, eax
0xe868: mov      r8, rsp
0xe86b: mov      rdi, r8
0xe86e: rep stosq      qword ptr [rdi], rax
0xe871: movsxd   rax, esi
0xe874: mov      rsi, r8
0xe877: lea      rbx, [rax + rax*4]
0xe87b: mov      rax, qword ptr [rbp + 0x38]
0xe87f: shl      rbx, 4
0xe883: mov      rdi, qword ptr [rax + rbx]
0xe887: call     0x18f90
0xe88c: mov      rax, qword ptr [rbp + 0x38]
0xe890: mov      edx, dword ptr [rsp + 0x18]
0xe894: mov      edi, dword ptr [rsp + 0x1c]
0xe898: mov      ecx, dword ptr [rsp + 0x20]
0xe89c: add      rax, rbx
0xe89f: cmp      qword ptr [rsp + 8], 0
0xe8a5: mov      dword ptr [rax + 0x10], edx
0xe8a8: mov      dword ptr [rax + 0x14], edi
0xe8ab: mov      dword ptr [rax + 0x18], ecx
0xe8ae: je       0xeab0
0xe8b4: mov      ecx, edx
0xe8b6: and      ecx, 0xf000
0xe8bc: cmp      ecx, 0x4000
0xe8c2: je       0xe908
0xe8c4: cmp      ecx, 0x8000
0xe8ca: je       0xeb88
0xe8d0: mov      byte ptr [rax + 0x4a], 0
0xe8d4: cmp      ecx, 0x1000
0xe8da: je       0xeb30
0xe8e0: cmp      ecx, 0xc000
0xe8e6: je       0xebb0
0xe8ec: cmp      ecx, 0xa000
0xe8f2: je       0xebc0
0xe8f8: mov      byte ptr [rax + 0x48], 1
0xe8fc: mov      rdx, qword ptr [rsp + 0x30]

```

```

0xe901: jmp      0xe911
0xe911:
0xe903: nop      dword ptr [rax + rax]
0xe908: mov      byte ptr [rax + 0x48], 0
0xe90c: mov      rdx, qword ptr [rsp + 0x30]
0xe911: mov      qword ptr [rax + 8], rdx
0xe915: call     0x1d310
0xe91a: mov      r12, rax
0xe91d: test     rax, rax
0xe920: je       0xeae4
0xe926: mov      rdi, rax
0xe929: call     0x49b0
0xe92e: mov      r13, qword ptr [rbp + 0x38]
0xe932: cmp      rax, 7
0xe936: lea      rdi, [rax + 1]
0xe93a: mov      eax, 9
0xe93f: cmovbe   rdi, rax
0xe943: add      r13, rbx
0xe946: call     0x1d670
0xe94b: mov      r8, r12
0xe94e: mov      esi, 1
0xe953: lea      rcx, [rip + 0x127c6]
0xe95a: mov      qword ptr [r13 + 0x28], rax
0xe95e: mov      rax, qword ptr [rbp + 0x38]
0xe962: mov      rdx, -1
0xe969: mov      rdi, qword ptr [rax + rbx + 0x28]
0xe96e: xor      eax, eax
0xe970: call     0x4f10
0xe975: mov      edi, dword ptr [rsp + 0x20]
0xe979: call     0x1d4c0
0xe97e: mov      r12, rax
0xe981: test     rax, rax
0xe984: je       0xeb40
0xe98a: mov      rdi, rax
0xe98d: call     0x49b0
0xe992: mov      r13, qword ptr [rbp + 0x38]
0xe996: cmp      rax, 7
0xe99a: lea      rdi, [rax + 1]
0xe99e: mov      eax, 9
0xe9a3: cmovbe   rdi, rax
0xe9a7: add      r13, rbx
0xe9aa: call     0x1d670
0xe9af: mov      r8, r12
0xe9b2: mov      esi, 1
0xe9b7: lea      rcx, [rip + 0x12762]
0xe9be: mov      qword ptr [r13 + 0x30], rax
0xe9c2: mov      rax, qword ptr [rbp + 0x38]
0xe9c6: mov      rdx, -1

```



```

0xe9cd: mov     rdi, qword ptr [rax + rbx + 0x30]
0xe9d2: xor     eax, eax
0xe9d4: call    0x4f10
0xe9d9: mov     rdx, qword ptr [rsp + 0x58]
0xe9de: mov     rax, qword ptr [rbp + 0x38]
0xe9e2: lea     rdi, [rsp + 0x58]
0xe9e7: mov     qword ptr [rax + rbx + 0x20], rdx
0xe9ec: call    0x4840
0xe9f1: mov     rcx, rax
0xe9f4: movsxd  rax, dword ptr [rax + 8]
0xe9f8: mov     r9d, dword ptr [rcx + 0xc]
0xe9fc: mov     rdx, rax
0xe9ff: imul    rax, rax, 0x2aaaaaab
0xea06: mov     esi, edx
0xea08: sar     esi, 0x1f
0xea0b: sar     rax, 0x21
0xea0f: sub     eax, esi
0xea11: lea     esi, [rax + rax*2]
0xea14: mov     eax, edx
0xea16: shl     esi, 2
0xea19: sub     eax, esi
0xea1b: mov     esi, 0xc
0xea20: cmov    eax, esi
0xea23: cmp     edx, 0xc
0xea26: mov     esi, 0x70
0xea2b: mov     edx, 0x61
0xea30: cmovge  edx, esi
0xea33: mov     esi, dword ptr [rcx + 0x10]
0xea36: add     rbx, qword ptr [rbp + 0x38]
0xea3a: lea     rdi, [rbx + 0x38]
0xea3e: push    rdx
0xea3f: mov     edx, dword ptr [rcx + 4]
0xea42: lea     r8d, [rsi + 1]
0xea46: mov     esi, 1
0xea4b: push    rdx
0xea4c: push    rax
0xea4d: movsxd  rax, dword ptr [rcx + 0x14]
0xea51: mov     rdx, rax
0xea54: imul    rax, rax, 0x51eb851f
0xea5b: mov     ecx, edx
0xea5d: sar     ecx, 0x1f
0xea60: sar     rax, 0x25
0xea64: sub     eax, ecx
0xea66: lea     rcx, [rip + 0x126bd]
0xea6d: imul    eax, eax, 0x64
0xea70: sub     edx, eax
0xea72: xor     eax, eax
0xea74: push    rdx

```

```

0xea75: mov     edx, 0x10
0xea7a: call    0x4f10
0xea7f: add     rsp, 0x20
0xea83: mov     rax, qword ptr [rsp + 0x128]
0xea8b: sub     rax, qword ptr fs:[0x28]
0xea94: jne     0xec3e
0xea9a: add     rsp, 0x138
0xeaa1: pop     rbx
0xeaa2: pop     rbp
0xeaa3: pop     r12
0xeaa5: pop     r13
0xeaa7: ret
0xeaa8: nop     dword ptr [rax + rax]
0xeab0: mov     byte ptr [rax + 0x48], 2
0xeab4: mov     rdi, qword ptr [rax]
0xeab7: call    0x18e80
0xeabc: xor     edx, edx
0xeabe: mov     edi, dword ptr [rsp + 0x1c]
0xeac2: cmp     eax, -1
0xeac5: cmovne eax, edx
0xeac8: mov     rdx, qword ptr [rbp + 0x38]
0xeacc: cdq     rdx
0xeace: mov     qword ptr [rdx + rbx + 8], rax
0xead3: call    0x1d310
0xead8: mov     r12, rax
0xeadb: test    rax, rax
0xeade: jne     0xe926
0xeae4: mov     r12, qword ptr [rbp + 0x38]
0xeae8: mov     edi, 0x20
0xeaed: call    0x1d670
0xeaf2: mov     r8d, dword ptr [rsp + 0x1c]
0xeaf7: mov     esi, 1
0xeafc: lea     rcx, [rip + 0x12622]
0xeb03: add     r12, rbx
0xeb06: mov     rdx, -1
0xeb0d: mov     qword ptr [r12 + 0x28], rax
0xeb12: mov     rax, qword ptr [rbp + 0x38]
0xeb16: mov     rdi, qword ptr [rax + rbx + 0x28]
0xeb1b: xor     eax, eax
0xeb1d: call    0x4f10
0xeb22: jmp     0xe975
0xe975:
0xeb27: nop     word ptr [rax + rax]
0xeb30: mov     byte ptr [rax + 0x48], 3
0xeb34: mov     rdx, qword ptr [rsp + 0x30]
0xeb39: jmp     0xe911
0xe911:
0xeb3e: nop

```

```

0xeb40: mov     r12, qword ptr [rbp + 0x38]
0xeb44: mov     edi, 0x20
0xeb49: call    0x1d670
0xeb4e: mov     r8d, dword ptr [rsp + 0x20]
0xeb53: mov     esi, 1
0xeb58: lea     rcx, [rip + 0x125c6]
0xeb5f: add     r12, rbx
0xeb62: mov     rdx, -1
0xeb69: mov     qword ptr [r12 + 0x30], rax
0xeb6e: mov     rax, qword ptr [rbp + 0x38]
0xeb72: mov     rdi, qword ptr [rax + rbx + 0x30]
0xeb77: xor     eax, eax
0xeb79: call    0x4f10
0xeb7e: jmp     0xe9d9
0xe9d9
0xeb83: nop     dword ptr [rax + rax]
0xeb88: xor     ecx, ecx
0xeb8a: and     edx, 0x49
0xeb8d: mov     byte ptr [rax + 0x48], 1
0xeb91: je      0xeb9b
0xeb93: mov     edx, dword ptr [rbp + 0x10]
0xeb96: test    edx, edx
0xeb98: sete    cl
0xeb9b: mov     byte ptr [rax + 0x4a], cl
0xeb9e: mov     rdx, qword ptr [rsp + 0x30]
0xeba3: jmp     0xe911
0xe911
0xeba8: nop     dword ptr [rax + rax]
0xebb0: mov     byte ptr [rax + 0x48], 4
0xebb4: mov     rdx, qword ptr [rsp + 0x30]
0xebb9: jmp     0xe911
0xe911
0xebbe: nop
0xebc0: mov     rdi, qword ptr [rax]
0xebc3: lea     rsi, [rsp + 0x90]
0xebcb: call    0x18f40
0xebd0: cmp     eax, -1
0xebd3: je      0xec12
0xebd5: mov     ecx, dword ptr [rsp + 0xa8]
0xebdc: mov     rax, qword ptr [rbp + 0x38]
0xebe0: mov     edx, ecx
0xebe2: add     rax, rbx
0xebe5: and     edx, 0xf000
0xebeb: cmp     edx, 0x4000
0xebf1: mov     rdx, qword ptr [rsp + 0xc0]
0xebf9: setne   byte ptr [rax + 0x48]
0xebfd: and     ecx, 0x49
0xec00: mov     qword ptr [rsp + 0x30], rdx

```

```

0xec05: setne    byte ptr [rax + 0x4a]
0xec09: mov     edi, dword ptr [rsp + 0x1c]
0xec0d: jmp     0xe911
0xe911:
0xec12: mov     rax, qword ptr [rbp + 0x38]
0xec16: add     rax, rbx
0xec19: mov     byte ptr [rax + 0x48], 2
0xec1d: mov     rdi, qword ptr [rax]
0xec20: call    0x18e80
0xec25: xor     edx, edx
0xec27: cmp     eax, -1
0xec2a: cmov    eax, edx
0xec2d: movsxd  rdx, eax
0xec30: mov     rax, qword ptr [rbp + 0x38]
0xec34: mov     qword ptr [rsp + 0x30], rdx
0xec39: add     rax, rbx
0xec3c: jmp     0xec09
0xec09:
0xec3e: call    0x49d0
0xec43: nop     word ptr cs:[rax + rax]
0xec4e: nop
0xec50: push    r15
0xec52: push    r14
0xec54: push    r13
0xec56: push    r12
0xec58: push    rbp
0xec59: push    rbx
0xec5a: mov     rbx, rdi
0xec5d: sub     rsp, 0x6c8
0xec64: mov     rax, qword ptr fs:[0x28]
0xec6d: mov     qword ptr [rsp + 0x6b8], rax
0xec75: xor     eax, eax
0xec77: lea     rbp, [rsp + 0x1f]
0xec7c: mov     rdi, rbp
0xec7f: call    0xa6a0
0xec84: cmp     dword ptr [rbx + 0x60], 2
0xec88: jg      0xef20
0xec8e: mov     eax, dword ptr [rbx + 0x64]
0xec91: cmp     eax, 0x27
0xec94: jle     0xeeb8
0xec9a: mov     esi, dword ptr [rbx + 0x70]
0xec9d: test    esi, esi
0xec9f: je      0xee38
0xeca5: mov     edx, dword ptr [rbx + 0x68]
0xeca8: test    edx, edx
0xecaa: jle     0xf0e8
0xecb0: sub     edx, 1
0xecb3: mov     rcx, qword ptr [rbx + 0x38]

```

```

0xecb7: xor     edi, edi
0xecb9: lea     rdx, [rdx + rdx*4]
0xecbd: shl     rdx, 4
0xecc1: lea     rax, [rcx + 8]
0xecc5: lea     rdx, [rcx + rdx + 0x58]
0xecca: nop
0xecd0: cmp     byte ptr [rax + 0x41], 0
0xecd4: je      0xece0
0xecd6: cmp     byte ptr [rax + 0x40], 1
0xecda: jne     0xece0
0xecdc: add     rdi, qword ptr [rax]
0xecdf: nop
0xece0: add     rax, 0x50
0xece4: cmp     rax, rdx
0xece7: jne     0xecd0
0xece9: mov     r8d, 1
0xecef: lea     rsi, [rsp + 0x20]
0xecf4: mov     ecx, 1
0xecf9: mov     edx, 0x19c
0xecfe: call    0x1c520
0xed03: cmp     byte ptr [rax], 0x20
0xed06: mov     r8, rax
0xed09: jne     0xed1d
0xed0b: lea     rdx, [rax + 1]
0xed0f: nop
0xed10: mov     r8, rdx
0xed13: add     rdx, 1
0xed17: cmp     byte ptr [rdx - 1], 0x20
0xed1b: je      0xed10
0xed1d: mov     r9d, dword ptr [rbx + 0x70]
0xed21: lea     rdx, [rip + 0x136bd]
0xed28: lea     rax, [rip + 0x126ec]
0xed2f: mov     esi, 1
0xed34: lea     r13, [rsp + 0x2b0]
0xed3c: lea     rcx, [rip + 0x12410]
0xed43: cmp     r9d, 1
0xed47: mov     rdi, r13
0xed4a: cmovle rax, rdx
0xed4e: sub     rsp, 8
0xed52: mov     edx, 0x400
0xed57: push    rax
0xed58: xor     eax, eax
0xed5a: call    0x4f10
0xed5f: mov     edi, dword ptr [rip + 0x1b607]
0xed65: call    0xa530
0xed6a: mov     edi, dword ptr [rip + 0x1b5f8]
0xed70: call    0xa4f0
0xed75: pop     rdx

```

```

0xed76: pop      rcx
0xed77: mov      rdi, r13
0xed7a: call     0x49b0
0xed7f: mov      r8, qword ptr [rbx + 0x20]
0xed83: mov      r12, rax
0xed86: cmp      eax, 8
0xed89: jae      0xee78
0xed8f: test     al, 4
0xed91: jne      0xf0b0
0xed97: test     eax, eax
0xed99: je       0xedb4
0xed9b: movzx    eax, byte ptr [rsp + 0x2b0]
0xeda3: mov      byte ptr [r8], al
0xeda6: test     r12b, 2
0xeda9: jne      0xf0ef
0xedb0: mov      r8, qword ptr [rbx + 0x20]
0xedb4: movsxd   rdx, dword ptr [rbx + 0x64]
0xedb8: lea      rcx, [r12 + 2]
0xedbd: cmp      rcx, rdx
0xedc0: jb       0xef00
0xedc6: mov      rdi, r8
0xedc9: mov      rsi, r12
0xedcc: call     0x184d0
0xedd1: mov      edi, dword ptr [rip + 0x1b549]
0xedd7: call     0xa510
0xeddc: mov      eax, dword ptr [rbx + 0x60]
0xeddf: mov      rdi, qword ptr [rbx + 8]
0xede3: mov      edx, 2
0xede8: lea      esi, [rax - 1]
0xede9: call     0xc680
0xedf0: mov      eax, dword ptr [rbx + 0x64]
0xedf3: mov      rdi, qword ptr [rbx + 8]
0xedf7: mov      rsi, qword ptr [rbx + 0x20]
0xedfb: lea      edx, [rax - 4]
0xedfe: call     0xc600
0xee03: mov      rdi, rbp
0xee06: call     0xa6b0
0xee0b: mov      rax, qword ptr [rsp + 0x6b8]
0xee13: sub      rax, qword ptr fs:[0x28]
0xee1c: jne      0xf107
0xee22: add      rsp, 0x6c8
0xee29: pop      rbx
0xee2a: pop      rbp
0xee2b: pop      r12
0xee2d: pop      r13
0xee2f: pop      r14
0xee31: pop      r15
0xee33: ret

```

```

0xee34: nop        dword ptr [rax]
0xee38: mov         eax, dword ptr [rip + 0x1c7e2]
0xee3e: test        eax, eax
0xee40: jne         0xefc0
0xee46: mov         byte ptr [rsp + 0x2b0], 0
0xee4e: lea         r13, [rsp + 0x2b0]
0xee56: mov         edi, dword ptr [rip + 0x1b508]
0xee5c: call        0xa530
0xee61: mov         edi, dword ptr [rip + 0x1b4f9]
0xee67: call        0xa4f0
0xee6c: jmp         0xed77
0xed77:
0xee71: nop        dword ptr [rax]
0xee78: mov         rax, qword ptr [rsp + 0x2b0]
0xee80: lea         rdi, [r8 + 8]
0xee84: mov         rsi, r13
0xee87: and         rdi, 0xfffffffffffffff8
0xee8b: mov         qword ptr [r8], rax
0xee8e: mov         eax, r12d
0xee91: mov         rdx, qword ptr [r13 + rax - 8]
0xee96: mov         qword ptr [r8 + rax - 8], rdx
0xee9b: sub         r8, rdi
0xee9e: sub         rsi, r8
0xeea1: add         r8d, r12d
0xeea4: shr         r8d, 3
0xeea8: mov         ecx, r8d
0xeeab: rep movsq   qword ptr [rdi], qword ptr [rsi]
0xeeae: jmp         0xedb0
0xedb0:
0xeeb3: nop        dword ptr [rax + rax]
0xeeb8: cmp         eax, 0xd
0xeebb: jle         0xee46
0xeebd: mov         esi, dword ptr [rbx + 0x28]
0xeec0: lea         r12, [rsp + 0x20]
0xeec5: mov         rdi, rbx
0xeec8: xor         eax, eax
0xeeca: mov         rdx, r12
0xeecd: lea         r13, [rsp + 0x2b0]
0xeed5: call        0xe560
0xeeda: mov         r8, r12
0xeedd: mov         edx, 0x400
0xeeee2: mov         rdi, r13
0xeeee5: lea         rcx, [rip + 0x12289]
0xeeec: mov         esi, 1
0xeef1: xor         eax, eax
0xeef3: call        0x4f10
0xeef8: jmp         0xee56
0xee56:

```

```

0xeefd: nop      dword ptr [rax]
0xef00: lea      edx, [rdx - 2]
0xef03: lea      rdi, [r8 + r12]
0xef07: mov      esi, 0x20
0xef0c: movsxd   rdx, edx
0xef0f: sub      rdx, r12
0xef12: call     0x4a50
0xef17: mov      r8, qword ptr [rbx + 0x20]
0xef1b: jmp      0xedc6
0xedc6
0xef20: mov      esi, dword ptr [rip + 0x1b3fa]
0xef26: xor      edi, edi
0xef28: mov      edx, esi
0xef2a: call     0xa570
0xef2f: mov      eax, dword ptr [rbx + 0x64]
0xef32: mov      rdi, qword ptr [rbx + 8]
0xef36: lea      edx, [rax - 1]
0xef39: mov      eax, dword ptr [rbx + 0x8c]
0xef3f: lea      esi, [rax + 1]
0xef42: call     0xc680
0xef47: mov      rdi, qword ptr [rbx + 8]
0xef4b: mov      esi, 0x20
0xef50: call     0xc650
0xef55: mov      eax, dword ptr [rbx + 0x60]
0xef58: mov      ecx, dword ptr [rbx + 0x68]
0xef5b: lea      edx, [rax - 1]
0xef5e: cmp      edx, ecx
0xef60: jg       0xec8e
0xef66: sub      eax, 2
0xef69: imul     eax, dword ptr [rbx + 0x28]
0xef6d: mov      esi, dword ptr [rip + 0x1b3b5]
0xef73: mov      edi, dword ptr [rip + 0x1b3b7]
0xef79: cdq
0xef7a: idiv     ecx
0xef7c: mov      edx, dword ptr [rip + 0x1b3aa]
0xef82: mov      dword ptr [rbx + 0x8c], eax
0xef88: call     0xa570
0xef8d: mov      eax, dword ptr [rbx + 0x64]
0xef90: mov      rdi, qword ptr [rbx + 8]
0xef94: lea      edx, [rax - 1]
0xef97: mov      eax, dword ptr [rbx + 0x8c]
0xef9d: lea      esi, [rax + 1]
0xefa0: call     0xc680
0xefa5: mov      rdi, qword ptr [rbx + 8]
0xefa9: mov      esi, 0x23
0xefae: call     0xc650
0xefb3: jmp      0xec8e
0xec8e

```



```

0xefb8: nop      dword ptr [rax + rax]
0xefc0: lea       r9, [rsp + 0x20]
0xefc5: mov      esi, dword ptr [rbx + 0x28]
0xefc8: xor      eax, eax
0xefca: mov      rdi, rbx
0xefcd: mov      rdx, r9
0xefd0: mov      qword ptr [rsp + 8], r9
0xefd5: lea      r13, [rsp + 0x2b0]
0xefdd: call     0xe560
0xefe2: mov      eax, dword ptr [rbx + 0x64]
0xefe5: lea      r12d, [rax - 0x1a]
0xefe9: movsxd   rax, dword ptr [rbx + 0x28]
0xefed: movsxd   r12, r12d
0xeff0: lea      r15, [rax + rax*4]
0xeff4: shl      r15, 4
0xeff8: add      r15, qword ptr [rbx + 0x38]
0xeffc: mov      rsi, qword ptr [r15]
0xefff: mov      rdi, rsi
0xf002: mov      qword ptr [rsp], rsi
0xf006: call     0x49b0
0xf00b: mov      rsi, qword ptr [rsp]
0xf00f: mov      ecx, 0x400
0xf014: mov      rdi, r13
0xf017: cmp      r12, rax
0xf01a: cmovbe   rax, r12
0xf01e: mov      rdx, rax
0xf021: mov      r14, rax
0xf024: call     0x4b70
0xf029: mov      rdx, r12
0xf02c: lea      rdi, [r13 + r14]
0xf031: mov      esi, 0x20
0xf036: sub      rdx, r14
0xf039: call     0x4a50
0xf03e: cmp      byte ptr [r15 + 0x48], 0
0xf043: mov      r9, qword ptr [rsp + 8]
0xf048: jne      0xf088
0xf04a: mov      rax, qword ptr [r15]
0xf04d: cmp      byte ptr [rax], 0x2e
0xf050: jne      0xf058
0xf052: cmp      byte ptr [rax + 1], 0x2e
0xf056: je       0xf0d0
0xf058: lea      r8, [rip + 0x120ec]
0xf05f: lea      rdi, [r13 + r12]
0xf064: lea      rcx, [rip + 0x120f8]
0xf06b: mov      rdx, -1
0xf072: xor      eax, eax
0xf074: mov      esi, 1
0xf079: call     0x4f10

```

```

0xf07e: jmp      0xee56
0xee56:
0xf083: nop      dword ptr [rax + rax]
0xf088: mov      r8, qword ptr [r15 + 8]
0xf08c: lea      rdi, [r13 + r12]
0xf091: mov      esi, 1
0xf096: xor      eax, eax
0xf098: lea      rcx, [rip + 0x120cf]
0xf09f: mov      rdx, -1
0xf0a6: call     0x4f10
0xf0ab: jmp      0xee56
0xee56:
0xf0b0: mov      eax, dword ptr [rsp + 0x2b0]
0xf0b7: mov      dword ptr [r8], eax
0xf0ba: mov      eax, r12d
0xf0bd: mov      edx, dword ptr [r13 + rax - 4]
0xf0c2: mov      dword ptr [r8 + rax - 4], edx
0xf0c7: mov      r8, qword ptr [rbx + 0x20]
0xf0cb: jmp      0xedb4
0xedb4:
0xf0d0: cmp      byte ptr [rax + 2], 0
0xf0d4: lea      r8, [rip + 0x12068]
0xf0db: je       0xf05f
0xf0dd: jmp      0xf058
0xf058:
0xf0e2: nop      word ptr [rax + rax]
0xf0e8: xor      edi, edi
0xf0ea: jmp      0xece9
0xece9:
0xf0ef: mov      eax, r12d
0xf0f2: movzx    edx, word ptr [r13 + rax - 2]
0xf0f8: mov      word ptr [r8 + rax - 2], dx
0xf0fe: mov      r8, qword ptr [rbx + 0x20]
0xf102: jmp      0xedb4
0xedb4:
0xf107: call     0x49d0
0xf10c: nop      dword ptr [rax]
0xf110: mov      eax, dword ptr [rdi + 0x2c]
0xf113: mov      edx, dword ptr [rdi + 0x28]
0xf116: mov      ecx, dword ptr [rdi + 0x60]
0xf119: mov      esi, dword ptr [rdi + 0x68]
0xf11c: cmp      eax, edx
0xf11e: jg       0xf130
0xf120: lea      edi, [rcx + rax - 2]
0xf124: cmp      edx, edi
0xf126: jge      0xf130
0xf128: cmp      edi, esi
0xf12a: jle      0xf159

```

```

0xf12c: nop      dword ptr [rax]
0xf130: sub      ecx, 2
0xf133: mov      eax, ecx
0xf135: shr      eax, 0x1f
0xf138: add      eax, ecx
0xf13a: sar      eax, 1
0xf13c: sub      edx, eax
0xf13e: mov      eax, 0
0xf143: lea      edi, [rcx + rdx]
0xf146: cmovns   eax, edx
0xf149: cmp      esi, edi
0xf14b: jg       0xf159
0xf14d: mov      eax, esi
0xf14f: mov      edx, 0
0xf154: sub      eax, ecx
0xf156: cmovs    eax, edx
0xf159: ret
0xf15a: nop      word ptr [rax + rax]
0xf160: endbr64
0xf164: push     r12
0xf166: push     rbp
0xf167: mov      rbp, rdi
0xf16a: mov      edi, 0x98
0xf16f: sub      rsp, 8
0xf173: call     0x1d670
0xf178: mov      ecx, 0x100
0xf17d: mov      rdi, rbp
0xf180: mov      dword ptr [rax + 0x50], 0xffffffff
0xf187: mov      r12, rax
0xf18a: mov      qword ptr [rax + 0x20], 0
0xf192: mov      qword ptr [rax + 0x28], 0
0xf19a: mov      qword ptr [rax], 0
0xf1a1: mov      qword ptr [rax + 0x48], 0
0xf1a9: mov      qword ptr [rax + 0x58], 0
0xf1b1: movabs   rax, 0x100000001
0xf1bb: mov      qword ptr [r12 + 0x60], rax
0xf1c0: movabs   rax, 0x3fffffff00000000
0xf1ca: mov      dword ptr [r12 + 0x68], 0
0xf1d3: mov      word ptr [r12 + 0x6c], cx
0xf1d9: mov      qword ptr [r12 + 0x80], 0
0xf1e5: mov      qword ptr [r12 + 0x88], 1
0xf1f1: mov      qword ptr [r12 + 0x70], 0
0xf1fa: mov      qword ptr [r12 + 0x10], rax
0xf1ff: mov      qword ptr [r12 + 0x38], 0
0xf208: call     0x49c0
0xf20d: cmp      eax, -1
0xf210: je       0xf3b0
0xf216: xor      eax, eax

```

```

0xf218: call    0x18fe0
0xf21d: mov     qword ptr [r12 + 0x18], rax
0xf222: mov     rdi, rax
0xf225: test    rax, rax
0xf228: je      0xf3c8
0xf22e: call    0x18040
0xf233: mov     rdi, qword ptr [r12 + 0x18]
0xf238: call    0x49b0
0xf23d: mov     qword ptr [r12 + 0x78], rax
0xf242: xor     eax, eax
0xf244: call    0xc590
0xf249: mov     edx, dword ptr [rip + 0x1c3b1]
0xf24f: mov     qword ptr [r12 + 8], rax
0xf254: test    edx, edx
0xf256: je      0xf280
0xf258: mov     eax, dword ptr [rip + 0x1c3d6]
0xf25e: mov     dword ptr [r12 + 0x80], eax
0xf266: mov     eax, dword ptr [rip + 0x1c3c4]
0xf26c: mov     dword ptr [r12 + 0x84], eax
0xf274: add     rsp, 8
0xf278: mov     rax, r12
0xf27b: pop     rbp
0xf27c: pop     r12
0xf27e: ret
0xf27f: nop
0xf280: lea     rdi, [rip + 0x11406]
0xf287: call    0x17c50
0xf28c: xor     esi, esi
0xf28e: lea     rdi, [rip + 0x11ee5]
0xf295: call    0x17d10
0xf29a: lea     rdi, [rip + 0x11411]
0xf2a1: mov     dword ptr [rip + 0x1c385], eax
0xf2a7: call    0x17c50
0xf2ac: xor     ecx, ecx
0xf2ae: mov     edx, 6
0xf2b3: lea     rsi, [rip + 0x1b326]
0xf2ba: lea     rdi, [rip + 0x11ecb]
0xf2c1: call    0x17d70
0xf2c6: xor     ecx, ecx
0xf2c8: mov     edx, 9
0xf2cd: lea     rsi, [rip + 0x1b2ac]
0xf2d4: mov     dword ptr [r12 + 0x80], eax
0xf2dc: lea     rdi, [rip + 0x11ec0]
0xf2e3: mov     dword ptr [rip + 0x1c34b], eax
0xf2e9: call    0x17d70
0xf2ee: mov     esi, 1
0xf2f3: lea     rdi, [rip + 0x11ebf]
0xf2fa: mov     dword ptr [r12 + 0x84], eax

```

```

0xf302: mov     dword ptr [rip + 0x1c328], eax
0xf308: call    0x17e50
0xf30d: mov     esi, 1
0xf312: lea     rdi, [rip + 0x11eac]
0xf319: mov     dword ptr [rip + 0x1c301], eax
0xf31f: call    0x17e50
0xf324: lea     rdi, [rip + 0x1a91d]
0xf32b: mov     dword ptr [rip + 0x1c2f3], eax
0xf331: mov     eax, dword ptr [rip + 0x1a929]
0xf337: test    eax, eax
0xf339: lea     rax, [rip + 0x1a8f0]
0xf340: cmov    rdi, rax
0xf344: call    0x17c50
0xf349: mov     edx, 0x14
0xf34e: lea     rsi, [rip + 0x1b18b]
0xf355: lea     rdi, [rip + 0x1afc4]
0xf35c: call    0x17f70
0xf361: mov     edi, 8
0xf366: call    0x19100
0xf36b: mov     edi, 8
0xf370: mov     qword ptr [rip + 0x1c2a1], rax
0xf377: call    0x19100
0xf37c: mov     edi, 8
0xf381: mov     qword ptr [rip + 0x1c288], rax
0xf388: call    0x19100
0xf38d: mov     dword ptr [rip + 0x1c269], 1
0xf397: mov     qword ptr [rip + 0x1c26a], rax
0xf39e: add     rsp, 8
0xf3a2: mov     rax, r12
0xf3a5: pop     rbp
0xf3a6: pop     r12
0xf3a8: ret
0xf3a9: nop     dword ptr [rax]
0xf3b0: lea     rdi, [rip + 0x12641]
0xf3b7: call    0x8b00
0xf3bc: jmp     0xf216
0xf216
0xf3c1: nop     dword ptr [rax]
0xf3c8: lea     rdi, [rip + 0x11581]
0xf3cf: call    0x8b00
0xf3d4: mov     rdi, qword ptr [r12 + 0x18]
0xf3d9: jmp     0xf22e
0xf22e
0xf3de: nop
0xf3e0: endbr64
0xf3e4: push    r12
0xf3e6: mov     r12, rdi
0xf3e9: push    rbp

```

```

0xf3ea: push    rbx
0xf3eb: mov     rdi, qword ptr [rdi]
0xf3ee: test    rdi, rdi
0xf3f1: je      0xf3f8
0xf3f3: call    0x4ac0
0xf3f8: mov     eax, dword ptr [r12 + 0x68]
0xf3fd: mov     r8, qword ptr [r12 + 0x38]
0xf402: test    eax, eax
0xf404: jle     0xf43e
0xf406: xor     ebx, ebx
0xf408: nop     dword ptr [rax + rax]
0xf410: lea     rbp, [rbx + rbx*4]
0xf414: shl     rbp, 4
0xf418: mov     rdi, qword ptr [r8 + rbp]
0xf41c: test    rdi, rdi
0xf41f: je      0xf433
0xf421: call    0x18d10
0xf426: mov     r8, qword ptr [r12 + 0x38]
0xf42b: mov     qword ptr [r8 + rbp], 0
0xf433: add     rbx, 1
0xf437: cmp     dword ptr [r12 + 0x68], ebx
0xf43c: jg      0xf410
0xf43e: mov     rdi, r8
0xf441: call    0x18d10
0xf446: mov     rdi, qword ptr [r12 + 0x20]
0xf44b: call    0x18d10
0xf450: mov     rdi, qword ptr [r12 + 8]
0xf455: call    0xc5c0
0xf45a: pop     rbx
0xf45b: mov     rdi, r12
0xf45e: pop     rbp
0xf45f: pop     r12
0xf461: jmp     0x18d10
0x18d10:
0xf466: nop     word ptr cs:[rax + rax]
0xf470: endbr64
0xf474: mov     rax, qword ptr [rdi + 8]
0xf478: ret
0xf479: nop     dword ptr [rax]
0xf480: endbr64
0xf484: mov     dword ptr [rdi + 0x14], 0x3fffffff
0xf48b: ret
0xf48c: nop     dword ptr [rax]
0xf490: endbr64
0xf494: movsxd  rax, dword ptr [rdi + 0x28]
0xf498: lea     rax, [rax + rax*4]
0xf49c: shl     rax, 4
0xf4a0: add     rax, qword ptr [rdi + 0x38]

```

```

0xf4a4: mov     rax, qword ptr [rax]
0xf4a7: ret
0xf4a8: nop     dword ptr [rax + rax]
0xf4b0: endbr64
0xf4b4: movsxd  rax, dword ptr [rdi + 0x28]
0xf4b8: lea     rax, [rax + rax*4]
0xf4bc: shl     rax, 4
0xf4c0: add     rax, qword ptr [rdi + 0x38]
0xf4c4: mov     eax, dword ptr [rax + 0x14]
0xf4c7: ret
0xf4c8: nop     dword ptr [rax + rax]
0xf4d0: endbr64
0xf4d4: movsxd  rax, dword ptr [rdi + 0x28]
0xf4d8: lea     rax, [rax + rax*4]
0xf4dc: shl     rax, 4
0xf4e0: add     rax, qword ptr [rdi + 0x38]
0xf4e4: mov     eax, dword ptr [rax + 0x18]
0xf4e7: ret
0xf4e8: nop     dword ptr [rax + rax]
0xf4f0: endbr64
0xf4f4: movsxd  rax, dword ptr [rdi + 0x28]
0xf4f8: lea     rax, [rax + rax*4]
0xf4fc: shl     rax, 4
0xf500: add     rax, qword ptr [rdi + 0x38]
0xf504: mov     eax, dword ptr [rax + 0x10]
0xf507: ret
0xf508: nop     dword ptr [rax + rax]
0xf510: endbr64
0xf514: movsxd  rax, dword ptr [rdi + 0x28]
0xf518: lea     rax, [rax + rax*4]
0xf51c: shl     rax, 4
0xf520: add     rax, qword ptr [rdi + 0x38]
0xf524: movsx   eax, byte ptr [rax + 0x48]
0xf528: ret
0xf529: nop     dword ptr [rax]
0xf530: endbr64
0xf534: mov     eax, dword ptr [rdi + 0x60]
0xf537: mov     esi, dword ptr [rdi + 0x68]
0xf53a: mov     r8d, 0
0xf540: lea     edx, [rax - 2]
0xf543: mov     eax, dword ptr [rdi + 0x28]
0xf546: mov     ecx, edx
0xf548: shr     ecx, 0x1f
0xf54b: add     ecx, edx
0xf54d: sar     ecx, 1
0xf54f: sub     eax, ecx
0xf551: lea     ecx, [rdx + rax]
0xf554: cmovs   eax, r8d

```

```

0xf558: cmp     esi, ecx
0xf55a: jg       0xf568
0xf55c: mov     eax, esi
0xf55e: sub     eax, edx
0xf560: mov     edx, 0
0xf565: cmovs   eax, edx
0xf568: mov     dword ptr [rdi + 0x2c], eax
0xf56b: ret
0xf56c: nop     dword ptr [rax]
0xf570: endbr64
0xf574: mov     byte ptr [rdi + 0x6d], 1
0xf578: ret
0xf579: nop     dword ptr [rax]
0xf580: endbr64
0xf584: mov     byte ptr [rdi + 0x6d], 0
0xf588: ret
0xf589: nop     dword ptr [rax]
0xf590: endbr64
0xf594: mov     eax, dword ptr [rip + 0x1c092]
0xf59a: push    r14
0xf59c: mov     r14d, ecx
0xf59f: push    r13
0xf5a1: mov     r13d, edx
0xf5a4: push    r12
0xf5a6: mov     r12d, esi
0xf5a9: push    rbp
0xf5aa: mov     rbp, r8
0xf5ad: push    rbx
0xf5ae: mov     rbx, rdi
0xf5b1: mov     dword ptr [rdi + 0x58], esi
0xf5b4: mov     dword ptr [rdi + 0x5c], edx
0xf5b7: mov     dword ptr [rdi + 0x60], ecx
0xf5ba: mov     dword ptr [rdi + 0x64], r8d
0xf5be: mov     dword ptr [rdi + 0x8c], 0
0xf5c8: test    eax, eax
0xf5ca: jle     0xf648
0xf5cc: lea     edx, [rcx - 1]
0xf5cf: cmp     eax, edx
0xf5d1: jge     0xf648
0xf5d3: mov     dword ptr [rbx + 0x88], eax
0xf5d9: mov     rdi, qword ptr [rbx + 0x20]
0xf5dd: movsxd  rsi, ebp
0xf5e0: call    0x1d690
0xf5e5: mov     rdi, qword ptr [rbx + 8]
0xf5e9: mov     ecx, r14d
0xf5ec: mov     edx, r13d
0xf5ef: mov     qword ptr [rbx + 0x20], rax
0xf5f3: mov     esi, r12d

```



```

0xf5f6: mov     r8d, ebp
0xf5f9: call    0xc5e0
0xf5fe: mov     eax, dword ptr [rbx + 0x60]
0xf601: mov     esi, dword ptr [rbx + 0x68]
0xf604: mov     edi, 0
0xf609: lea     edx, [rax - 2]
0xf60c: mov     eax, dword ptr [rbx + 0x28]
0xf60f: mov     ecx, edx
0xf611: shr     ecx, 0x1f
0xf614: add     ecx, edx
0xf616: sar     ecx, 1
0xf618: sub     eax, ecx
0xf61a: lea     ecx, [rdx + rax]
0xf61d: cmovs   eax, edi
0xf620: cmp     esi, ecx
0xf622: jg      0xf630
0xf624: mov     eax, esi
0xf626: sub     eax, edx
0xf628: mov     edx, 0
0xf62d: cmovs   eax, edx
0xf630: mov     dword ptr [rbx + 0x2c], eax
0xf633: mov     dword ptr [rbx + 0x74], 0
0xf63a: pop     rbx
0xf63b: pop     rbp
0xf63c: pop     r12
0xf63e: pop     r13
0xf640: pop     r14
0xf642: ret
0xf643: nop     dword ptr [rax + rax]
0xf648: mov     eax, ecx
0xf64a: shr     eax, 0x1f
0xf64d: add     eax, ecx
0xf64f: sar     eax, 1
0xf651: jmp     0xf5d3
0xf5d3:
0xf653: nop     word ptr cs:[rax + rax]
0xf65e: nop
0xf660: endbr64
0xf664: mov     byte ptr [rdi + 0x6e], sil
0xf668: ret
0xf669: nop     dword ptr [rax]
0xf670: endbr64
0xf674: push    r13
0xf676: push    r12
0xf678: push    rbp
0xf679: push    rbx
0xf67a: sub     rsp, 8
0xf67e: test    ecx, ecx

```

```

0xf680: js      0xf6ce
0xf682: movsxd   rax, ecx
0xf685: mov      rbp, rsi
0xf688: mov      r13, rdx
0xf68b: mov      r12d, ecx
0xf68e: lea      rbx, [rax + rax*4]
0xf692: shl      rbx, 4
0xf696: add      rbx, qword ptr [rdi + 0x38]
0xf69a: jmp      0xf6ae
0xf6ae:
0xf69c: nop      dword ptr [rax]
0xf6a0: sub      r12d, 1
0xf6a4: sub      rbx, 0x50
0xf6a8: cmp      r12d, -1
0xf6ac: je       0xf6c0
0xf6ae: mov      rsi, qword ptr [rbx]
0xf6b1: mov      rdx, r13
0xf6b4: mov      rdi, rbp
0xf6b7: call     0x4c30
0xf6bc: test     eax, eax
0xf6be: jne      0xf6a0
0xf6c0: add      rsp, 8
0xf6c4: mov      eax, r12d
0xf6c7: pop      rbx
0xf6c8: pop      rbp
0xf6c9: pop      r12
0xf6cb: pop      r13
0xf6cd: ret
0xf6ce: mov      r12d, 0xffffffff
0xf6d4: jmp      0xf6c0
0xf6c0:
0xf6d6: nop      word ptr cs:[rax + rax]
0xf6e0: endbr64
0xf6e4: push     r14
0xf6e6: push     r13
0xf6e8: push     r12
0xf6ea: push     rbp
0xf6eb: push     rbx
0xf6ec: mov      r14d, dword ptr [rdi + 0x68]
0xf6f0: cmp      ecx, r14d
0xf6f3: jge      0xf740
0xf6f5: movsxd   rax, ecx
0xf6f8: mov      rbp, rsi
0xf6fb: mov      r13, rdx
0xf6fe: mov      r12d, ecx
0xf701: lea      rbx, [rax + rax*4]
0xf705: shl      rbx, 4
0xf709: add      rbx, qword ptr [rdi + 0x38]

```

```

0xf70d: jmp      0xf71d
0xf71d
0xf70f: nop
0xf710: add     r12d, 1
0xf714: add     rbx, 0x50
0xf718: cmp     r12d, r14d
0xf71b: je      0xf740
0xf71d: mov     rsi, qword ptr [rbx]
0xf720: mov     rdx, r13
0xf723: mov     rdi, rbp
0xf726: call    0x4c30
0xf72b: test    eax, eax
0xf72d: jne     0xf710
0xf72f: pop     rbx
0xf730: mov     eax, r12d
0xf733: pop     rbp
0xf734: pop     r12
0xf736: pop     r13
0xf738: pop     r14
0xf73a: ret
0xf73b: nop     dword ptr [rax + rax]
0xf740: mov     r12d, 0xffffffff
0xf746: pop     rbx
0xf747: pop     rbp
0xf748: mov     eax, r12d
0xf74b: pop     r12
0xf74d: pop     r13
0xf74f: pop     r14
0xf751: ret
0xf752: nop     word ptr cs:[rax + rax]
0xf75d: nop     dword ptr [rax]
0xf760: endbr64
0xf764: push    r14
0xf766: mov     r14d, ecx
0xf769: push    r13
0xf76b: mov     r13, rdx
0xf76e: push    r12
0xf770: mov     r12, rdi
0xf773: push    rbp
0xf774: push    rbx
0xf775: mov     rbx, rsi
0xf778: call    0x49b0
0xf77d: mov     rdi, rbx
0xf780: mov     rbp, rax
0xf783: call    0x49b0
0xf788: lea     rdi, [rbp + rax + 1]
0xf78d: call    0x1d670
0xf792: mov     r8, rbx

```

```

0xf795: mov     rcx, r12
0xf798: mov     esi, 1
0xf79d: mov     rbp, rax
0xf7a0: mov     rdi, rax
0xf7a3: mov     rdx, -1
0xf7aa: xor     eax, eax
0xf7ac: call    0x4f10
0xf7b1: mov     edx, r14d
0xf7b4: mov     rsi, r13
0xf7b7: mov     rdi, rbp
0xf7ba: call    0x8070
0xf7bf: mov     rdi, rbp
0xf7c2: mov     r12d, eax
0xf7c5: call    0x18d10
0xf7ca: pop     rbx
0xf7cb: mov     eax, r12d
0xf7ce: pop     rbp
0xf7cf: pop     r12
0xf7d1: pop     r13
0xf7d3: pop     r14
0xf7d5: ret
0xf7d6: nop     word ptr cs:[rax + rax]
0xf7e0: endbr64
0xf7e4: push    r15
0xf7e6: mov     r15d, r8d
0xf7e9: push    r14
0xf7eb: mov     r14, rcx
0xf7ee: push    r13
0xf7f0: mov     r13, rdx
0xf7f3: push    r12
0xf7f5: mov     r12, rdi
0xf7f8: push    rbp
0xf7f9: push    rbx
0xf7fa: mov     rbx, rsi
0xf7fd: sub     rsp, 0x18
0xf801: call    0x49b0
0xf806: mov     rdi, rbx
0xf809: mov     qword ptr [rsp + 8], rax
0xf80e: call    0x49b0
0xf813: mov     rdi, r13
0xf816: mov     rbp, rax
0xf819: call    0x49b0
0xf81e: mov     rdx, qword ptr [rsp + 8]
0xf823: lea     rdi, [rdx + rbp + 1]
0xf828: add     rdi, rax
0xf82b: call    0x1d670
0xf830: mov     r9, r13
0xf833: mov     r8, rbx

```

```

0xf836: mov     rcx, r12
0xf839: mov     rbp, rax
0xf83c: mov     rdi, rax
0xf83f: mov     esi, 1
0xf844: xor     eax, eax
0xf846: mov     rdx, -1
0xf84d: call    0x4f10
0xf852: mov     edx, r15d
0xf855: mov     rsi, r14
0xf858: mov     rdi, rbp
0xf85b: call    0x8070
0xf860: mov     rdi, rbp
0xf863: mov     r12d, eax
0xf866: call    0x18d10
0xf86b: add     rsp, 0x18
0xf86f: mov     eax, r12d
0xf872: pop     rbx
0xf873: pop     rbp
0xf874: pop     r12
0xf876: pop     r13
0xf878: pop     r14
0xf87a: pop     r15
0xf87c: ret
0xf87d: nop     dword ptr [rax]
0xf880: endbr64
0xf884: push    r15
0xf886: mov     r15, rdi
0xf889: mov     rdi, rsi
0xf88c: push    r14
0xf88e: push    r13
0xf890: push    r12
0xf892: push    rbp
0xf893: mov     rbp, rsi
0xf896: push    rbx
0xf897: mov     ebx, edx
0xf899: sub     rsp, 0x198
0xf8a0: mov     rax, qword ptr fs:[0x28]
0xf8a9: mov     qword ptr [rsp + 0x188], rax
0xf8b1: xor     eax, eax
0xf8b3: call    0x4950
0xf8b8: mov     r12, rax
0xf8bb: test    rax, rax
0xf8be: je      0xfda8
0xf8c4: mov     rdi, rbp
0xf8c7: call    0x49c0
0xf8cc: cmp     eax, -1
0xf8cf: je      0x10050
0xf8d5: mov     rdi, qword ptr [r15]

```

```

0xf8d8: test    rdi, rdi
0xf8db: je        0xf8e2
0xf8dd: call     0x4ac0
0xf8e2: mov     rax, qword ptr [r15 + 0x78]
0xf8e6: mov     qword ptr [r15], r12
0xf8e9: lea     rdi, [rax + 1]
0xf8ed: call     0x1d670
0xf8f2: mov     rsi, qword ptr [r15 + 0x18]
0xf8f6: mov     rdi, rax
0xf8f9: mov     qword ptr [rsp + 0x18], rax
0xf8fe: call     0x48b0
0xf903: cmp     byte ptr [rbp], 0x2f
0xf907: je      0xf9b20
0xf90d: xor     eax, eax
0xf90f: call     0x18fe0
0xf914: mov     r12, rax
0xf917: test    rax, rax
0xf91a: je      0xff20
0xf920: mov     rdi, qword ptr [r15 + 0x18]
0xf924: call     0x18d10
0xf929: mov     qword ptr [r15 + 0x18], r12
0xf92d: mov     rdi, r12
0xf930: call     0x18040
0xf935: mov     rbp, qword ptr [r15 + 0x18]
0xf939: mov     rdi, rbp
0xf93c: call     0x49b0
0xf941: lea     rsi, [rsp + 0xf0]
0xf949: mov     rdi, rbp
0xf94c: mov     qword ptr [r15 + 0x78], rax
0xf950: call     0x18f40
0xf955: cmp     qword ptr [rsp + 0x120], 0x7ff
0xf961: jg      0xff08
0xf967: lea     rsi, [rsp + 0x70]
0xf96c: lea     rdi, [rip + 0x11d6b]
0xf973: call     0x4c00
0xf978: xor     eax, eax
0xf97a: cmp     qword ptr [rsp + 0x70], 0x4d44
0xf983: sete    al
0xf986: mov     dword ptr [r15 + 0x10], eax
0xf98a: test    ebx, ebx
0xf98c: je      0xf99b
0xf98e: mov     r13d, dword ptr [r15 + 0x70]
0xf992: test    r13d, r13d
0xf995: jne     0xff78
0xf99b: mov     rbp, qword ptr [r15 + 0x38]
0xf99f: test    rbp, rbp
0xf9a2: je      0x10018
0xf9a8: mov     r12d, dword ptr [r15 + 0x68]

```

```

0xf9ac: test    r12d, r12d
0xf9af: jle     0xf9e6
0xf9b1: xor     ebx, ebx
0xf9b3: nop     dword ptr [rax + rax]
0xf9b8: lea     r12, [rbx + rbx*4]
0xf9bc: shl     r12, 4
0xf9c0: mov     rdi, qword ptr [rbp + r12]
0xf9c5: test    rdi, rdi
0xf9c8: je      0xf9dc
0xf9ca: call    0x18d10
0xf9cf: mov     rbp, qword ptr [r15 + 0x38]
0xf9d3: mov     qword ptr [rbp + r12], 0
0xf9dc: add     rbx, 1
0xf9e0: cmp     dword ptr [r15 + 0x68], ebx
0xf9e4: jg      0xf9b8
0xf9e6: mov     rdi, rbp
0xf9e9: xor     ebp, ebp
0xf9eb: call    0x18d10
0xf9f0: mov     qword ptr [r15 + 0x38], 0
0xf9f8: mov     dword ptr [rsp + 0xc], 0
0xfa00: mov     dword ptr [rsp], 0
0xfa07: mov     edi, 0x50
0xfa0c: call    0x1d670
0xfa11: mov     dword ptr [r15 + 0x70], 0
0xfa19: mov     dword ptr [r15 + 0x30], 0
0xfa21: mov     dword ptr [rsp + 0x14], 0xffffffff
0xfa29: mov     dword ptr [rsp + 0x10], 0
0xfa31: mov     qword ptr [r15 + 0x38], rax
0xfa35: xor     eax, eax
0xfa37: nop     word ptr [rax + rax]
0xfa40: mov     dword ptr [r15 + 0x68], eax
0xfa44: mov     rdi, qword ptr [r15]
0xfa47: call    0x4c10
0xfa4c: test    rax, rax
0xfa4f: je      0xfbf0
0xfa55: movabs  rdx, 0xffff000000
0xfa5f: and     rdx, qword ptr [rax + 0x10]
0xfa63: cmp     rdx, 0x2e000000
0xfa6a: je      0xfbf0
0xfa70: cmp     byte ptr [rax + 0x13], 0x2e
0xfa74: lea     r13, [rax + 0x13]
0xfa78: je      0xfbf0
0xfa7e: mov     eax, dword ptr [r15 + 0x68]
0xfa82: add     eax, 1
0xfa85: mov     rdi, qword ptr [r15 + 0x38]
0xfa89: cdqe
0xfa8b: lea     rsi, [rax + rax*4]
0xfa8f: shl     rsi, 4

```

```

0xfa93: call    0x1d690
0xfa98: mov     ebx, dword ptr [rsp + 0xc]
0xfa9c: mov     qword ptr [r15 + 0x38], rax
0xfaa0: mov     r14, rax
0xfaa3: test    ebx, ebx
0xfaa5: jne     0xfb90
0xfaab: movsxd  rax, dword ptr [r15 + 0x68]
0xfaaf: lea     rdx, [rax + rax*4]
0xfab3: shl     rdx, 4
0xfab7: add     r14, rdx
0xfaba: mov     byte ptr [r14 + 0x49], 0
0xfabf: mov     byte ptr [r14 + 0x4b], 0
0xfac4: mov     rdi, r13
0xfac7: call    0x49b0
0xfacc: lea     edi, [rax + 1]
0xfacf: mov     rbx, rax
0xfad2: movsxd  rdi, edi
0xfad5: call    0x1d670
0xfada: mov     rsi, r13
0xfadd: mov     qword ptr [r14], rax
0xfae0: movsxd  rax, dword ptr [r15 + 0x68]
0xfae4: lea     rax, [rax + rax*4]
0xfae8: shl     rax, 4
0xfaec: add     rax, qword ptr [r15 + 0x38]
0xfaf0: mov     rdi, qword ptr [rax]
0xfaf3: call    0x48b0
0xfaf8: mov     eax, dword ptr [r15 + 0x30]
0xfafc: mov     esi, dword ptr [r15 + 0x68]
0xfb00: mov     rdi, r15
0xfb03: cmp     eax, ebx
0xfb05: cmovl   eax, ebx
0xfb08: mov     dword ptr [r15 + 0x30], eax
0xfb0c: xor     eax, eax
0xfb0e: call    0xe840
0xfb13: mov     eax, dword ptr [r15 + 0x68]
0xfb17: add     eax, 1
0xfb1a: jmp     0xfa40
0xfa40:
0xfb1f: nop
0xfb20: mov     rdi, rbp
0xfb23: call    0x1d7a0
0xfb28: mov     qword ptr [r15 + 0x18], rax
0xfb2c: mov     r12, rax
0xfb2f: jmp     0xf92d
0xf92d:
0xfb34: nop     dword ptr [rax]
0xfb38: cmp     byte ptr [r13 + 1], 0x2e
0xfb3d: jne     0xfa7e

```



```

0xfb43: cmp    byte ptr [r13 + 2], 0
0xfb48: jne    0xfa7e
0xfb4e: mov    rdx, qword ptr [r15 + 0x18]
0xfb52: mov    eax, dword ptr [r15 + 0x68]
0xfb56: cmp    byte ptr [rdx + 1], 0
0xfb5a: je     0xfb80
0xfb5c: mov    dword ptr [rsp + 0x14], eax
0xfb60: mov    dword ptr [rsp + 0x10], 1
0xfb68: jmp    0xfa82
0xfa82:
0xfb6d: nop    dword ptr [rax]
0xfb70: mov    eax, dword ptr [r15 + 0x68]
0xfb74: jmp    0xfa40
0xfa40:
0xfb79: nop    dword ptr [rax]
0xfb80: mov    dword ptr [rsp + 0x10], 1
0xfb88: jmp    0xfa40
0xfa40:
0xfb8d: nop    dword ptr [rax]
0xfb90: mov    r11d, dword ptr [rsp]
0xfb94: test   r11d, r11d
0xfb97: jle    0x1002c
0xfb9d: mov    rbx, rbp
0xfba0: xor    r12d, r12d
0xfba3: jmp    0xfbba
0xfbba:
0xfba5: nop    dword ptr [rax]
0xfba8: add    r12d, 1
0xfbac: add    rbx, 0x50
0xfbb0: cmp    dword ptr [rsp], r12d
0xfbb4: je     0xfaab
0xfbb8: mov    rsi, qword ptr [rbx]
0xfbbd: mov    rdi, r13
0xfbc0: call   0x4b30
0xfbc5: test   eax, eax
0xfbc7: jne    0xfba8
0xfbc9: movsxd rax, dword ptr [r15 + 0x68]
0xfbcd: lea    rdx, [rax + rax*4]
0xfbd1: movsx  eax, byte ptr [rbx + 0x49]
0xfbd5: shl    rdx, 4
0xfbd9: add    r14, rdx
0xfbd9: mov    byte ptr [r14 + 0x49], al
0xfbe0: add    dword ptr [r15 + 0x70], eax
0xfbe4: jmp    0xfabf
0xfabf:
0xfbe9: nop    dword ptr [rax]
0xfbf0: mov    r9d, dword ptr [rsp + 0x10]
0xfbf5: test   r9d, r9d

```

```

0xfbf8: je      0xfe48
0xfbfe: movsxd  rax, dword ptr [rsp + 0x14]
0xfc03: cmp     eax, -1
0xfc06: je      0xfe00
0xfc0c: mov     rdi, qword ptr [r15 + 0x38]
0xfc10: lea     rax, [rax + rax*4]
0xfc14: shl     rax, 4
0xfc18: add     rax, rdi
0xfc1b: mov     edx, dword ptr [rax + 0x10]
0xfc1e: and     edx, 0xf000
0xfc24: cmp     edx, 0x4000
0xfc2a: je      0xfc33
0xfc2c: mov     dword ptr [rax + 0x10], 0x4000
0xfc33: mov     edx, dword ptr [rsp + 0xc]
0xfc37: test    edx, edx
0xfc39: je      0xffb0
0xfc3f: mov     eax, dword ptr [rsp]
0xfc42: test    eax, eax
0xfc44: jle     0xfc7d
0xfc46: sub     eax, 1
0xfc49: mov     rbx, rbp
0xfc4c: lea     rax, [rax + rax*4]
0xfc50: shl     rax, 4
0xfc54: lea     r12, [rbp + rax + 0x50]
0xfc59: nop     dword ptr [rax]
0xfc60: mov     rdi, qword ptr [rbx]
0xfc63: test    rdi, rdi
0xfc66: je      0xfc74
0xfc68: call    0x18d10
0xfc6d: mov     qword ptr [rbx], 0
0xfc74: add     rbx, 0x50
0xfc78: cmp     r12, rbx
0xfc7b: jne     0xfc60
0xfc7d: mov     rdi, rbp
0xfc80: call    0x18d10
0xfc85: mov     esi, dword ptr [r15 + 0x68]
0xfc89: mov     rdi, qword ptr [r15 + 0x38]
0xfc8d: mov     eax, dword ptr [r15 + 0x84]
0xfc94: mov     dword ptr [r15 + 0x74], 0
0xfc9c: mov     dword ptr [rip + 0x1b986], eax
0xfca2: movsxd  rax, dword ptr [rsp + 0x14]
0xfca7: cmp     eax, -1
0xfcaa: je      0xfe29
0xfcb0: lea     rax, [rax + rax*4]
0xfcb4: shl     rax, 4
0xfcb8: add     rax, rdi
0xfcbb: movdqu  xmm5, xmmword ptr [rax]
0xfcbf: movdqu  xmm4, xmmword ptr [rdi]

```

```

0xfcc3: sub     esi, 1
0xfcc6: add     rdi, 0x50
0xfcca: movdqu  xmm3, xmmword ptr [rdi - 0x40]
0xfccf: movdqu  xmm2, xmmword ptr [rdi - 0x30]
0xfcd4: movsxd  rsi, esi
0xfcd7: mov     edx, 0x50
0xfcdc: movups  xmmword ptr [rdi - 0x50], xmm5
0xfce0: movdqu  xmm6, xmmword ptr [rax + 0x10]
0xfce5: movdqu  xmm1, xmmword ptr [rdi - 0x20]
0xfcea: lea     rcx, [rip - 0x1671]
0xfcf1: movdqu  xmm0, xmmword ptr [rdi - 0x10]
0xfcf6: movaps  xmmword ptr [rsp + 0x20], xmm4
0xfcfb: movups  xmmword ptr [rdi - 0x40], xmm6
0xfcff: movdqu  xmm7, xmmword ptr [rax + 0x20]
0xfd04: movaps  xmmword ptr [rsp + 0x30], xmm3
0xfd09: movups  xmmword ptr [rdi - 0x30], xmm7
0xfd0d: movdqu  xmm5, xmmword ptr [rax + 0x30]
0xfd12: movaps  xmmword ptr [rsp + 0x40], xmm2
0xfd17: movups  xmmword ptr [rdi - 0x20], xmm5
0xfd1b: movdqu  xmm6, xmmword ptr [rax + 0x40]
0xfd20: movaps  xmmword ptr [rsp + 0x50], xmm1
0xfd25: movups  xmmword ptr [rdi - 0x10], xmm6
0xfd29: movups  xmmword ptr [rax], xmm4
0xfd2c: movups  xmmword ptr [rax + 0x10], xmm3
0xfd30: movups  xmmword ptr [rax + 0x20], xmm2
0xfd34: movups  xmmword ptr [rax + 0x30], xmm1
0xfd38: movups  xmmword ptr [rax + 0x40], xmm0
0xfd3c: movaps  xmmword ptr [rsp + 0x60], xmm0
0xfd41: call    0x48e0
0xfd46: mov     edx, dword ptr [r15 + 0x68]
0xfd4a: mov     eax, dword ptr [r15 + 0x28]
0xfd4e: mov     rdi, r15
0xfd51: mov     r13d, 1
0xfd57: cmp     edx, eax
0xfd59: lea     ecx, [rdx - 1]
0xfd5c: cmovle  eax, ecx
0xfd5f: mov     dword ptr [r15 + 0x28], eax
0xfd63: xor     eax, eax
0xfd65: call    0xf110
0xfd6a: mov     rdi, qword ptr [rsp + 0x18]
0xfd6f: mov     dword ptr [r15 + 0x2c], eax
0xfd73: call    0x18d10
0xfd78: mov     rax, qword ptr [rsp + 0x188]
0xfd80: sub     rax, qword ptr fs:[0x28]
0xfd89: jne     0x100b2
0xfd8f: add     rsp, 0x198
0xfd96: mov     eax, r13d
0xfd99: pop     rbx

```

```

0xfd9a: pop    rbp
0xfd9b: pop    r12
0xfd9d: pop    r13
0xfd9f: pop    r14
0xfda1: pop    r15
0xfda3: ret
0xfda4: nop    dword ptr [rax]
0xfda8: cmp    byte ptr [rbp], 0x2e
0xfdac: jne    0xfdf0
0xfdae: cmp    byte ptr [rbp + 1], 0x2e
0xfdb2: jne    0xfdf0
0xfdb4: movzx  r13d, byte ptr [rbp + 2]
0xfdb9: test   r13d, r13d
0xfdbc: jne    0xfdf0
0xfdbe: xor    eax, eax
0xfdc0: call   0x18fe0
0xfdc5: mov    rbp, rax
0xfdc8: test   rax, rax
0xfdcb: je     0x10060
0xfdd1: mov    rdi, rbp
0xfdd4: call   0x4950
0xfdd9: mov    r12, rax
0xfddc: test   rax, rax
0xfddf: jne    0xf8c4
0xfde5: mov    rdi, rbp
0xfde8: call   0x18d10
0xfded: jmp    0xfd78
0xfd78:
0xfdef: nop
0xfdf0: xor    r13d, r13d
0xfdf3: jmp    0xfd78
0xfd78:
0xfdf8: nop    dword ptr [rax + rax]
0xfe00: mov    eax, dword ptr [rsp + 0xc]
0xfe04: test   eax, eax
0xfe06: jne    0xfc3f
0xfe0c: mov    esi, dword ptr [r15 + 0x68]
0xfe10: mov    rdi, qword ptr [r15 + 0x38]
0xfe14: mov    dword ptr [r15 + 0x74], 0
0xfe1c: mov    eax, dword ptr [r15 + 0x84]
0xfe23: mov    dword ptr [rip + 0x1b7ff], eax
0xfe29: movsxd rsi, esi
0xfe2c: lea    rcx, [rip - 0x17b3]
0xfe33: mov    edx, 0x50
0xfe38: call   0x48e0
0xfe3d: jmp    0xfd46
0xfd46:
0xfe42: nop    word ptr [rax + rax]

```

```

0xfe48: mov     eax, dword ptr [r15 + 0x68]
0xfe4c: mov     rdi, qword ptr [r15 + 0x38]
0xfe50: add     eax, 1
0xfe53: cdqe
0xfe55: lea     rsi, [rax + rax*4]
0xfe59: shl     rsi, 4
0xfe5d: call    0x1d690
0xfe62: movsxd  rdx, dword ptr [r15 + 0x68]
0xfe66: mov     edi, 3
0xfe6b: mov     qword ptr [r15 + 0x38], rax
0xfe6f: lea     rdx, [rdx + rdx*4]
0xfe73: shl     rdx, 4
0xfe77: lea     rbx, [rax + rdx]
0xfe7b: mov     byte ptr [rbx + 0x49], 0
0xfe7f: mov     byte ptr [rbx + 0x4b], 1
0xfe83: call    0x1d670
0xfe88: mov     ecx, 0x2e2e
0xfe8d: mov     edx, 2
0xfe92: mov     rdi, r15
0xfe95: mov     qword ptr [rbx], rax
0xfe98: movsxd  rax, dword ptr [r15 + 0x68]
0xfe9c: lea     rax, [rax + rax*4]
0xfea0: shl     rax, 4
0xfea4: add     rax, qword ptr [r15 + 0x38]
0xfea8: mov     rax, qword ptr [rax]
0xfeab: mov     word ptr [rax], cx
0xfeae: mov     byte ptr [rax + 2], 0
0xfeb2: mov     eax, dword ptr [r15 + 0x30]
0xfeb6: mov     esi, dword ptr [r15 + 0x68]
0xfeba: cmp     eax, edx
0xfebc: cmovl   eax, edx
0xfebf: mov     dword ptr [r15 + 0x30], eax
0xfec3: xor     eax, eax
0xfec5: call    0xe840
0xfeca: movsxd  rax, dword ptr [r15 + 0x68]
0xfece: mov     rdi, qword ptr [r15 + 0x38]
0xfed2: mov     r8d, dword ptr [rsp + 0xc]
0xfed7: mov     rsi, rax
0xfeda: lea     rax, [rax + rax*4]
0xfede: shl     rax, 4
0xfef2: add     esi, 1
0xfef5: add     rax, rdi
0xfef8: mov     dword ptr [rax + 0x10], 0x4000
0xfefb: mov     byte ptr [rax + 0x48], 0
0xfef3: mov     dword ptr [r15 + 0x68], esi
0xfef7: test    r8d, r8d
0xfefa: je      0xfc8d
0xff00: jmp     0xfc3f

```

```

0xfc3f
0xff05: nop      dword ptr [rax]
0xff08: mov      rdi, qword ptr [rip + 0x1a571]
0xff0f: call     0xdb60
0xff14: xor      eax, eax
0xff16: call     0xa820
0xff1b: jmp      0xf967
0xf967
0xff20: mov      r13, qword ptr [r15 + 0x18]
0xff24: mov      rdi, r13
0xff27: call     0x49b0
0xff2c: mov      qword ptr [r15 + 0x78], rax
0xff30: cmp      byte ptr [rbp], 0x2e
0xff34: mov      r12, rax
0xff37: jne      0xffd0
0xff3d: cmp      byte ptr [rbp + 1], 0x2e
0xff41: jne      0xffd0
0xff47: cmp      byte ptr [rbp + 2], 0
0xff4b: jne      0xffd0
0xff51: mov      esi, 0x2f
0xff56: mov      rdi, r13
0xff59: call     0x4a20
0xff5e: test     rax, rax
0xff61: je       0x1008b
0xff67: mov      byte ptr [rax], 0
0xff6a: mov      r12, qword ptr [r15 + 0x18]
0xff6e: jmp      0xf92d
0xf92d
0xff73: nop      dword ptr [rax + rax]
0xff78: mov      rsi, qword ptr [r15 + 0x18]
0xff7c: mov      rdi, qword ptr [rsp + 0x18]
0xff81: call     0x4b30
0xff86: test     eax, eax
0xff88: jne      0xf99b
0xff8e: mov      eax, dword ptr [r15 + 0x68]
0xff92: mov      rbp, qword ptr [r15 + 0x38]
0xff96: mov      dword ptr [rsp + 0xc], 1
0xff9e: mov      qword ptr [r15 + 0x38], 0
0xffa6: mov      dword ptr [rsp], eax
0xffa9: jmp      0xfa07
0xfa07
0xffae: nop
0xffb0: mov      edx, dword ptr [r15 + 0x84]
0xffb7: mov      esi, dword ptr [r15 + 0x68]
0xffbb: mov      dword ptr [r15 + 0x74], 0
0xffc3: mov      dword ptr [rip + 0x1b65f], edx
0xffc9: jmp      0xfcbb
0xfcbb

```

```

0xffce: nop
0xffd0: mov     rdi, rbp
0xffd3: mov     r14d, 0x2f
0xffd9: call    0x49b0
0xffde: mov     rdi, r13
0xffe1: lea     rsi, [r12 + rax + 2]
0xffe6: call    0x1d690
0xffeb: mov     qword ptr [r15 + 0x18], rax
0xffef: mov     rdi, rax
0xfff2: mov     r12, rax
0xfff5: call    0x49b0
0xfffa: mov     rsi, rbp
0xfffd: mov     word ptr [r12 + rax], r14w
0x10002: mov     rdi, qword ptr [r15 + 0x18]
0x10006: call    0x4df0
0x1000b: mov     r12, qword ptr [r15 + 0x18]
0x1000f: jmp     0xf92d
0xf92d
0x10014: nop     dword ptr [rax]
0x10018: mov     dword ptr [rsp + 0xc], 0
0x10020: mov     dword ptr [rsp], 0
0x10027: jmp     0xfa07
0xfa07
0x1002c: movsxd  rax, dword ptr [r15 + 0x68]
0x10030: mov     r10d, dword ptr [rsp]
0x10034: lea     rdx, [rax + rax*4]
0x10038: shl     rdx, 4
0x1003c: add     r14, rdx
0x1003f: test    r10d, r10d
0x10042: jne     0xfabf
0x10048: jmp     0xfaba
0xfaba
0x1004d: nop     dword ptr [rax]
0x10050: mov     rdi, r12
0x10053: xor     r13d, r13d
0x10056: call    0x4ac0
0x1005b: jmp     0xfd78
0xfd78
0x10060: mov     rdi, qword ptr [r15 + 0x18]
0x10064: call    0x1d7a0
0x10069: mov     esi, 0x2f
0x1006e: mov     rdi, rax
0x10071: mov     rbp, rax
0x10074: call    0x4a20
0x10079: mov     r12, rax
0x1007c: test    rax, rax
0x1007f: je      0x100a4
0x10081: mov     byte ptr [r12], 0

```

```

0x10086:      jmp      0xfdd1
0xfdd1
0x1008b:      lea      rdi, [rip + 0x11151]
0x10092:      mov      qword ptr [rsp], rax
0x10096:      call     0x8b00
0x1009b:      mov      rax, qword ptr [rsp]
0x1009f:      jmp      0xff67
0xff67
0x100a4:      lea      rdi, [rip + 0x1112a]
0x100ab:      call     0x8b00
0x100b0:      jmp      0x10081
0x10081
0x100b2:      call     0x49d0
0x100b7:      nop      word ptr [rax + rax]
0x100c0:      endbr64
0x100c4:      mov      eax, dword ptr [rdi + 0x70]
0x100c7:      mov      dword ptr [rdi + 0x50], 0xffffffff
0x100ce:      mov      dword ptr [rdi + 0x54], eax
0x100d1:      ret
0x100d2:      nop      word ptr cs:[rax + rax]
0x100dd:      nop      dword ptr [rax]
0x100e0:      endbr64
0x100e4:      mov      eax, dword ptr [rdi + 0x54]
0x100e7:      movsxd   rdx, dword ptr [rdi + 0x50]
0x100eb:      test     eax, eax
0x100ed:      je       0x10128
0x100ef:      mov      ecx, dword ptr [rdi + 0x68]
0x100f2:      lea      eax, [rdx + 1]
0x100f5:      cmp      ecx, eax
0x100f7:      jle      0x10154
0x100f9:      mov      rsi, qword ptr [rdi + 0x38]
0x100fd:      lea      rdx, [rdx + rdx*4]
0x10101:      shl      rdx, 4
0x10105:      lea      rdx, [rsi + rdx + 0x99]
0x1010d:      jmp      0x1011b
0x1011b
0x1010f:      nop
0x10110:      add      eax, 1
0x10113:      add      rdx, 0x50
0x10117:      cmp      ecx, eax
0x10119:      je       0x10154
0x1011b:      cmp      byte ptr [rdx], 0
0x1011e:      je       0x10110
0x10120:      mov      dword ptr [rdi + 0x50], eax
0x10123:      ret
0x10124:      nop      dword ptr [rax]
0x10128:      test     edx, edx
0x1012a:      je       0x10154

```



```

0x1012c:    mov     rdx, qword ptr [rdi + 0x18]
0x10130:    mov     eax, dword ptr [rdi + 0x28]
0x10133:    mov     dword ptr [rdi + 0x50], 0
0x1013a:    cmp     byte ptr [rdx], 0x2f
0x1013d:    jne     0x10150
0x1013f:    cmp     byte ptr [rdx + 1], 0
0x10143:    jne     0x10150
0x10145:    ret
0x10146:    nop     word ptr cs:[rax + rax]
0x10150:    test    eax, eax
0x10152:    jne     0x10145
0x10154:    mov     eax, 0xffffffff
0x10159:    ret
0x1015a:    nop     word ptr [rax + rax]
0x10160:    endbr64
0x10164:    push    r13
0x10166:    push    r12
0x10168:    push    rbp
0x10169:    push    rbx
0x1016a:    sub     rsp, 0x18
0x1016e:    mov     rax, qword ptr fs:[0x28]
0x10177:    mov     qword ptr [rsp + 8], rax
0x1017c:    xor     eax, eax
0x1017e:    cmp     dword ptr [rdi + 0x64], 0x13
0x10182:    jle     0x1018d
0x10184:    cmp     byte ptr [rdi + 0x6d], 0
0x10188:    mov     rbx, rdi
0x1018b:    jne     0x101b0
0x1018d:    mov     rax, qword ptr [rsp + 8]
0x10192:    sub     rax, qword ptr fs:[0x28]
0x1019b:    jne     0x10236
0x101a1:    add     rsp, 0x18
0x101a5:    pop     rbx
0x101a6:    pop     rbp
0x101a7:    pop     r12
0x101a9:    pop     r13
0x101ab:    ret
0x101ac:    nop     dword ptr [rax]
0x101b0:    lea     r13, [rsp + 7]
0x101b5:    mov     rdi, r13
0x101b8:    call    0xa6a0
0x101bd:    movsxd  rdx, dword ptr [rbx + 0x64]
0x101c1:    mov     rdi, qword ptr [rbx + 0x20]
0x101c5:    mov     esi, 0x20
0x101ca:    mov     rbp, rdx
0x101cd:    lea     r12d, [rdx - 0x13]
0x101d1:    call    0x4a50
0x101d6:    mov     rsi, qword ptr [rbx + 0x20]

```

```

0x101da:    mov     rdi, qword ptr [rbx + 0x18]
0x101de:    mov     edx, r12d
0x101e1:    call    0x18200
0x101e6:    mov     rdi, qword ptr [rbx + 0x20]
0x101ea:    movsxd   rsi, r12d
0x101ed:    call    0x184d0
0x101f2:    mov     edx, dword ptr [rip + 0x1a128]
0x101f8:    mov     esi, dword ptr [rip + 0x1a152]
0x101fe:    mov     edi, dword ptr [rip + 0x1a150]
0x10204:    call    0xa570
0x10209:    mov     rdi, qword ptr [rbx + 8]
0x1020d:    xor     esi, esi
0x1020f:    mov     edx, 2
0x10214:    call    0xc680
0x10219:    mov     rdi, qword ptr [rbx + 8]
0x1021d:    mov     rsi, qword ptr [rbx + 0x20]
0x10221:    lea     edx, [rbp - 0x12]
0x10224:    call    0xc600
0x10229:    mov     rdi, r13
0x1022c:    call    0xa6b0
0x10231:    jmp     0x1018d
0x1018d:
0x10236:    call    0x49d0
0x1023b:    nop     dword ptr [rax + rax]
0x10240:    endbr64
0x10244:    push    r15
0x10246:    push    r14
0x10248:    push    r13
0x1024a:    push    r12
0x1024c:    push    rbp
0x1024d:    push    rbx
0x1024e:    sub     rsp, 0x2f8
0x10255:    mov     rax, qword ptr fs:[0x28]
0x1025e:    mov     qword ptr [rsp + 0x2e8], rax
0x10266:    xor     eax, eax
0x10268:    cmp     byte ptr [rdi + 0x6d], 0
0x1026c:    jne     0x102a0
0x1026e:    mov     rax, qword ptr [rsp + 0x2e8]
0x10276:    sub     rax, qword ptr fs:[0x28]
0x1027f:    jne     0x103f2
0x10285:    add     rsp, 0x2f8
0x1028c:    pop     rbx
0x1028d:    pop     rbp
0x1028e:    pop     r12
0x10290:    pop     r13
0x10292:    pop     r14
0x10294:    pop     r15
0x10296:    ret

```

```

0x10297:      nop      word ptr [rax + rax]
0x102a0:      mov      eax, dword ptr [rdi + 0x64]
0x102a3:      lea      r14, [rsp + 0xf]
0x102a8:      mov      rbx, rdi
0x102ab:      mov      rdi, r14
0x102ae:      lea      r12d, [rax - 4]
0x102b2:      call     0xa6a0
0x102b7:      mov      qword ptr [rsp + 0x18], -1
0x102c0:      cmp      r12d, 5
0x102c4:      jle      0x10350
0x102ca:      mov      rdi, qword ptr [rbx + 0x18]
0x102ce:      xor      esi, esi
0x102d0:      lea      rdx, [rsp + 0x10]
0x102d5:      call     0x1c490
0x102da:      test     eax, eax
0x102dc:      js       0x10350
0x102de:      cmp      qword ptr [rsp + 0x18], -1
0x102e4:      je       0x10350
0x102e6:      call     0x4a60
0x102eb:      test     eax, eax
0x102ed:      je       0x103e8
0x102f3:      mov      rdi, qword ptr [rsp + 0x28]
0x102f8:      xor      eax, eax
0x102fa:      test     rdi, rdi
0x102fd:      mov      edx, 0xc
0x10302:      mov      r8d, 1
0x10308:      cmovs    rdi, rax
0x1030c:      imul     rdi, qword ptr [rsp + 0x10]
0x10312:      mov      eax, 0x9c
0x10317:      mov      ecx, 1
0x1031c:      lea      rsi, [rsp + 0x50]
0x10321:      cmp      rdi, 0x2710
0x10328:      cmovge   edx, eax
0x1032b:      call     0x1c520
0x10330:      mov      edi, dword ptr [rip + 0x1a026]
0x10336:      mov      rbp, rax
0x10339:      call     0xa530
0x1033e:      mov      edi, dword ptr [rip + 0x1a014]
0x10344:      call     0xa4f0
0x10349:      jmp      0x10388
0x10388:
0x1034b:      nop      dword ptr [rax + rax]
0x10350:      lea      rbp, [rsp + 0x50]
0x10355:      mov      ecx, 0x51
0x1035a:      movabs   rax, 0x2020202020202020
0x10364:      mov      rdi, rbp
0x10367:      rep stosq      qword ptr [rdi], rax
0x1036a:      mov      eax, 0x2020

```

```

0x1036f:      mov     word ptr [rdi], ax
0x10372:      mov     byte ptr [rdi + 2], 0x20
0x10376:      xor     edi, edi
0x10378:      call    0xa530
0x1037d:      mov     edi, dword ptr [rip + 0x19f9d]
0x10383:      call    0xa4f0
0x10388:      mov     edi, dword ptr [rip + 0x19f92]
0x1038e:      movsxd   r15, r12d
0x10391:      call    0xa510
0x10396:      mov     rdi, rbp
0x10399:      mov     r13d, dword ptr [rbx + 0x64]
0x1039d:      call    0x49b0
0x103a2:      mov     rdi, qword ptr [rbx + 8]
0x103a6:      cmp     rax, r15
0x103a9:      lea     edx, [r13 - 2]
0x103ad:      cmova    eax, r12d
0x103b1:      xor     esi, esi
0x103b3:      sub     edx, eax
0x103b5:      call    0xc680
0x103ba:      mov     rdi, rbp
0x103bd:      call    0x49b0
0x103c2:      mov     rdi, qword ptr [rbx + 8]
0x103c6:      mov     rsi, rbp
0x103c9:      cmp     r15, rax
0x103cc:      cmovae   r12d, eax
0x103d0:      mov     edx, r12d
0x103d3:      call    0xc600
0x103d8:      mov     rdi, r14
0x103db:      call    0xa6b0
0x103e0:      jmp     0x1026e
0x1026e
0x103e5:      nop     dword ptr [rax]
0x103e8:      mov     rdi, qword ptr [rsp + 0x20]
0x103ed:      jmp     0x102f8
0x102f8
0x103f2:      call    0x49d0
0x103f7:      nop     word ptr [rax + rax]
0x10400:      endbr64
0x10404:      cmp     byte ptr [rdi + 0x6d], 0
0x10408:      jne     0x10410
0x1040a:      ret
0x1040b:      nop     dword ptr [rax + rax]
0x10410:      xor     eax, eax
0x10412:      jmp     0xec50
0xec50
0x10417:      nop     word ptr [rax + rax]
0x10420:      endbr64
0x10424:      push    r12

```

```

0x10426:    push    rbp
0x10427:    push    rbx
0x10428:    sub     rsp, 0x2a0
0x1042f:    mov     rax, qword ptr fs:[0x28]
0x10438:    mov     qword ptr [rsp + 0x298], rax
0x10440:    xor     eax, eax
0x10442:    cmp     edx, 5
0x10445:    ja      0x10660
0x1044b:    mov     ebp, ecx
0x1044d:    mov     edx, edx
0x1044f:    lea     rcx, [rip + 0x1146e]
0x10456:    mov     rbx, rdi
0x10459:    movsxd  rax, dword ptr [rcx + rdx*4]
0x1045d:    add     rax, rcx
0x10460:    jmp     rax
rax
->->->                                indirect JMP
0x853f: xor     eax, eax
0x8541: call    0xa5a0
0x8546: xor     eax, eax
0x8548: call    0xde0
0x854d: xor     eax, eax
0x854f: call    0xd740
0x8554: jmp     0x847d
0x847d
0x8559: nop     dword ptr [rax]
0x8560: mov     rdi, r13
0x8563: call    0xdb10
0x8568: test    rbp, rbp
0x856b: je      0x857b
0x856d: mov     rsi, qword ptr [r13]
0x8571: mov     rdi, rbp
0x8574: xor     eax, eax
0x8576: call    0x7e00
0x857b: mov     rax, qword ptr [r13]
0x857f: mov     rdx, qword ptr [rsp + 0x28]
0x8584: sub     rdx, qword ptr fs:[0x28]
0x858d: jne     0x86b3
0x8593: add     rsp, 0x38
0x8597: pop     rbx
0x8598: pop     rbp
0x8599: pop     r12
0x859b: pop     r13
0x859d: pop     r14
0x859f: pop     r15
0x85a1: ret
0x85a2: nop     word ptr [rax + rax]
0x85a8: test    rbp, rbp

```

```

0x85ab: je      0x8519
0x85b1: lea      eax, [r14 + 1]
0x85b5: mov      rsi, qword ptr [rsp]
0x85b9: mov      rdi, rbp
0x85bc: mov      edx, eax
0x85be: mov      r15d, eax
0x85c1: call     0x19200
0x85c6: test     rax, rax
0x85c9: je      0x86a7
0x85cf: mov      r8, qword ptr [rsp + 0x20]
0x85d4: test     r8, r8
0x85d7: je      0x86a7
0x85dd: xor      edi, edi
0x85df: mov      qword ptr [rsp + 8], r8
0x85e4: mov      r14d, r15d
0x85e7: call     0xd1a0
0x85ec: mov      rdi, qword ptr [rsp + 8]
0x85f1: call     0xd610
0x85f6: xor      eax, eax
0x85f8: call     0xd870
0x85fd: xor      eax, eax
0x85ff: call     0xd740
0x8604: jmp     0x8519
0x8519
0x8609: nop      dword ptr [rax]
0x8610: test     rbp, rbp
0x8613: je      0x8519
0x8619: xor      edi, edi
0x861b: test     r14d, r14d
0x861e: je      0x8680
0x8620: call     0xd1a0
0x8625: sub      r14d, 1
0x8629: je      0x864b
0x862b: mov      rsi, qword ptr [rsp]
0x862f: mov      rdi, rbp
0x8632: mov      edx, r14d
0x8635: call     0x19200
0x863a: mov      rdi, rax
0x863d: test     rax, rax
0x8640: cmovne   rdi, qword ptr [rsp + 0x20]
0x8646: call     0xd610
0x864b: xor      eax, eax
0x864d: call     0xd870
0x8652: xor      eax, eax
0x8654: call     0xd740
0x8659: jmp     0x8519
0x8519
0x865e: nop

```

```

0x8660: xor     edi, edi
0x8662: xor     eax, eax
0x8664: call    0x7bd0
0x8669: jmp     0x8519
0x8519
0x866e: nop
0x8670: call    0xa5a0
0x8675: jmp     0x8505
0x8505
0x867a: nop     word ptr [rax + rax]
0x8680: call    0xd1a0
0x8685: xor     eax, eax
0x8687: call    0xd870
0x868c: xor     eax, eax
0x868e: call    0xd740
0x8693: jmp     0x8519
0x8519
0x8698: nop     dword ptr [rax + rax]
0x86a0: xor     eax, eax
0x86a2: jmp     0x857f
0x857f
0x86a7: xor     eax, eax
0x86a9: call    0xa5a0
0x86ae: jmp     0x8519
0x8519
0x86b3: call    0x49d0
0x86b8: nop     dword ptr [rax + rax]
0x86c0: endbr64
0x86c4: push    r15
0x86c6: push    r14
0x86c8: push    r13
0x86ca: push    r12
0x86cc: push    rbp
0x86cd: push    rbx
0x86ce: sub     rsp, 0x18
0x86d2: test    edx, edx
0x86d4: je      0x8890
0x86da: mov     ebx, edx
0x86dc: cmp     edx, 3
0x86df: je      0x88f8
0x86e5: mov     rbp, rcx
0x86e8: test    rcx, rcx
0x86eb: je      0x8864
0x86f1: mov     dword ptr [rcx], 0
0x86f7: xor     eax, eax
0x86f9: mov     r13, rsi
0x86fc: lea     r15, [rip + 0x2091d]
0x8703: movabs  r14, 0xea0ea0ea0ea0ea1

```

```

0x870d: call    0xd870
0x8712: xor     eax, eax
0x8714: call    0xd740
0x8719: xor     eax, eax
0x871b: call    0xa820
0x8720: xor     edi, edi
0x8722: call    0xb180
0x8727: mov     r12, rax
0x872a: test    rax, rax
0x872d: je      0x8928
0x8733: mov     rax, qword ptr [rax]
0x8736: movzx   edi, byte ptr [rax]
0x8739: mov     rax, qword ptr [r12 + 0x10]
0x873e: test    rax, rax
0x8741: je      0x874d
0x8743: cmp     byte ptr [rax + 0x1b], 0
0x8747: jne     0x8808
0x874d: cmp     edi, -7
0x8750: jge     0x87b0
0x8752: cmp     edi, -0x3d
0x8755: je      0x8840
0x875b: cmp     edi, -0x32
0x875e: jne     0x88bb
0x8764: xor     eax, eax
0x8766: call    0xca40
0x876b: test    eax, eax
0x876d: jne     0x8880
0x8773: mov     dword ptr [rbp], 1
0x877a: xor     eax, eax
0x877c: call    0xdea0
0x8781: xor     eax, eax
0x8783: call    0xd870
0x8788: xor     eax, eax
0x878a: call    0xd740
0x878f: xor     eax, eax
0x8791: call    0xa820
0x8796: mov     rdi, r13
0x8799: call    0xdb10
0x879e: mov     r12, qword ptr [r13]
0x87a2: jmp     0x8867
0x8867:
0x87a7: nop     word ptr [rax + rax]
0x87b0: cmp     edi, 7
0x87b3: je      0x8840
0x87b9: mov     dword ptr [rsp + 8], edi
0x87bd: call    0x4f00
0x87c2: mov     r8, rax
0x87c5: movsxd  rax, dword ptr [rsp + 8]

```



```

0x87ca: mov     rdx, qword ptr [r8]
0x87cd: mov     rdi, rax
0x87d0: test    byte ptr [rdx + rax*2 + 1], 0x40
0x87d5: jne     0x8948
0x87db: xor     eax, eax
0x87dd: xor     r12d, r12d
0x87e0: call    0xdea0
0x87e5: xor     eax, eax
0x87e7: call    0xd870
0x87ec: xor     eax, eax
0x87ee: call    0xd740
0x87f3: xor     eax, eax
0x87f5: call    0xa820
0x87fa: mov     rdi, r13
0x87fd: call    0xdb10
0x8802: jmp     0x8867
0x8867:
0x8804: nop     dword ptr [rax]
0x8808: mov     rcx, qword ptr [rax]
0x880b: sub     rcx, r15
0x880e: mov     rax, rcx
0x8811: sar     rcx, 0x3f
0x8815: imul    r14
0x8818: sar     rdx, 1
0x881b: sub     rdx, rcx
0x881e: mov     edi, edx
0x8820: not     edi
0x8822: cmp     edx, 7
0x8825: jne     0x874d
0x882b: xor     edi, edi
0x882d: xor     eax, eax
0x882f: call    0x7bd0
0x8834: jmp     0x8720
0x8720:
0x8839: nop     dword ptr [rax]
0x8840: xor     eax, eax
0x8842: call    0xdea0
0x8847: xor     eax, eax
0x8849: call    0xd870
0x884e: xor     eax, eax
0x8850: call    0xd740
0x8855: xor     eax, eax
0x8857: call    0xa820
0x885c: mov     rdi, r13
0x885f: call    0xdb10
0x8864: xor     r12d, r12d
0x8867: add     rsp, 0x18
0x886b: mov     rax, r12

```

```

0x886e: pop    rbx
0x886f: pop    rbp
0x8870: pop    r12
0x8872: pop    r13
0x8874: pop    r14
0x8876: pop    r15
0x8878: ret
0x8879: nop    dword ptr [rax]
0x8880: xor     eax, eax
0x8882: call   0xa5a0
0x8887: jmp    0x877a
0x877a:
0x888c: nop    dword ptr [rax]
0x8890: xor     eax, eax
0x8892: mov     qword ptr [rsp + 8], rdi
0x8897: call   0xc8c0
0x889c: mov     qword ptr [rip + 0x2205d], rax
0x88a3: xor     eax, eax
0x88a5: call   0xcff0
0x88aa: mov     rdi, qword ptr [rsp + 8]
0x88af: test    rdi, rdi
0x88b2: je      0x8864
0x88b4: call   0xd540
0x88b9: jmp     0x8864
0x8864:
0x88bb: cmp     edi, -0x44
0x88be: jne     0x88df
0x88c0: cmp     ebx, 1
0x88c3: jne     0x88df
0x88c5: xor     eax, eax
0x88c7: call   0xca40
0x88cc: cmp     eax, 1
0x88cf: mov     eax, 2
0x88d4: sbb     eax, -1
0x88d7: mov     dword ptr [rbp], eax
0x88da: jmp     0x877a
0x877a:
0x88df: cmp     edi, -0x45
0x88e2: jne     0x87b9
0x88e8: cmp     ebx, 2
0x88eb: je      0x88c5
0x88ed: jmp     0x87b9
0x87b9:
0x88f2: nop    word ptr [rax + rax]
0x88f8: mov     rdi, qword ptr [rip + 0x22001]
0x88ff: xor     r12d, r12d
0x8902: call   0xc970
0x8907: xor     eax, eax

```

```

0x8909: call    0xd870
0x890e: xor      eax, eax
0x8910: call    0xd740
0x8915: xor      eax, eax
0x8917: call    0xa820
0x891c: jmp     0x8867
0x8867
0x8921: nop      dword ptr [rax]
0x8928: xor      eax, eax
0x892a: call    0xdea0
0x892f: xor      eax, eax
0x8931: call    0xd740
0x8936: xor      eax, eax
0x8938: call    0xa820
0x893d: jmp     0x8867
0x8867
0x8942: nop      word ptr [rax + rax]
0x8948: call    0xcd30
0x894d: mov     dword ptr [rbp], 3
0x8954: jmp     0x877a
0x877a
0x8959: nop      dword ptr [rax]
0x8960: endbr64
0x8964: mov     rdi, qword ptr [rip + 0x21f2d]
0x896b: push    rbx
0x896c: test    rdi, rdi
0x896f: je      0x89ac
0x8971: mov     eax, dword ptr [rip + 0x21f1d]
0x8977: test    eax, eax
0x8979: jle     0x899c
0x897b: xor     ebx, ebx
0x897d: nop      dword ptr [rax]
0x8980: mov     rdi, qword ptr [rdi + rbx*8]
0x8984: add     rbx, 1
0x8988: call    0x18d10
0x898d: cmp     dword ptr [rip + 0x21f01], ebx
0x8993: mov     rdi, qword ptr [rip + 0x21efe]
0x899a: jg      0x8980
0x899c: call    0x18d10
0x89a1: mov     qword ptr [rip + 0x21eec], 0
0x89ac: mov     dword ptr [rip + 0x21ede], 0
0x89b6: pop     rbx
0x89b7: mov     dword ptr [rip + 0x21ecf], 0
0x89c1: ret
0x89c2: nop      word ptr cs:[rax + rax]
0x89cd: nop      dword ptr [rax]
0x89d0: endbr64
0x89d4: push    rbp

```

```

0x89d5: mov     rbp, rdi
0x89d8: push    rbx
0x89d9: sub     rsp, 8
0x89dd: mov     esi, dword ptr [rip + 0x21eb1]
0x89e3: mov     rdi, qword ptr [rip + 0x21eae]
0x89ea: mov     dword ptr [rip + 0x21ea0], esi
0x89f0: add     esi, 1
0x89f3: mov     dword ptr [rip + 0x21e9b], esi
0x89f9: movsxd  rsi, esi
0x89fc: shl     rsi, 3
0x8a00: call    0x1d690
0x8a05: movsxd  rdx, dword ptr [rip + 0x21e84]
0x8a0c: mov     rdi, rbp
0x8a0f: mov     qword ptr [rip + 0x21e82], rax
0x8a16: lea     rbx, [rax + rdx*8]
0x8a1a: call    0x1d7a0
0x8a1f: mov     qword ptr [rbx], rax
0x8a22: add     rsp, 8
0x8a26: pop     rbx
0x8a27: pop     rbp
0x8a28: ret
0x8a29: nop     dword ptr [rax]
0x8a30: endbr64
0x8a34: mov     rdx, rsi
0x8a37: mov     esi, dword ptr [rip + 0x21e57]
0x8a3d: mov     eax, dword ptr [rip + 0x21e4d]
0x8a43: lea     ecx, [rsi - 1]
0x8a46: cmp     ecx, eax
0x8a48: jle     0x8a70
0x8a4a: add     eax, 1
0x8a4d: mov     r8d, 1
0x8a53: mov     esi, 0x21
0x8a58: mov     rcx, qword ptr [rip + 0x21e39]
0x8a5f: mov     dword ptr [rip + 0x21e2b], eax
0x8a65: cdqe
0x8a67: mov     rcx, qword ptr [rcx + rax*8]
0x8a6b: jmp     0x13e30
0x13e30
0x8a70: xor     eax, eax
0x8a72: jmp     0xa5a0
0xa5a0
0x8a77: nop     word ptr [rax + rax]
0x8a80: endbr64
0x8a84: mov     eax, dword ptr [rip + 0x21e06]
0x8a8a: mov     rdx, rsi
0x8a8d: test    eax, eax
0x8a8f: je      0x8ac0
0x8a91: sub     eax, 1

```

```

0x8a94: mov     r8d, 1
0x8a9a: mov     esi, 0x21
0x8a9f: mov     rcx, qword ptr [rip + 0x21df2]
0x8aa6: mov     dword ptr [rip + 0x21de4], eax
0x8aac: cdqe
0x8aae: mov     rcx, qword ptr [rcx + rax*8]
0x8ab2: jmp     0x13e30
0x13e30
0x8ab7: nop     word ptr [rax + rax]
0x8ac0: xor     eax, eax
0x8ac2: jmp     0xa5a0
0xa5a0
0x8ac7: nop     word ptr [rax + rax]
0x8ad0: endbr64
0x8ad4: sub     rsp, 8
0x8ad8: xor     edi, edi
0x8ada: call    0xbbc0
0x8adf: xor     eax, eax
0x8ae1: call    0xde40
0x8ae6: xor     eax, eax
0x8ae8: add     rsp, 8
0x8aec: jmp     0x19580
0x19580
0x8af1: nop     word ptr cs:[rax + rax]
0x8afc: nop     dword ptr [rax]
0x8b00: endbr64
0x8b04: push    rbx
0x8b05: xor     eax, eax
0x8b07: mov     rbx, rdi
0x8b0a: call    0x9c90
0x8b0f: cmp     eax, 1
0x8b12: je      0x8b42
0x8b14: mov     rdi, qword ptr [rip + 0x21d65]
0x8b1b: mov     rcx, qword ptr [rip + 0x21dc6]
0x8b22: mov     r8, rbx
0x8b25: xor     eax, eax
0x8b27: lea     rdx, [rip + 0x179e8]
0x8b2e: mov     esi, 1
0x8b33: call    0x4e60
0x8b38: mov     edi, 1
0x8b3d: call    0x4e20
0x8b42: xor     eax, eax
0x8b44: call    0x8ad0
0x8b49: jmp     0x8b14
0x8b14
0x8b4b: nop     dword ptr [rax + rax]
0x8b50: endbr64
0x8b54: push    r15

```

```

0x8b56: push    r14
0x8b58: push    r13
0x8b5a: push    r12
0x8b5c: mov     r12, rsi
0x8b5f: push    rbp
0x8b60: push    rbx
0x8b61: sub     rsp, 0x68
0x8b65: mov     r14, qword ptr [rdi + 8]
0x8b69: mov     qword ptr [rsp + 0x18], rdi
0x8b6e: mov     qword ptr [rsp + 8], rdx
0x8b73: mov     rdi, r14
0x8b76: mov     qword ptr [rsp + 0x10], rcx
0x8b7b: mov     rax, qword ptr fs:[0x28]
0x8b84: mov     qword ptr [rsp + 0x58], rax
0x8b89: xor     eax, eax
0x8b8b: mov     qword ptr [rsp + 0x48], 0
0x8b94: mov     qword ptr [rsp + 0x50], 0
0x8b9d: call    0x49b0
0x8ba2: lea     r13, [rax + 1]
0x8ba6: mov     rdi, r13
0x8ba9: call    0x1d670
0x8bae: mov     qword ptr [r12], rax
0x8bb2: mov     rbx, rax
0x8bb5: movzx   eax, byte ptr [r14]
0x8bb9: test    al, al
0x8bbb: je      0x925c
0x8bc1: mov     dword ptr [rsp + 0x24], 0
0x8bc9: jmp     0x8be5
0x8be5:
0x8bcb: nop     dword ptr [rax + rax]
0x8bd0: mov     byte ptr [rbx], al
0x8bd2: mov     r14, rbp
0x8bd5: add     rbx, 1
0x8bd9: movzx   eax, byte ptr [r14]
0x8bdd: test    al, al
0x8bdf: je      0x8d38
0x8be5: lea     rbp, [r14 + 1]
0x8be9: cmp     al, 0x25
0x8beb: jne     0x8bd0
0x8bed: call    0x4f00
0x8bf2: movsx   rcx, byte ptr [r14 + 1]
0x8bf7: mov     r15, qword ptr [rsp + 0x10]
0x8bfc: mov     rax, qword ptr [rax]
0x8bff: mov     rdx, rcx
0x8c02: test    byte ptr [rax + rcx*2 + 1], 2
0x8c07: lea     eax, [rcx - 0x3f]
0x8c0a: cmovne  r15, qword ptr [rsp + 8]
0x8c10: cmp     al, 0x3b

```

```

0x8c12: ja      0x8c38
0x8c14: lea      rcx, [rip + 0x174f1]
0x8c1b: movzx     eax, al
0x8c1e: movsxd     rax, dword ptr [rcx + rax*4]
0x8c22: add      rax, rcx
0x8c25: jmp      rax
rax
->->->                                indirect JMP
0x8505: mov      eax, dword ptr [rsp + 0x1c]
0x8509: lea      edx, [rax - 1]
0x850c: mov      dword ptr [rsp + 0x1c], edx
0x8510: test     eax, eax
0x8512: jne      0x84f0
0x8514: call     0xd870
0x8519: xor      eax, eax
0x851b: call     0xdea0
0x8520: xor      eax, eax
0x8522: call     0xd740
0x8527: xor      eax, eax
0x8529: call     0xa820
0x852e: mov      rdi, r12
0x8531: call     0xb180
0x8536: test     rax, rax
0x8539: jne      0x848e
0x853f: xor      eax, eax
0x8541: call     0xa5a0
0x8546: xor      eax, eax
0x8548: call     0xdea0
0x854d: xor      eax, eax
0x854f: call     0xd740
0x8554: jmp      0x847d
0x847d
0x8559: nop      dword ptr [rax]
0x8560: mov      rdi, r13
0x8563: call     0xdb10
0x8568: test     rbp, rbp
0x856b: je       0x857b
0x856d: mov      rsi, qword ptr [r13]
0x8571: mov      rdi, rbp
0x8574: xor      eax, eax
0x8576: call     0x7e00
0x857b: mov      rax, qword ptr [r13]
0x857f: mov      rdx, qword ptr [rsp + 0x28]
0x8584: sub      rdx, qword ptr fs:[0x28]
0x858d: jne      0x86b3
0x8593: add      rsp, 0x38
0x8597: pop      rbx
0x8598: pop      rbp

```

```

0x8599: pop    r12
0x859b: pop    r13
0x859d: pop    r14
0x859f: pop    r15
0x85a1: ret
0x85a2: nop    word ptr [rax + rax]
0x85a8: test   rbp, rbp
0x85ab: je     0x8519
0x85b1: lea     eax, [r14 + 1]
0x85b5: mov     rsi, qword ptr [rsp]
0x85b9: mov     rdi, rbp
0x85bc: mov     edx, eax
0x85be: mov     r15d, eax
0x85c1: call    0x19200
0x85c6: test   rax, rax
0x85c9: je     0x86a7
0x85cf: mov     r8, qword ptr [rsp + 0x20]
0x85d4: test   r8, r8
0x85d7: je     0x86a7
0x85dd: xor     edi, edi
0x85df: mov     qword ptr [rsp + 8], r8
0x85e4: mov     r14d, r15d
0x85e7: call    0xd1a0
0x85ec: mov     rdi, qword ptr [rsp + 8]
0x85f1: call    0xd610
0x85f6: xor     eax, eax
0x85f8: call    0xd870
0x85fd: xor     eax, eax
0x85ff: call    0xd740
0x8604: jmp     0x8519
0x8519
0x8609: nop    dword ptr [rax]
0x8610: test   rbp, rbp
0x8613: je     0x8519
0x8619: xor     edi, edi
0x861b: test   r14d, r14d
0x861e: je     0x8680
0x8620: call    0xd1a0
0x8625: sub     r14d, 1
0x8629: je     0x864b
0x862b: mov     rsi, qword ptr [rsp]
0x862f: mov     rdi, rbp
0x8632: mov     edx, r14d
0x8635: call    0x19200
0x863a: mov     rdi, rax
0x863d: test   rax, rax
0x8640: cmovne rdi, qword ptr [rsp + 0x20]
0x8646: call    0xd610

```



```

0x864b: xor     eax, eax
0x864d: call    0xd870
0x8652: xor     eax, eax
0x8654: call    0xd740
0x8659: jmp     0x8519
0x8519
0x865e: nop
0x8660: xor     edi, edi
0x8662: xor     eax, eax
0x8664: call    0x7bd0
0x8669: jmp     0x8519
0x8519
0x866e: nop
0x8670: call    0xa5a0
0x8675: jmp     0x8505
0x8505
0x867a: nop     word ptr [rax + rax]
0x8680: call    0xd1a0
0x8685: xor     eax, eax
0x8687: call    0xd870
0x868c: xor     eax, eax
0x868e: call    0xd740
0x8693: jmp     0x8519
0x8519
0x8698: nop     dword ptr [rax + rax]
0x86a0: xor     eax, eax
0x86a2: jmp     0x857f
0x857f
0x86a7: xor     eax, eax
0x86a9: call    0xa5a0
0x86ae: jmp     0x8519
0x8519
0x86b3: call    0x49d0
0x86b8: nop     dword ptr [rax + rax]
0x86c0: endbr64
0x86c4: push    r15
0x86c6: push    r14
0x86c8: push    r13
0x86ca: push    r12
0x86cc: push    rbp
0x86cd: push    rbx
0x86ce: sub     rsp, 0x18
0x86d2: test    edx, edx
0x86d4: je      0x8890
0x86da: mov     ebx, edx
0x86dc: cmp     edx, 3
0x86df: je      0x88f8
0x86e5: mov     rbp, rcx

```

```

0x86e8: test    rcx, rcx
0x86eb: je      0x8864
0x86f1: mov     dword ptr [rcx], 0
0x86f7: xor     eax, eax
0x86f9: mov     r13, rsi
0x86fc: lea     r15, [rip + 0x2091d]
0x8703: movabs  r14, 0xea0ea0ea0ea0ea1
0x870d: call    0xd870
0x8712: xor     eax, eax
0x8714: call    0xd740
0x8719: xor     eax, eax
0x871b: call    0xa820
0x8720: xor     edi, edi
0x8722: call    0xb180
0x8727: mov     r12, rax
0x872a: test    rax, rax
0x872d: je      0x8928
0x8733: mov     rax, qword ptr [rax]
0x8736: movzx   edi, byte ptr [rax]
0x8739: mov     rax, qword ptr [r12 + 0x10]
0x873e: test    rax, rax
0x8741: je      0x874d
0x8743: cmp     byte ptr [rax + 0x1b], 0
0x8747: jne     0x8808
0x874d: cmp     edi, -7
0x8750: jge     0x87b0
0x8752: cmp     edi, -0x3d
0x8755: je      0x8840
0x875b: cmp     edi, -0x32
0x875e: jne     0x88bb
0x8764: xor     eax, eax
0x8766: call    0xca40
0x876b: test    eax, eax
0x876d: jne     0x8880
0x8773: mov     dword ptr [rbp], 1
0x877a: xor     eax, eax
0x877c: call    0xdea0
0x8781: xor     eax, eax
0x8783: call    0xd870
0x8788: xor     eax, eax
0x878a: call    0xd740
0x878f: xor     eax, eax
0x8791: call    0xa820
0x8796: mov     rdi, r13
0x8799: call    0xdb10
0x879e: mov     r12, qword ptr [r13]
0x87a2: jmp     0x8867
0x8867

```

```

0x87a7: nop      word ptr [rax + rax]
0x87b0: cmp      edi, 7
0x87b3: je        0x8840
0x87b9: mov      dword ptr [rsp + 8], edi
0x87bd: call     0x4f00
0x87c2: mov      r8, rax
0x87c5: movsxd   rax, dword ptr [rsp + 8]
0x87ca: mov      rdx, qword ptr [r8]
0x87cd: mov      rdi, rax
0x87d0: test     byte ptr [rdx + rax*2 + 1], 0x40
0x87d5: jne      0x8948
0x87db: xor      eax, eax
0x87dd: xor      r12d, r12d
0x87e0: call     0xdea0
0x87e5: xor      eax, eax
0x87e7: call     0xd870
0x87ec: xor      eax, eax
0x87ee: call     0xd740
0x87f3: xor      eax, eax
0x87f5: call     0xa820
0x87fa: mov      rdi, r13
0x87fd: call     0xdb10
0x8802: jmp      0x8867
0x8867
0x8804: nop      dword ptr [rax]
0x8808: mov      rcx, qword ptr [rax]
0x880b: sub      rcx, r15
0x880e: mov      rax, rcx
0x8811: sar      rcx, 0x3f
0x8815: imul     r14
0x8818: sar      rdx, 1
0x881b: sub      rdx, rcx
0x881e: mov      edi, edx
0x8820: not      edi
0x8822: cmp      edx, 7
0x8825: jne      0x874d
0x882b: xor      edi, edi
0x882d: xor      eax, eax
0x882f: call     0x7bd0
0x8834: jmp      0x8720
0x8720
0x8839: nop      dword ptr [rax]
0x8840: xor      eax, eax
0x8842: call     0xdea0
0x8847: xor      eax, eax
0x8849: call     0xd870
0x884e: xor      eax, eax
0x8850: call     0xd740

```

```

0x8855: xor     eax, eax
0x8857: call    0xa820
0x885c: mov     rdi, r13
0x885f: call    0xdb10
0x8864: xor     r12d, r12d
0x8867: add     rsp, 0x18
0x886b: mov     rax, r12
0x886e: pop     rbx
0x886f: pop     rbp
0x8870: pop     r12
0x8872: pop     r13
0x8874: pop     r14
0x8876: pop     r15
0x8878: ret
0x8879: nop     dword ptr [rax]
0x8880: xor     eax, eax
0x8882: call    0xa5a0
0x8887: jmp     0x877a
0x877a:
0x888c: nop     dword ptr [rax]
0x8890: xor     eax, eax
0x8892: mov     qword ptr [rsp + 8], rdi
0x8897: call    0xc8c0
0x889c: mov     qword ptr [rip + 0x2205d], rax
0x88a3: xor     eax, eax
0x88a5: call    0xcff0
0x88aa: mov     rdi, qword ptr [rsp + 8]
0x88af: test    rdi, rdi
0x88b2: je      0x8864
0x88b4: call    0xd540
0x88b9: jmp     0x8864
0x8864:
0x88bb: cmp     edi, -0x44
0x88be: jne     0x88df
0x88c0: cmp     ebx, 1
0x88c3: jne     0x88df
0x88c5: xor     eax, eax
0x88c7: call    0xca40
0x88cc: cmp     eax, 1
0x88cf: mov     eax, 2
0x88d4: sbb     eax, -1
0x88d7: mov     dword ptr [rbp], eax
0x88da: jmp     0x877a
0x877a:
0x88df: cmp     edi, -0x45
0x88e2: jne     0x87b9
0x88e8: cmp     ebx, 2
0x88eb: je      0x88c5

```

```

0x88ed: jmp      0x87b9
0x87b9
0x88f2: nop      word ptr [rax + rax]
0x88f8: mov      rdi, qword ptr [rip + 0x22001]
0x88ff: xor      r12d, r12d
0x8902: call     0xc970
0x8907: xor      eax, eax
0x8909: call     0xd870
0x890e: xor      eax, eax
0x8910: call     0xd740
0x8915: xor      eax, eax
0x8917: call     0xa820
0x891c: jmp      0x8867
0x8867
0x8921: nop      dword ptr [rax]
0x8928: xor      eax, eax
0x892a: call     0xdea0
0x892f: xor      eax, eax
0x8931: call     0xd740
0x8936: xor      eax, eax
0x8938: call     0xa820
0x893d: jmp      0x8867
0x8867
0x8942: nop      word ptr [rax + rax]
0x8948: call     0xcd30
0x894d: mov      dword ptr [rbp], 3
0x8954: jmp      0x877a
0x877a
0x8959: nop      dword ptr [rax]
0x8960: endbr64
0x8964: mov      rdi, qword ptr [rip + 0x21f2d]
0x896b: push     rbx
0x896c: test     rdi, rdi
0x896f: je       0x89ac
0x8971: mov      eax, dword ptr [rip + 0x21f1d]
0x8977: test     eax, eax
0x8979: jle      0x899c
0x897b: xor      ebx, ebx
0x897d: nop      dword ptr [rax]
0x8980: mov      rdi, qword ptr [rdi + rbx*8]
0x8984: add      rbx, 1
0x8988: call     0x18d10
0x898d: cmp      dword ptr [rip + 0x21f01], ebx
0x8993: mov      rdi, qword ptr [rip + 0x21efe]
0x899a: jg       0x8980
0x899c: call     0x18d10
0x89a1: mov      qword ptr [rip + 0x21eec], 0
0x89ac: mov      dword ptr [rip + 0x21ede], 0

```

```

0x89b6: pop      rbx
0x89b7: mov      dword ptr [rip + 0x21ecf], 0
0x89c1: ret
0x89c2: nop      word ptr cs:[rax + rax]
0x89cd: nop      dword ptr [rax]
0x89d0: endbr64
0x89d4: push     rbp
0x89d5: mov      rbp, rdi
0x89d8: push     rbx
0x89d9: sub      rsp, 8
0x89dd: mov      esi, dword ptr [rip + 0x21eb1]
0x89e3: mov      rdi, qword ptr [rip + 0x21eae]
0x89ea: mov      dword ptr [rip + 0x21ea0], esi
0x89f0: add      esi, 1
0x89f3: mov      dword ptr [rip + 0x21e9b], esi
0x89f9: movsxd   rsi, esi
0x89fc: shl      rsi, 3
0x8a00: call     0x1d690
0x8a05: movsxd   rdx, dword ptr [rip + 0x21e84]
0x8a0c: mov      rdi, rbp
0x8a0f: mov      qword ptr [rip + 0x21e82], rax
0x8a16: lea      rbx, [rax + rdx*8]
0x8a1a: call     0x1d7a0
0x8a1f: mov      qword ptr [rbx], rax
0x8a22: add      rsp, 8
0x8a26: pop      rbx
0x8a27: pop      rbp
0x8a28: ret
0x8a29: nop      dword ptr [rax]
0x8a30: endbr64
0x8a34: mov      rdx, rsi
0x8a37: mov      esi, dword ptr [rip + 0x21e57]
0x8a3d: mov      eax, dword ptr [rip + 0x21e4d]
0x8a43: lea      ecx, [rsi - 1]
0x8a46: cmp      ecx, eax
0x8a48: jle      0x8a70
0x8a4a: add      eax, 1
0x8a4d: mov      r8d, 1
0x8a53: mov      esi, 0x21
0x8a58: mov      rcx, qword ptr [rip + 0x21e39]
0x8a5f: mov      dword ptr [rip + 0x21e2b], eax
0x8a65: cdqe
0x8a67: mov      rcx, qword ptr [rcx + rax*8]
0x8a6b: jmp      0x13e30
0x13e30
0x8a70: xor      eax, eax
0x8a72: jmp      0xa5a0
0xa5a0

```

```

0x8a77: nop      word ptr [rax + rax]
0x8a80: endbr64
0x8a84: mov      eax, dword ptr [rip + 0x21e06]
0x8a8a: mov      rdx, rsi
0x8a8d: test     eax, eax
0x8a8f: je       0x8ac0
0x8a91: sub      eax, 1
0x8a94: mov      r8d, 1
0x8a9a: mov      esi, 0x21
0x8a9f: mov      rcx, qword ptr [rip + 0x21df2]
0x8aa6: mov      dword ptr [rip + 0x21de4], eax
0x8aac: cdqe
0x8aae: mov      rcx, qword ptr [rcx + rax*8]
0x8ab2: jmp      0x13e30
0x13e30
0x8ab7: nop      word ptr [rax + rax]
0x8ac0: xor      eax, eax
0x8ac2: jmp      0xa5a0
0xa5a0
0x8ac7: nop      word ptr [rax + rax]
0x8ad0: endbr64
0x8ad4: sub      rsp, 8
0x8ad8: xor      edi, edi
0x8ada: call     0xbbc0
0x8adf: xor      eax, eax
0x8ae1: call     0xde40
0x8ae6: xor      eax, eax
0x8ae8: add      rsp, 8
0x8aec: jmp      0x19580
0x19580
0x8af1: nop      word ptr cs:[rax + rax]
0x8afc: nop      dword ptr [rax]
0x8b00: endbr64
0x8b04: push     rbx
0x8b05: xor      eax, eax
0x8b07: mov      rbx, rdi
0x8b0a: call     0x9c90
0x8b0f: cmp      eax, 1
0x8b12: je       0x8b42
0x8b14: mov      rdi, qword ptr [rip + 0x21d65]
0x8b1b: mov      rcx, qword ptr [rip + 0x21dc6]
0x8b22: mov      r8, rbx
0x8b25: xor      eax, eax
0x8b27: lea      rdx, [rip + 0x179e8]
0x8b2e: mov      esi, 1
0x8b33: call     0x4e60
0x8b38: mov      edi, 1
0x8b3d: call     0x4e20

```

```

0x8b42: xor      eax, eax
0x8b44: call     0x8ad0
0x8b49: jmp      0x8b14
0x8b14:
0x8b4b: nop      dword ptr [rax + rax]
0x8b50: endbr64
0x8b54: push     r15
0x8b56: push     r14
0x8b58: push     r13
0x8b5a: push     r12
0x8b5c: mov      r12, rsi
0x8b5f: push     rbp
0x8b60: push     rbx
0x8b61: sub      rsp, 0x68
0x8b65: mov      r14, qword ptr [rdi + 8]
0x8b69: mov      qword ptr [rsp + 0x18], rdi
0x8b6e: mov      qword ptr [rsp + 8], rdx
0x8b73: mov      rdi, r14
0x8b76: mov      qword ptr [rsp + 0x10], rcx
0x8b7b: mov      rax, qword ptr fs:[0x28]
0x8b84: mov      qword ptr [rsp + 0x58], rax
0x8b89: xor      eax, eax
0x8b8b: mov      qword ptr [rsp + 0x48], 0
0x8b94: mov      qword ptr [rsp + 0x50], 0
0x8b9d: call     0x49b0
0x8ba2: lea      r13, [rax + 1]
0x8ba6: mov      rdi, r13
0x8ba9: call     0x1d670
0x8bae: mov      qword ptr [r12], rax
0x8bb2: mov      rbx, rax
0x8bb5: movzx    eax, byte ptr [r14]
0x8bb9: test     al, al
0x8bbb: je       0x925c
0x8bc1: mov      dword ptr [rsp + 0x24], 0
0x8bc9: jmp      0x8be5
0x8be5:
0x8bcb: nop      dword ptr [rax + rax]
0x8bd0: mov      byte ptr [rbx], al
0x8bd2: mov      r14, rbp
0x8bd5: add      rbx, 1
0x8bd9: movzx    eax, byte ptr [r14]
0x8bdd: test     al, al
0x8bdf: je       0x8d38
0x8be5: lea      rbp, [r14 + 1]
0x8be9: cmp      al, 0x25
0x8beb: jne      0x8bd0
0x8bed: call     0x4f00
0x8bf2: movsx    rcx, byte ptr [r14 + 1]

```



```

0x8bf7: mov     r15, qword ptr [rsp + 0x10]
0x8bfc: mov     rax, qword ptr [rax]
0x8bff: mov     rdx, rcx
0x8c02: test    byte ptr [rax + rcx*2 + 1], 2
0x8c07: lea     eax, [rcx - 0x3f]
0x8c0a: cmovne  r15, qword ptr [rsp + 8]
0x8c10: cmp     al, 0x3b
0x8c12: ja      0x8c38
0x8c14: lea     rcx, [rip + 0x174f1]
0x8c1b: movzx   eax, al
0x8c1e: movsxd  rax, dword ptr [rcx + rax*4]
0x8c22: add     rax, rcx
0x8c25: jmp     rax
rax
->->->                                indirect JMP
0x13f50: mov     rax, qword ptr [rsp + 0x38]
0x13f55: sub     rax, qword ptr fs:[0x28]
0x13f5e: jne     0x14ce9
0x13f64: add     rsp, 0x48
0x13f68: xor     eax, eax
0x13f6a: pop     rbx
0x13f6b: pop     rbp
0x13f6c: pop     r12
0x13f6e: pop     r13
0x13f70: pop     r14
0x13f72: pop     r15
0x13f74: ret
0x13f75: nop     dword ptr [rax]
0x13f78: movsxd  rax, dword ptr [rdi + 0x28]
0x13f7c: mov     esi, 0x2e
0x13f81: lea     rax, [rax + rax*4]
0x13f85: shl     rax, 4
0x13f89: add     rax, qword ptr [rdi + 0x38]
0x13f8d: mov     r12, qword ptr [rax]
0x13f90: mov     rdi, r12
0x13f93: call    0x4a20
0x13f98: mov     rbx, rax
0x13f9b: test    rax, rax
0x13f9e: je      0x13f50
0x13fa0: cmp     r12, rax
0x13fa3: je      0x13f50
0x13fa5: mov     rdi, rax
0x13fa8: call    0x49b0
0x13fad: lea     rdi, [rax + 2]
0x13fb1: call    0x4c20
0x13fb6: mov     r8, rbx
0x13fb9: mov     esi, 1
0x13fbe: xor     ebx, ebx

```

```

0x13fc0:    mov     rdi, rax
0x13fc3:    mov     r13, rax
0x13fc6:    xor     eax, eax
0x13fc8:    mov     rdx, -1
0x13fcf:    lea     rcx, [rip + 0xd3be]
0x13fd6:    call    0x4f10
0x13fdb:    mov     eax, dword ptr [rbp + 0x68]
0x13fde:    test    eax, eax
0x13fe0:    jle     0x14132
0x13fe6:    nop     word ptr cs:[rax + rax]
0x13ff0:    mov     rax, qword ptr [rbp + 0x38]
0x13ff4:    lea     r12, [rbx + rbx*4]
0x13ff8:    shl     r12, 4
0x13ffc:    add     rax, r12
0x13fff:    cmp     byte ptr [rax + 0x48], 0
0x14003:    je      0x14048
0x14005:    cmp     dword ptr [rip + 0x17618], 1
0x1400c:    mov     rsi, qword ptr [rax]
0x1400f:    mov     rdi, r13
0x14012:    sbb     edx, edx
0x14014:    and     edx, 4
0x14017:    add     edx, 1
0x1401a:    call    0x4a40
0x1401f:    test    eax, eax
0x14021:    jne     0x14048
0x14023:    add     r12, qword ptr [rbp + 0x38]
0x14027:    movzx   eax, byte ptr [r12 + 0x49]
0x1402d:    cmp     r15d, 0x32
0x14031:    je      0x149b3
0x14037:    test    al, al
0x14039:    je      0x14048
0x1403b:    mov     byte ptr [r12 + 0x49], 0
0x14041:    sub     dword ptr [rbp + 0x70], 1
0x14045:    nop     dword ptr [rax]
0x14048:    add     rbx, 1
0x1404c:    cmp     dword ptr [rbp + 0x68], ebx
0x1404f:    jg      0x13ff0
0x14051:    jmp     0x14132
0x14132:
0x14056:    nop     word ptr cs:[rax + rax]
0x14060:    xor     eax, eax
0x14062:    mov     rdi, rcx
0x14065:    call    0x13ca0
0x1406a:    cmp     qword ptr [rax], 0
0x1406e:    mov     r14, rax
0x14071:    mov     r13, rax
0x14074:    je      0x1412a
0x1407a:    nop     word ptr [rax + rax]

```

```

0x14080:    mov     edx, dword ptr [rbp + 0x68]
0x14083:    xor     ebx, ebx
0x14085:    test    edx, edx
0x14087:    jle     0x140f9
0x14089:    nop     dword ptr [rax]
0x14090:    mov     rax, qword ptr [rbp + 0x38]
0x14094:    lea     r12, [rbx + rbx*4]
0x14098:    shl     r12, 4
0x1409c:    add     rax, r12
0x1409f:    cmp     byte ptr [rax + 0x48], 0
0x140a3:    je      0x140f0
0x140a5:    cmp     dword ptr [rip + 0x17578], 1
0x140ac:    mov     rsi, qword ptr [rax]
0x140af:    sbb     edx, edx
0x140b1:    mov     rdi, qword ptr [r13]
0x140b5:    and     edx, 4
0x140b8:    add     edx, 1
0x140bb:    call    0x4a40
0x140c0:    test    eax, eax
0x140c2:    jne     0x140f0
0x140c4:    add     r12, qword ptr [rbp + 0x38]
0x140c8:    movzx   eax, byte ptr [r12 + 0x49]
0x140ce:    cmp     r15d, 0x24
0x140d2:    je      0x14840
0x140d8:    test    al, al
0x140da:    je      0x140f0
0x140dc:    mov     byte ptr [r12 + 0x49], 0
0x140e2:    sub     dword ptr [rbp + 0x70], 1
0x140e6:    nop     word ptr cs:[rax + rax]
0x140f0:    add     rbx, 1
0x140f4:    cmp     dword ptr [rbp + 0x68], ebx
0x140f7:    jg      0x14090
0x140f9:    add     r13, 8
0x140fd:    cmp     qword ptr [r13], 0
0x14102:    jne     0x14080
0x14108:    mov     rdi, qword ptr [r14]
0x1410b:    test    rdi, rdi
0x1410e:    je      0x1412a
0x14110:    mov     rbx, r14
0x14113:    nop     dword ptr [rax + rax]
0x14118:    call    0x18d10
0x1411d:    mov     rdi, qword ptr [rbx + 8]
0x14121:    add     rbx, 8
0x14125:    test    rdi, rdi
0x14128:    jne     0x14118
0x1412a:    mov     rdi, r14
0x1412d:    call    0x18d10
0x14132:    cmp     byte ptr [rbp + 0x6d], 0

```

```

0x14136:      je      0x13f50
0x1413c:      mov     rdi, rbp
0x1413f:      xor     eax, eax
0x14141:      call    0x10d70
0x14146:      cmp     byte ptr [rbp + 0x6d], 0
0x1414a:      je      0x13f50
0x14150:      mov     rdi, rbp
0x14153:      xor     eax, eax
0x14155:      call    0xec50
0x1415a:      jmp     0x13f50
0x13f50
0x1415f:      mov     rax, qword ptr [rdi + 0x18]
0x14163:      cmp     byte ptr [rax], 0x2f
0x14166:      je      0x149a4
0x1416c:      mov     esi, dword ptr [rbp + 0x28]
0x1416f:      test    esi, esi
0x14171:      je      0x14187
0x14173:      mov     dword ptr [rbp + 0x28], 0
0x1417a:      mov     rdi, rbp
0x1417d:      xor     eax, eax
0x1417f:      call    0xf110
0x14184:      mov     dword ptr [rbp + 0x2c], eax
0x14187:      mov     rax, qword ptr [rsp + 0x38]
0x1418c:      sub     rax, qword ptr fs:[0x28]
0x14195:      jne     0x14ce9
0x1419b:      add     rsp, 0x48
0x1419f:      mov     rsi, r12
0x141a2:      mov     rdi, rbp
0x141a5:      xor     eax, eax
0x141a7:      pop     rbx
0x141a8:      pop     rbp
0x141a9:      pop     r12
0x141ab:      pop     r13
0x141ad:      pop     r14
0x141af:      pop     r15
0x141b1:      jmp     0x162b0
0x162b0
0x141b6:      mov     edx, dword ptr [rdi + 0x68]
0x141b9:      test    edx, edx
0x141bb:      jle     0x141ec
0x141bd:      sub     edx, 1
0x141c0:      mov     rcx, qword ptr [rdi + 0x38]
0x141c4:      lea     rdx, [rdx + rdx*4]
0x141c8:      shl     rdx, 4
0x141cc:      lea     rax, [rcx + 0x49]
0x141d0:      lea     rdx, [rcx + rdx + 0x99]
0x141d8:      nop     dword ptr [rax + rax]
0x141e0:      mov     byte ptr [rax], 0

```

```

0x141e3:      add     rax, 0x50
0x141e7:      cmp     rdx, rax
0x141ea:      jne     0x141e0
0x141ec:      cmp     byte ptr [rbp + 0x6d], 0
0x141f0:      mov     dword ptr [rbp + 0x70], 0
0x141f7:      je      0x13f50
0x141fd:      jmp     0x1413c
0x1413c:
0x14202:      movzx   edx, byte ptr [rdi + 0x6d]
0x14206:      mov     ebx, r8d
0x14209:      xor     r13d, r13d
0x1420c:      xor     r12d, r12d
0x1420f:      test    r8d, r8d
0x14212:      jne     0x14249
0x14214:      jmp     0x1488f
0x1488f:
0x14219:      nop     dword ptr [rax]
0x14220:      test    r12d, r12d
0x14223:      jne     0x14240
0x14225:      mov     r12d, 1
0x1422b:      test    dl, dl
0x1422d:      je      0x14240
0x1422f:      mov     rdi, rbp
0x14232:      call    0x10760
0x14237:      movzx   edx, byte ptr [rbp + 0x6d]
0x1423b:      nop     dword ptr [rax + rax]
0x14240:      sub     ebx, 1
0x14243:      je      0x14870
0x14249:      mov     esi, dword ptr [rbp + 0x28]
0x1424c:      test    esi, esi
0x1424e:      je      0x14870
0x14254:      lea     eax, [rsi - 1]
0x14257:      mov     dword ptr [rbp + 0x28], eax
0x1425a:      cmp     esi, dword ptr [rbp + 0x2c]
0x1425d:      jne     0x14220
0x1425f:      sub     esi, dword ptr [rbp + 0x88]
0x14265:      mov     eax, 0
0x1426a:      mov     r13d, 1
0x14270:      cmovs   esi, eax
0x14273:      mov     dword ptr [rbp + 0x2c], esi
0x14276:      jmp     0x14240
0x14240:
0x14278:      mov     rsi, r12
0x1427b:      xor     eax, eax
0x1427d:      call    0x15bb0
0x14282:      jmp     0x13f50
0x13f50:
0x14287:      mov     rsi, r12

```

```

0x1428a:    xor     eax, eax
0x1428c:    call   0x15910
0x14291:    jmp     0x13f50
0x13f50:
0x14296:    mov     edx, dword ptr [rcx]
0x14298:    mov     rsi, r12
0x1429b:    xor     eax, eax
0x1429d:    call   0x167a0
0x142a2:    jmp     0x13f50
0x13f50:
0x142a7:    mov     edx, dword ptr [rcx]
0x142a9:    mov     rsi, r12
0x142ac:    xor     eax, eax
0x142ae:    call   0x12fd0
0x142b3:    jmp     0x13f50
0x13f50:
0x142b8:    mov     rdi, qword ptr [rdi + 0x48]
0x142bc:    call   0x19120
0x142c1:    mov     qword ptr [rbp + 0x48], 0
0x142c9:    jmp     0x13f50
0x13f50:
0x142ce:    mov     eax, dword ptr [rdi + 0x30]
0x142d1:    mov     edx, dword ptr [rdi + 0x64]
0x142d4:    mov     esi, 4
0x142d9:    add     r8d, dword ptr [rdi + 0x74]
0x142dd:    add     eax, 1
0x142e0:    cmp     edx, 0x27
0x142e3:    jle     0x14300
0x142e5:    movsxd  rcx, dword ptr [rdi + 0x80]
0x142ec:    cmp     edx, 0x49
0x142ef:    jg      0x142f6
0x142f1:    cmp     ecx, 5
0x142f4:    je      0x14300
0x142f6:    lea     rsi, [rip + 0xd6e3]
0x142fd:    mov     esi, dword ptr [rsi + rcx*4]
0x14300:    sub     edx, esi
0x14302:    sub     eax, edx
0x14304:    mov     edx, 0
0x14309:    cmovs   eax, edx
0x1430c:    cmp     eax, r8d
0x1430f:    cmovg   eax, r8d
0x14313:    mov     dword ptr [rbp + 0x74], eax
0x14316:    mov     rdi, rbp
0x14319:    xor     eax, eax
0x1431b:    call   0x10ec0
0x14320:    xor     eax, eax
0x14322:    mov     rdi, rbp
0x14325:    call   0x10160

```

```

0x1432a:      cmp     byte ptr [rbp + 0x6d], 0
0x1432e:      je      0x1433a
0x14330:      mov     rdi, rbp
0x14333:      xor     eax, eax
0x14335:      call    0xec50
0x1433a:      mov     rdi, rbp
0x1433d:      xor     eax, eax
0x1433f:      call    0x10240
0x14344:      jmp     0x13f40
0x13f40
0x14349:      sub     dword ptr [rdi + 0x74], r8d
0x1434d:      mov     eax, dword ptr [rdi + 0x74]
0x14350:      xor     edx, edx
0x14352:      test    eax, eax
0x14354:      cmovs   eax, edx
0x14357:      jmp     0x14313
0x14313
0x14359:      xor     esi, esi
0x1435b:      mov     rdi, rcx
0x1435e:      mov     edx, 0xa
0x14363:      call    0x4ba0
0x14368:      mov     rsi, r12
0x1436b:      mov     rdi, rbp
0x1436e:      movsxd  rdx, eax
0x14371:      xor     eax, eax
0x14373:      call    0x137c0
0x14378:      jmp     0x13f50
0x13f50
0x1437d:      mov     rsi, r12
0x14380:      xor     eax, eax
0x14382:      call    0x12ca0
0x14387:      jmp     0x13f50
0x13f50
0x1438c:      mov     eax, dword ptr [rdi + 0x60]
0x1438f:      mov     edx, dword ptr [r12 + 0x60]
0x14394:      mov     dword ptr [rdi + 0x60], edx
0x14397:      mov     edx, dword ptr [r12 + 0x64]
0x1439c:      mov     dword ptr [r12 + 0x60], eax
0x143a1:      mov     eax, dword ptr [rdi + 0x64]
0x143a4:      mov     dword ptr [rdi + 0x64], edx
0x143a7:      mov     edx, dword ptr [r12 + 0x58]
0x143ac:      mov     dword ptr [r12 + 0x64], eax
0x143b1:      mov     eax, dword ptr [rdi + 0x58]
0x143b4:      mov     dword ptr [rdi + 0x58], edx
0x143b7:      mov     edx, dword ptr [r12 + 0x5c]
0x143bc:      mov     dword ptr [r12 + 0x58], eax
0x143c1:      mov     eax, dword ptr [rdi + 0x5c]
0x143c4:      mov     dword ptr [rdi + 0x5c], edx

```

```

0x143c7:      mov     dword ptr [r12 + 0x5c], eax
0x143cc:      movsxd  rcx, dword ptr [rdi + 0x60]
0x143d0:      movsxd  rdx, dword ptr [rdi + 0x5c]
0x143d4:      movsxd  rsi, dword ptr [rdi + 0x58]
0x143d8:      movsxd  r8, dword ptr [rdi + 0x64]
0x143dc:      call    0xf590
0x143e1:      movsxd  rcx, dword ptr [r12 + 0x60]
0x143e6:      movsxd  rdx, dword ptr [r12 + 0x5c]
0x143eb:      mov     rdi, r12
0x143ee:      movsxd  rsi, dword ptr [r12 + 0x58]
0x143f3:      movsxd  r8, dword ptr [r12 + 0x64]
0x143f8:      call    0xf590
0x143fd:      jmp     0x13f50
0x13f50
0x14402:      mov     rsi, r12
0x14405:      xor     eax, eax
0x14407:      call    0x14e20
0x1440c:      jmp     0x13f50
0x13f50
0x14411:      mov     eax, dword ptr [rdi + 0x84]
0x14417:      add     eax, 1
0x1441a:      movsxd  r15, eax
0x1441d:      cdq
0x1441e:      imul    r15, r15, 0x38e38e39
0x14425:      sar     r15, 0x21
0x14429:      sub     r15d, edx
0x1442c:      lea     edx, [r15 + r15*8]
0x14430:      sub     eax, edx
0x14432:      mov     r15d, eax
0x14435:      jmp     0x13e84
0x13e84
0x1443a:      mov     eax, dword ptr [rdi + 0x80]
0x14440:      add     eax, 1
0x14443:      movsxd  r15, eax
0x14446:      cdq
0x14447:      imul    r15, r15, 0x2aaaaaab
0x1444e:      shr     r15, 0x20
0x14452:      sub     r15d, edx
0x14455:      lea     edx, [r15 + r15*2]
0x14459:      add     edx, edx
0x1445b:      sub     eax, edx
0x1445d:      mov     r15d, eax
0x14460:      jmp     0x13f1c
0x13f1c
0x14465:      mov     edx, dword ptr [rdi + 0x68]
0x14468:      mov     dword ptr [rdi + 0x70], 0
0x1446f:      test    edx, edx
0x14471:      jle     0x14132

```



```

0x14477:      sub     edx, 1
0x1447a:      mov     rcx, qword ptr [rdi + 0x38]
0x1447e:      lea     rdx, [rdx + rdx*4]
0x14482:      shl     rdx, 4
0x14486:      lea     rax, [rcx + 0x49]
0x1448a:      lea     rcx, [rcx + rdx + 0x99]
0x14492:      nop     word ptr [rax + rax]
0x14498:      cmp     byte ptr [rax - 1], 0
0x1449c:      je      0x144ac
0x1449e:      cmp     byte ptr [rax], 0
0x144a1:      sete    dl
0x144a4:      mov     byte ptr [rax], dl
0x144a6:      movzx   edx, dl
0x144a9:      add     dword ptr [rbp + 0x70], edx
0x144ac:      add     rax, 0x50
0x144b0:      cmp     rcx, rax
0x144b3:      jne     0x14498
0x144b5:      jmp     0x14132
0x14132:
0x144ba:      mov     rsi, rcx
0x144bd:      xor     eax, eax
0x144bf:      call    0x161d0
0x144c4:      jmp     0x13f50
0x13f50:
0x144c9:      mov     rdx, rcx
0x144cc:      mov     rsi, r12
0x144cf:      xor     eax, eax
0x144d1:      call    0x165f0
0x144d6:      jmp     0x13f50
0x13f50:
0x144db:      mov     rcx, qword ptr [rcx + 8]
0x144df:      mov     rdi, rcx
0x144e2:      mov     qword ptr [rsp + 8], rcx
0x144e7:      call    0x49b0
0x144ec:      mov     rcx, qword ptr [rsp + 8]
0x144f1:      mov     qword ptr [rsp + 0x10], rax
0x144f6:      mov     r14, rax
0x144f9:      mov     eax, dword ptr [r13]
0x144fd:      cmp     eax, 2
0x14500:      je      0x14bb4
0x14506:      cmp     eax, 3
0x14509:      je      0x14aea
0x1450f:      cmp     eax, 1
0x14512:      je      0x148db
0x14518:      mov     rax, qword ptr [rbp + 0x40]
0x1451c:      mov     qword ptr [r13 + 0x10], rax
0x14520:      jmp     0x13f50
0x13f50:

```

```

0x14525:    mov     rdi, qword ptr [rcx + 8]
0x14529:    mov     qword ptr [rsp + 8], rdi
0x1452e:    call    0x49b0
0x14533:    mov     rdi, qword ptr [rsp + 8]
0x14538:    mov     qword ptr [rsp + 0x10], rax
0x1453d:    mov     r14, rax
0x14540:    mov     eax, dword ptr [r13]
0x14544:    cmp     eax, 2
0x14547:    je      0x14a96
0x1454d:    cmp     eax, 3
0x14550:    je      0x149de
0x14556:    cmp     eax, 1
0x14559:    je      0x148db
0x1455f:    mov     rax, qword ptr [rbp + 0x40]
0x14563:    mov     qword ptr [rsp + 0x10], rax
0x14568:    mov     rax, qword ptr [rsp + 0x10]
0x1456d:    mov     qword ptr [r13 + 0x10], rax
0x14571:    jmp     0x13f50
0x13f50:
0x14576:    mov     edi, 0x10
0x1457b:    call    0x19100
0x14580:    lea     rsi, [rsp + 0x20]
0x14585:    mov     qword ptr [rsp + 0x28], 0
0x1458e:    mov     qword ptr [rbp + 0x48], rax
0x14592:    mov     rdi, rax
0x14595:    mov     eax, dword ptr [rbp + 0x28]
0x14598:    mov     dword ptr [rsp + 0x20], eax
0x1459c:    call    0x19150
0x145a1:    mov     qword ptr [rbp + 0x40], 0
0x145a9:    mov     byte ptr [rbp + 0x6e], 0
0x145ad:    jmp     0x13f50
0x13f50:
0x145b2:    xor     esi, esi
0x145b4:    mov     edx, 0xa
0x145b9:    mov     rdi, rcx
0x145bc:    call    0x4ba0
0x145c1:    test    eax, eax
0x145c3:    jle     0x13f50
0x145c9:    mov     ecx, dword ptr [rbp + 0x60]
0x145cc:    lea     edx, [rcx - 1]
0x145cf:    cmp     edx, eax
0x145d1:    jle     0x13f50
0x145d7:    mov     dword ptr [r12 + 0x88], eax
0x145df:    mov     dword ptr [rbp + 0x88], eax
0x145e5:    jmp     0x13f50
0x13f50:
0x145ea:    mov     eax, dword ptr [rdi + 0x68]
0x145ed:    sub     eax, 1

```

```

0x145f0:    cmp     dword ptr [rdi + 0x28], eax
0x145f3:    je      0x13f50
0x145f9:    mov     dword ptr [rdi + 0x28], eax
0x145fc:    mov     rdi, rbp
0x145ff:    xor     eax, eax
0x14601:    call    0xf110
0x14606:    mov     dword ptr [rbp + 0x2c], eax
0x14609:    jmp     0x14132
0x14132:
0x1460e:    mov     ecx, dword ptr [rdi + 0x28]
0x14611:    test    ecx, ecx
0x14613:    je      0x13f50
0x14619:    mov     dword ptr [rdi + 0x28], 0
0x14620:    jmp     0x145fc
0x145fc:
0x14622:    mov     r9d, dword ptr [rdi + 0x68]
0x14626:    mov     r10d, dword ptr [rdi + 0x28]
0x1462a:    lea     eax, [r9 - 1]
0x1462e:    cmp     r10d, eax
0x14631:    je      0x13f50
0x14637:    mov     r8d, dword ptr [rdi + 0x60]
0x1463b:    mov     ecx, eax
0x1463d:    sub     ecx, dword ptr [rdi + 0x2c]
0x14640:    lea     edx, [r8 - 2]
0x14644:    cmp     ecx, edx
0x14646:    jl      0x14656
0x14648:    mov     ecx, eax
0x1464a:    sub     ecx, r10d
0x1464d:    cmp     edx, ecx
0x1464f:    jg      0x14656
0x14651:    lea     eax, [r10 + r8 - 2]
0x14656:    mov     dword ptr [rbp + 0x28], eax
0x14659:    mov     rdi, rbp
0x1465c:    xor     eax, eax
0x1465e:    sub     r8d, 1
0x14662:    call    0xf110
0x14667:    mov     dword ptr [rbp + 0x2c], eax
0x1466a:    movzx   eax, byte ptr [rbp + 0x6d]
0x1466e:    cmp     r9d, r8d
0x14671:    jge     0x148ce
0x14677:    test    al, al
0x14679:    je      0x13f50
0x1467f:    mov     esi, r10d
0x14682:    mov     rdi, rbp
0x14685:    call    0x10760
0x1468a:    cmp     byte ptr [rbp + 0x6d], 0
0x1468e:    mov     esi, dword ptr [rbp + 0x28]
0x14691:    je      0x13f50

```

```

0x14697:      mov     rdi, rbp
0x1469a:      call   0x10760
0x1469f:      jmp     0x14146
0x14146:
0x146a4:      mov     r8d, dword ptr [rdi + 0x28]
0x146a8:      test    r8d, r8d
0x146ab:      je       0x13f50
0x146b1:      mov     r9d, dword ptr [rdi + 0x60]
0x146b5:      lea     eax, [r9 - 2]
0x146b9:      cmp     r8d, eax
0x146bc:      jge     0x149ca
0x146c2:      mov     dword ptr [rdi + 0x28], 0
0x146c9:      xor     eax, eax
0x146cb:      call   0xf110
0x146d0:      mov     dword ptr [rbp + 0x2c], eax
0x146d3:      sub     r9d, 1
0x146d7:      movzx   eax, byte ptr [rbp + 0x6d]
0x146db:      cmp     r9d, dword ptr [rbp + 0x68]
0x146df:      jle     0x148ce
0x146e5:      mov     esi, r8d
0x146e8:      test    al, al
0x146ea:      je       0x13f50
0x146f0:      jmp     0x14682
0x14682:
0x146f2:      movzx   edi, byte ptr [rdi + 0x6d]
0x146f6:      mov     ebx, r8d
0x146f9:      xor     r13d, r13d
0x146fc:      xor     r12d, r12d
0x146ff:      test    r8d, r8d
0x14702:      jne     0x14739
0x14704:      jmp     0x148c0
0x148c0:
0x14709:      nop     dword ptr [rax]
0x14710:      test    r12d, r12d
0x14713:      jne     0x14730
0x14715:      mov     r12d, 1
0x1471b:      test    dil, dil
0x1471e:      je       0x14730
0x14720:      mov     rdi, rbp
0x14723:      mov     esi, r9d
0x14726:      call   0x10760
0x1472b:      movzx   edi, byte ptr [rbp + 0x6d]
0x1472f:      nop
0x14730:      sub     ebx, 1
0x14733:      je       0x148a0
0x14739:      mov     eax, dword ptr [rbp + 0x68]
0x1473c:      mov     r9d, dword ptr [rbp + 0x28]
0x14740:      sub     eax, 1

```

```

0x14743:      cmp     r9d, eax
0x14746:      jge     0x148a0
0x1474c:      mov     esi, dword ptr [rbp + 0x2c]
0x1474f:      mov     ecx, dword ptr [rbp + 0x60]
0x14752:      lea     edx, [r9 + 1]
0x14756:      mov     dword ptr [rbp + 0x28], edx
0x14759:      sub     ecx, 2
0x1475c:      sub     edx, esi
0x1475e:      cmp     edx, ecx
0x14760:      jl      0x14710
0x14762:      sub     eax, ecx
0x14764:      add     esi, dword ptr [rbp + 0x88]
0x1476a:      mov     r13d, 1
0x14770:      add     eax, 1
0x14773:      cmp     esi, eax
0x14775:      cmovg   esi, eax
0x14778:      mov     dword ptr [rbp + 0x2c], esi
0x1477b:      jmp     0x14730
0x14730:
0x1477d:      mov     rax, qword ptr [rdi + 0x18]
0x14781:      mov     esi, dword ptr [rdi + 0x28]
0x14784:      cmp     byte ptr [rax], 0x2f
0x14787:      je      0x1497c
0x1478d:      test    esi, esi
0x1478f:      je      0x147d0
0x14791:      movsxd  rax, esi
0x14794:      lea     rax, [rax + rax*4]
0x14798:      shl     rax, 4
0x1479c:      add     rax, qword ptr [rbp + 0x38]
0x147a0:      movzx   edx, byte ptr [rax + 0x49]
0x147a4:      test    dl, dl
0x147a6:      sete    byte ptr [rax + 0x49]
0x147aa:      cmp     dl, 1
0x147ad:      sbb     eax, eax
0x147af:      and     eax, 2
0x147b2:      sub     eax, 1
0x147b5:      add     dword ptr [rbp + 0x70], eax
0x147b8:      cmp     byte ptr [rbp + 0x6d], 0
0x147bc:      je      0x147d0
0x147be:      mov     rdi, rbp
0x147c1:      mov     dword ptr [rsp + 8], r8d
0x147c6:      call    0x10760
0x147cb:      mov     r8d, dword ptr [rsp + 8]
0x147d0:      xor     ecx, ecx
0x147d2:      mov     rdx, r12
0x147d5:      mov     esi, 0x1c
0x147da:      mov     rdi, rbp
0x147dd:      call    0x13e30

```

```

0x147e2:      jmp      0x13f50
0x13f50
0x147e7:      mov      rsi, r12
0x147ea:      xor      eax, eax
0x147ec:      call     0x15400
0x147f1:      jmp      0x13f50
0x13f50
0x147f6:      mov      edx, dword ptr [rdi + 0x68]
0x147f9:      test     edx, edx
0x147fb:      jle      0x14132
0x14801:      sub      edx, 1
0x14804:      mov      rcx, qword ptr [rdi + 0x38]
0x14808:      lea      rdx, [rdx + rdx*4]
0x1480c:      shl      rdx, 4
0x14810:      lea      rax, [rcx + 0x49]
0x14814:      lea      rdx, [rcx + rdx + 0x99]
0x1481c:      nop      dword ptr [rax]
0x14820:      cmp      byte ptr [rax], 0
0x14823:      jne      0x14832
0x14825:      cmp      byte ptr [rax - 1], 0
0x14829:      je       0x14832
0x1482b:      mov      byte ptr [rax], 1
0x1482e:      add      dword ptr [rbp + 0x70], 1
0x14832:      add      rax, 0x50
0x14836:      cmp      rax, rdx
0x14839:      jne      0x14820
0x1483b:      jmp      0x14132
0x14132
0x14840:      test     al, al
0x14842:      jne      0x140f0
0x14848:      mov      byte ptr [r12 + 0x49], 1
0x1484e:      add      dword ptr [rbp + 0x70], 1
0x14852:      jmp      0x140f0
0x140f0
0x14857:      lea      rdi, [rip + 0xcb3a]
0x1485e:      call     0x8b00
0x14863:      jmp      0x13f50
0x13f50
0x14868:      nop      dword ptr [rax + rax]
0x14870:      test     r13d, r13d
0x14873:      je       0x14931
0x14879:      test     dl, dl
0x1487b:      je       0x13f50
0x14881:      mov      rdi, rbp
0x14884:      xor      eax, eax
0x14886:      call     0x10d70
0x1488b:      movzx    edx, byte ptr [rbp + 0x6d]
0x1488f:      test     dl, dl

```

```

0x14891:      jne      0x14150
0x14897:      jmp      0x13f50
0x13f50
0x1489c:      nop      dword ptr [rax]
0x148a0:      test     r13d, r13d
0x148a3:      je       0x14956
0x148a9:      test     dil, dil
0x148ac:      je       0x13f50
0x148b2:      mov      rdi, rbp
0x148b5:      xor      eax, eax
0x148b7:      call     0x10d70
0x148bc:      movzx     edi, byte ptr [rbp + 0x6d]
0x148c0:      test     dil, dil
0x148c3:      jne      0x14150
0x148c9:      jmp      0x13f50
0x13f50
0x148ce:      test     al, al
0x148d0:      je       0x13f50
0x148d6:      jmp      0x1413c
0x1413c
0x148db:      mov      rdi, qword ptr [rbp + 0x48]
0x148df:      lea      r14, [rsp + 0x20]
0x148e4:      mov      rsi, r14
0x148e7:      call     0x191a0
0x148ec:      mov      edx, dword ptr [rsp + 0x20]
0x148f0:      mov      rax, qword ptr [rbp + 0x40]
0x148f4:      cmp      rax, qword ptr [rsp + 0x10]
0x148f9:      jne      0x1451c
0x148ff:      mov      r8d, dword ptr [rbp + 0x28]
0x14903:      cmp      r8d, edx
0x14906:      jl       0x14cb2
0x1490c:      jg       0x14ccf
0x14912:      mov      rdi, qword ptr [rbp + 0x48]
0x14916:      mov      edx, 1
0x1491b:      mov      rsi, r14
0x1491e:      call     0x19200
0x14923:      mov      rax, qword ptr [rsp + 0x28]
0x14928:      mov      qword ptr [rbp + 0x40], rax
0x1492c:      jmp      0x1451c
0x1451c
0x14931:      test     r12d, r12d
0x14934:      je       0x1488f
0x1493a:      mov      esi, dword ptr [rbp + 0x28]
0x1493d:      test     dl, dl
0x1493f:      je       0x13f50
0x14945:      mov      rdi, rbp
0x14948:      call     0x10760
0x1494d:      movzx     edx, byte ptr [rbp + 0x6d]

```

```

0x14951:      jmp      0x1488f
0x1488f
0x14956:      test     r12d, r12d
0x14959:      je       0x148c0
0x1495f:      mov      esi, dword ptr [rbp + 0x28]
0x14962:      test     dil, dil
0x14965:      je       0x13f50
0x1496b:      mov      rdi, rbp
0x1496e:      call     0x10760
0x14973:      movzx    edi, byte ptr [rbp + 0x6d]
0x14977:      jmp      0x148c0
0x148c0
0x1497c:      cmp      byte ptr [rax + 1], 0
0x14980:      je       0x14791
0x14986:      jmp      0x1478d
0x1478d
0x1498b:      movsxd   rsi, eax
0x1498e:      lea      rcx, [rip - 0x6315]
0x14995:      mov      edx, 0x50
0x1499a:      call     0x48e0
0x1499f:      jmp      0x13ed7
0x13ed7
0x149a4:      cmp      byte ptr [rax + 1], 0
0x149a8:      je       0x13f50
0x149ae:      jmp      0x1416c
0x1416c
0x149b3:      test     al, al
0x149b5:      jne      0x14048
0x149bb:      mov      byte ptr [r12 + 0x49], 1
0x149c1:      add      dword ptr [rbp + 0x70], 1
0x149c5:      jmp      0x14048
0x14048
0x149ca:      mov      edx, r8d
0x149cd:      sub      edx, eax
0x149cf:      xor      eax, eax
0x149d1:      mov      dword ptr [rdi + 0x28], edx
0x149d4:      call     0xf110
0x149d9:      jmp      0x146d0
0x146d0
0x149de:      mov      eax, dword ptr [rbp + 0x28]
0x149e1:      mov      rdi, qword ptr [rbp + 0x48]
0x149e5:      lea      rsi, [rsp + 0x20]
0x149ea:      mov      qword ptr [rsp + 0x28], r14
0x149ef:      mov      dword ptr [rsp + 0x20], eax
0x149f3:      call     0x19150
0x149f8:      mov      rax, qword ptr [r13 + 8]
0x149fc:      mov      ebx, dword ptr [rbp + 0x28]
0x149ff:      mov      qword ptr [rsp + 8], rax

```



```

0x14a04:    test    ebx, ebx
0x14a06:    js      0x14a80
0x14a08:    movsxd  rax, ebx
0x14a0b:    lea     r8, [rax + rax*4]
0x14a0f:    shl     r8, 4
0x14a13:    add     r8, qword ptr [rbp + 0x38]
0x14a17:    mov     r15, r8
0x14a1a:    jmp     0x14a2c
0x14a2c:
0x14a1c:    nop     dword ptr [rax]
0x14a20:    sub     ebx, 1
0x14a23:    sub     r15, 0x50
0x14a27:    cmp     ebx, -1
0x14a2a:    je      0x14a80
0x14a2c:    mov     rsi, qword ptr [r15]
0x14a2f:    mov     rdi, qword ptr [rsp + 8]
0x14a34:    mov     rdx, r14
0x14a37:    call    0x4c30
0x14a3c:    test    eax, eax
0x14a3e:    jne     0x14a20
0x14a40:    cmp     byte ptr [rbp + 0x6e], 0
0x14a44:    mov     qword ptr [rbp + 0x40], r14
0x14a48:    je      0x14c8d
0x14a4e:    mov     dword ptr [rbp + 0x28], ebx
0x14a51:    xor     eax, eax
0x14a53:    mov     rdi, rbp
0x14a56:    call    0xf110
0x14a5b:    cmp     byte ptr [rbp + 0x6d], 0
0x14a5f:    mov     dword ptr [rbp + 0x2c], eax
0x14a62:    je      0x14a77
0x14a64:    xor     eax, eax
0x14a66:    mov     rdi, rbp
0x14a69:    call    0x10d70
0x14a6e:    mov     rax, qword ptr [rbp + 0x40]
0x14a72:    mov     qword ptr [rsp + 0x10], rax
0x14a77:    mov     byte ptr [rbp + 0x6e], 0
0x14a7b:    jmp     0x14568
0x14568:
0x14a80:    mov     rax, qword ptr [rbp + 0x40]
0x14a84:    mov     dword ptr [r13], 0xffffffff
0x14a8c:    mov     qword ptr [rsp + 0x10], rax
0x14a91:    jmp     0x14568
0x14568:
0x14a96:    cmp     byte ptr [rbp + 0x6e], 0
0x14a9a:    je      0x14c0e
0x14aa0:    mov     eax, dword ptr [rbp + 0x68]
0x14aa3:    lea     ebx, [rax - 1]
0x14aa6:    test    ebx, ebx

```

```

0x14aa8:      js      0x14a80
0x14aaa:      movsxd  rax, ebx
0x14aad:      lea     r15, [rax + rax*4]
0x14ab1:      shl     r15, 4
0x14ab5:      add     r15, qword ptr [rbp + 0x38]
0x14ab9:      jmp     0x14ad1
0x14ad1:
0x14abb:      nop     dword ptr [rax + rax]
0x14ac0:      sub     ebx, 1
0x14ac3:      sub     r15, 0x50
0x14ac7:      mov     rdi, qword ptr [rsp + 8]
0x14acc:      cmp     ebx, -1
0x14acf:      je      0x14a80
0x14ad1:      mov     rsi, qword ptr [r15]
0x14ad4:      mov     rdx, r14
0x14ad7:      mov     qword ptr [rsp + 8], rdi
0x14adc:      call    0x4c30
0x14ae1:      test    eax, eax
0x14ae3:      jne     0x14ac0
0x14ae5:      jmp     0x14a40
0x14a40:
0x14aea:      mov     eax, dword ptr [rbp + 0x28]
0x14aed:      mov     rdi, qword ptr [rbp + 0x48]
0x14af1:      lea     rsi, [rsp + 0x20]
0x14af6:      mov     qword ptr [rsp + 0x28], r14
0x14afb:      mov     dword ptr [rsp + 0x20], eax
0x14aff:      call    0x19150
0x14b04:      mov     rax, qword ptr [r13 + 8]
0x14b08:      mov     ebx, dword ptr [rbp + 0x28]
0x14b0b:      mov     qword ptr [rsp + 8], rax
0x14b10:      mov     eax, dword ptr [rbp + 0x68]
0x14b13:      mov     dword ptr [rsp + 0x1c], eax
0x14b17:      cmp     ebx, eax
0x14b19:      jge     0x14ba3
0x14b1f:      movsxd  rax, ebx
0x14b22:      lea     rcx, [rax + rax*4]
0x14b26:      shl     rcx, 4
0x14b2a:      add     rcx, qword ptr [rbp + 0x38]
0x14b2e:      mov     r15, rcx
0x14b31:      jmp     0x14b45
0x14b45:
0x14b33:      nop     dword ptr [rax + rax]
0x14b38:      add     ebx, 1
0x14b3b:      add     r15, 0x50
0x14b3f:      cmp     ebx, dword ptr [rsp + 0x1c]
0x14b43:      je      0x14ba3
0x14b45:      mov     rsi, qword ptr [r15]
0x14b48:      mov     rdi, qword ptr [rsp + 8]

```

```

0x14b4d:    mov     rdx, r14
0x14b50:    call   0x4c30
0x14b55:    test   eax, eax
0x14b57:    jne     0x14b38
0x14b59:    cmp     ebx, -1
0x14b5c:    je      0x14ba3
0x14b5e:    cmp     byte ptr [rbp + 0x6e], 0
0x14b62:    mov     qword ptr [rbp + 0x40], r14
0x14b66:    je      0x14c70
0x14b6c:    mov     dword ptr [rbp + 0x28], ebx
0x14b6f:    xor     eax, eax
0x14b71:    mov     rdi, rbp
0x14b74:    call   0xf110
0x14b79:    cmp     byte ptr [rbp + 0x6d], 0
0x14b7d:    mov     dword ptr [rbp + 0x2c], eax
0x14b80:    je      0x14b95
0x14b82:    xor     eax, eax
0x14b84:    mov     rdi, rbp
0x14b87:    call   0x10d70
0x14b8c:    mov     rax, qword ptr [rbp + 0x40]
0x14b90:    mov     qword ptr [rsp + 0x10], rax
0x14b95:    mov     byte ptr [rbp + 0x6e], 0
0x14b99:    mov     rax, qword ptr [rsp + 0x10]
0x14b9e:    jmp     0x1451c
0x1451c:
0x14ba3:    mov     dword ptr [r13], 0xffffffff
0x14bab:    mov     rax, qword ptr [rbp + 0x40]
0x14baf:    jmp     0x1451c
0x1451c:
0x14bb4:    xor     ebx, ebx
0x14bb6:    cmp     byte ptr [rbp + 0x6e], 0
0x14bba:    je      0x14c41
0x14bc0:    mov     eax, dword ptr [rbp + 0x68]
0x14bc3:    mov     dword ptr [rsp + 0x1c], eax
0x14bc7:    cmp     ebx, eax
0x14bc9:    jge     0x14ba3
0x14bcb:    movsxd  rax, ebx
0x14bce:    lea     r15, [rax + rax*4]
0x14bd2:    shl     r15, 4
0x14bd6:    add     r15, qword ptr [rbp + 0x38]
0x14bda:    jmp     0x14bf2
0x14bf2:
0x14bdc:    nop     dword ptr [rax]
0x14be0:    add     ebx, 1
0x14be3:    add     r15, 0x50
0x14be7:    cmp     ebx, dword ptr [rsp + 0x1c]
0x14beb:    mov     rcx, qword ptr [rsp + 8]
0x14bf0:    je      0x14ba3

```

```

0x14bf2:    mov     rsi, qword ptr [r15]
0x14bf5:    mov     rdx, r14
0x14bf8:    mov     rdi, rcx
0x14bfb:    mov     qword ptr [rsp + 8], rcx
0x14c00:    call    0x4c30
0x14c05:    test    eax, eax
0x14c07:    jne     0x14be0
0x14c09:    jmp     0x14b59
0x14b59
0x14c0e:    mov     eax, dword ptr [rbp + 0x28]
0x14c11:    mov     rdi, qword ptr [rbp + 0x48]
0x14c15:    lea     rsi, [rsp + 0x20]
0x14c1a:    mov     qword ptr [rsp + 0x28], r14
0x14c1f:    mov     dword ptr [rsp + 0x20], eax
0x14c23:    call    0x19150
0x14c28:    cmp     byte ptr [rbp + 0x6e], 0
0x14c2c:    jne     0x14cee
0x14c32:    mov     eax, dword ptr [rbp + 0x28]
0x14c35:    mov     rdi, qword ptr [r13 + 8]
0x14c39:    lea     ebx, [rax - 1]
0x14c3c:    jmp     0x14aa6
0x14aa6
0x14c41:    mov     eax, dword ptr [rbp + 0x28]
0x14c44:    mov     rdi, qword ptr [rbp + 0x48]
0x14c48:    lea     rsi, [rsp + 0x20]
0x14c4d:    mov     qword ptr [rsp + 0x28], r14
0x14c52:    mov     dword ptr [rsp + 0x20], eax
0x14c56:    call    0x19150
0x14c5b:    cmp     byte ptr [rbp + 0x6e], 0
0x14c5f:    jne     0x14c67
0x14c61:    mov     eax, dword ptr [rbp + 0x28]
0x14c64:    lea     ebx, [rax + 1]
0x14c67:    mov     rcx, qword ptr [r13 + 8]
0x14c6b:    jmp     0x14bc0
0x14bc0
0x14c70:    sub     ebx, dword ptr [rbp + 0x28]
0x14c73:    xor     ecx, ecx
0x14c75:    mov     rdx, r12
0x14c78:    mov     esi, 0x1c
0x14c7d:    mov     r8d, ebx
0x14c80:    mov     rdi, rbp
0x14c83:    call    0x13e30
0x14c88:    jmp     0x14518
0x14518
0x14c8d:    mov     r8d, dword ptr [rbp + 0x28]
0x14c91:    xor     ecx, ecx
0x14c93:    mov     rdx, r12
0x14c96:    mov     esi, 0x1b

```

```

0x14c9b:    mov     rdi, rbp
0x14c9e:    sub     r8d, ebx
0x14ca1:    call    0x13e30
0x14ca6:    jmp     0x1455f
0x1455f:
0x14cab:    xor     eax, eax
0x14cad:    jmp     0x13f00
0x13f00:
0x14cb2:    sub     edx, r8d
0x14cb5:    xor     ecx, ecx
0x14cb7:    mov     esi, 0x1c
0x14cbc:    mov     rdi, rbp
0x14cbf:    mov     r8d, edx
0x14cc2:    mov     rdx, r12
0x14cc5:    call    0x13e30
0x14cca:    jmp     0x14912
0x14912:
0x14ccf:    sub     r8d, edx
0x14cd2:    xor     ecx, ecx
0x14cd4:    mov     rdx, r12
0x14cd7:    mov     esi, 0x1b
0x14cdc:    mov     rdi, rbp
0x14cdf:    call    0x13e30
0x14ce4:    jmp     0x14912
0x14912:
0x14ce9:    call    0x49d0
0x14cee:    mov     rdi, qword ptr [r13 + 8]
0x14cf2:    jmp     0x14aa0
0x14aa0:
0x14cf7:    nop     word ptr [rax + rax]
0x14d00:    endbr64
0x14d04:    push    rbp
0x14d05:    xor     eax, eax
0x14d07:    mov     rbp, rdi
0x14d0a:    mov     dword ptr [rdi + 0x28], 0
0x14d11:    call    0xf110
0x14d16:    mov     rsi, qword ptr [rbp + 0x18]
0x14d1a:    xor     edx, edx
0x14d1c:    mov     ecx, 0x1e
0x14d21:    mov     dword ptr [rbp + 0x2c], eax
0x14d24:    lea     rdi, [rip + 0xc677]
0x14d2b:    xor     eax, eax
0x14d2d:    call    0xf760
0x14d32:    mov     rax, qword ptr [rbp + 0x18]
0x14d36:    cmp     byte ptr [rax], 0x2f
0x14d39:    jne     0x14d51
0x14d3b:    cmp     byte ptr [rax + 1], 0
0x14d3f:    jne     0x14d51

```

```

0x14d41:    lea     rdi, [rip + 0xc672]
0x14d48:    call    0x8b00
0x14d4d:    mov     rax, qword ptr [rbp + 0x18]
0x14d51:    mov     edx, 0x2f
0x14d56:    mov     word ptr [rax], dx
0x14d59:    mov     rdi, qword ptr [rbp + 0x18]
0x14d5d:    mov     qword ptr [rbp + 0x78], 1
0x14d65:    call    0x49c0
0x14d6a:    mov     rdi, rbp
0x14d6d:    mov     r8d, 1
0x14d73:    xor     ecx, ecx
0x14d75:    xor     edx, edx
0x14d77:    mov     esi, 0x22
0x14d7c:    pop     rbp
0x14d7d:    jmp     0x13e30
0x13e30:
0x14d82:    nop     word ptr cs:[rax + rax]
0x14d8d:    nop     dword ptr [rax]
0x14d90:    endbr64
0x14d94:    push    rbp
0x14d95:    cmp     byte ptr [rdi + 0x6d], 0
0x14d99:    mov     rbp, rdi
0x14d9c:    mov     eax, esi
0x14d9e:    mov     byte ptr [rdi + 0x6c], sil
0x14da2:    je      0x14db0
0x14da4:    mov     esi, dword ptr [rdi + 0x28]
0x14da7:    call    0x10760
0x14dac:    movzx   eax, byte ptr [rbp + 0x6c]
0x14db0:    test    al, al
0x14db2:    jne     0x14dc0
0x14db4:    pop     rbp
0x14db5:    ret
0x14db6:    nop     word ptr cs:[rax + rax]
0x14dc0:    mov     rdi, qword ptr [rbp + 0x18]
0x14dc4:    call    0x49c0
0x14dc9:    cmp     eax, -1
0x14dcc:    jne     0x14db4
0x14dce:    xor     eax, eax
0x14dd0:    mov     rdi, rbp
0x14dd3:    call    0x14d00
0x14dd8:    xor     eax, eax
0x14dda:    mov     rdi, rbp
0x14ddd:    call    0x10ec0
0x14de2:    xor     eax, eax
0x14de4:    mov     rdi, rbp
0x14de7:    call    0x10160
0x14dec:    cmp     byte ptr [rbp + 0x6d], 0
0x14df0:    jne     0x14e10

```

```

0x14df2:    xor     eax, eax
0x14df4:    mov     rdi, rbp
0x14df7:    call    0x10240
0x14dfc:    cmp     byte ptr [rbp + 0x6d], 0
0x14e00:    je      0x14db4
0x14e02:    mov     rdi, rbp
0x14e05:    xor     eax, eax
0x14e07:    pop     rbp
0x14e08:    jmp     0x10d70
0x10d70
0x14e0d:    nop     dword ptr [rax]
0x14e10:    mov     rdi, rbp
0x14e13:    xor     eax, eax
0x14e15:    call    0xec50
0x14e1a:    jmp     0x14df2
0x14df2
0x14e1c:    nop     dword ptr [rax]
0x14e20:    endbr64
0x14e24:    push    r15
0x14e26:    mov     r15, rsi
0x14e29:    push    r14
0x14e2b:    push    r13
0x14e2d:    push    r12
0x14e2f:    push    rbp
0x14e30:    mov     rbp, rdi
0x14e33:    push    rbx
0x14e34:    sub     rsp, 0x58
0x14e38:    mov     edx, dword ptr [rdi + 0x70]
0x14e3b:    mov     rax, qword ptr fs:[0x28]
0x14e44:    mov     qword ptr [rsp + 0x48], rax
0x14e49:    xor     eax, eax
0x14e4b:    mov     qword ptr [rsp + 0x38], 0
0x14e54:    mov     qword ptr [rsp + 0x40], 0
0x14e5d:    mov     dword ptr [rdi + 0x90], 1
0x14e67:    test    edx, edx
0x14e69:    jne     0x15138
0x14e6f:    movsxd  rax, dword ptr [rdi + 0x28]
0x14e73:    mov     rdx, rax
0x14e76:    lea     rax, [rax + rax*4]
0x14e7a:    shl     rax, 4
0x14e7e:    add     rax, qword ptr [rdi + 0x38]
0x14e82:    mov     r12, qword ptr [rax]
0x14e85:    test    edx, edx
0x14e87:    jne     0x14ea0
0x14e89:    mov     rax, qword ptr [rdi + 0x18]
0x14e8d:    cmp     byte ptr [rax], 0x2f
0x14e90:    jne     0x15110
0x14e96:    cmp     byte ptr [rax + 1], 0

```

```

0x14e9a:    jne     0x15110
0x14ea0:    mov     rdi, r12
0x14ea3:    call    0x49b0
0x14ea8:    lea     rdi, [rax + 0x11]
0x14eac:    call    0x1d670
0x14eb1:    mov     edx, 0xe
0x14eb6:    mov     rsi, r12
0x14eb9:    lea     rdi, [rip + 0x16720]
0x14ec0:    mov     r13, rax
0x14ec3:    call    0x4880
0x14ec8:    lea     rcx, [rip + 0xc500]
0x14ecf:    mov     esi, 1
0x14ed4:    mov     rdi, r13
0x14ed7:    mov     r8, rax
0x14eda:    mov     rdx, -1
0x14ee1:    xor     eax, eax
0x14ee3:    call    0x4f10
0x14ee8:    mov     rdi, r12
0x14eeb:    call    0x49b0
0x14ef0:    mov     rdi, qword ptr [r15 + 0x18]
0x14ef4:    mov     rbx, rax
0x14ef7:    call    0x49b0
0x14efc:    lea     rdi, [rbx + rax + 2]
0x14f01:    call    0x1d670
0x14f06:    mov     r8, qword ptr [r15 + 0x18]
0x14f0a:    mov     r9, r12
0x14f0d:    lea     rcx, [rip + 0xb7ed]
0x14f14:    mov     rdi, rax
0x14f17:    mov     r14, rax
0x14f1a:    mov     esi, 1
0x14f1f:    xor     eax, eax
0x14f21:    mov     rdx, -1
0x14f28:    call    0x4f10
0x14f2d:    mov     rdi, r13
0x14f30:    lea     rsi, [rsp + 0x40]
0x14f35:    mov     rdx, r14
0x14f38:    mov     rcx, qword ptr [rip + 0x166d9]
0x14f3f:    call    0x83e0
0x14f44:    mov     rdi, r13
0x14f47:    test    rax, rax
0x14f4a:    je      0x153c3
0x14f50:    call    0x18d10
0x14f55:    movsxd  rax, dword ptr [rbp + 0x28]
0x14f59:    lea     rax, [rax + rax*4]
0x14f5d:    shl     rax, 4
0x14f61:    add     rax, qword ptr [rbp + 0x38]
0x14f65:    mov     eax, dword ptr [rax + 0x10]
0x14f68:    and     eax, 0xf000

```



```

0x14f6d:    cmp     eax, 0x4000
0x14f72:    je      0x152f0
0x14f78:    mov     rdi, qword ptr [rip + 0x15509]
0x14f7f:    call    0xdb60
0x14f84:    xor     eax, eax
0x14f86:    call    0xa820
0x14f8b:    mov     rdi, qword ptr [rsp + 0x40]
0x14f90:    xor     eax, eax
0x14f92:    call    0x19940
0x14f97:    mov     rdi, qword ptr [rsp + 0x40]
0x14f9c:    mov     r13, rax
0x14f9f:    call    0x18d10
0x14fa4:    mov     rdi, r12
0x14fa7:    xor     eax, eax
0x14fa9:    mov     rsi, r13
0x14fac:    mov     qword ptr [rsp + 0x40], r13
0x14fb1:    call    0x11190
0x14fb6:    mov     rdi, r14
0x14fb9:    mov     ebx, eax
0x14fbb:    call    0x18d10
0x14fc0:    test    ebx, ebx
0x14fc2:    jne     0x153a0
0x14fc8:    call    0x4bf0
0x14fcd:    mov     ebx, eax
0x14fcf:    call    0x49e0
0x14fd4:    movsxd  rdx, dword ptr [rbp + 0x28]
0x14fd8:    mov     r9d, ebx
0x14fdb:    mov     rsi, r12
0x14fde:    mov     r8d, eax
0x14fe1:    mov     rdi, rbp
0x14fe4:    xor     eax, eax
0x14fe6:    lea     rdx, [rdx + rdx*4]
0x14fea:    shl     rdx, 4
0x14fee:    add     rdx, qword ptr [rbp + 0x38]
0x14ff2:    mov     ecx, dword ptr [rdx + 0x10]
0x14ff5:    mov     rdx, qword ptr [rsp + 0x40]
0x14ffa:    call    0x11330
0x14fff:    mov     rdi, qword ptr [rsp + 0x40]
0x15004:    mov     ebx, eax
0x15006:    call    0x18d10
0x1500b:    lea     eax, [rbx - 1]
0x1500e:    cmp     eax, 1
0x15011:    jbe     0x15039
0x15013:    cdqe
0x15015:    lea     rdx, [rip + 0x15404]
0x1501c:    xor     ecx, ecx
0x1501e:    mov     rsi, r12
0x15021:    mov     rdx, qword ptr [rdx + rax*8]

```

```

0x15025:    mov     r8d, 0x1e
0x1502b:    lea     rdi, [rip + 0xc277]
0x15032:    xor     eax, eax
0x15034:    call    0xf7e0
0x15039:    xor     eax, eax
0x1503b:    call    0xe4e0
0x15040:    mov     rdi, rbp
0x15043:    xor     eax, eax
0x15045:    call    0x10240
0x1504a:    xor     eax, eax
0x1504c:    mov     rdi, r15
0x1504f:    call    0x10240
0x15054:    xor     eax, eax
0x15056:    call    0xa820
0x1505b:    mov     rsi, qword ptr [r15 + 0x18]
0x1505f:    xor     eax, eax
0x15061:    mov     edx, 1
0x15066:    mov     rdi, r15
0x15069:    call    0xf880
0x1506e:    test    eax, eax
0x15070:    je      0x153e0
0x15076:    xor     eax, eax
0x15078:    mov     rdi, r15
0x1507b:    call    0x10ec0
0x15080:    xor     eax, eax
0x15082:    mov     rdi, r15
0x15085:    call    0x10160
0x1508a:    cmp     byte ptr [r15 + 0x6d], 0
0x1508f:    je      0x1509b
0x15091:    mov     rdi, r15
0x15094:    xor     eax, eax
0x15096:    call    0xec50
0x1509b:    xor     eax, eax
0x1509d:    mov     rdi, r15
0x150a0:    call    0x10240
0x150a5:    cmp     byte ptr [r15 + 0x6d], 0
0x150aa:    je      0x150b6
0x150ac:    mov     rdi, r15
0x150af:    xor     eax, eax
0x150b1:    call    0x10d70
0x150b6:    mov     rsi, qword ptr [rbp + 0x18]
0x150ba:    xor     eax, eax
0x150bc:    mov     edx, 1
0x150c1:    mov     rdi, rbp
0x150c4:    call    0xf880
0x150c9:    test    eax, eax
0x150cb:    je      0x153d0
0x150d1:    xor     eax, eax

```

```

0x150d3:    mov     rdi, rbp
0x150d6:    call   0x10ec0
0x150db:    xor     eax, eax
0x150dd:    mov     rdi, rbp
0x150e0:    call   0x10160
0x150e5:    cmp     byte ptr [rbp + 0x6d], 0
0x150e9:    je      0x150f5
0x150eb:    mov     rdi, rbp
0x150ee:    xor     eax, eax
0x150f0:    call   0xec50
0x150f5:    xor     eax, eax
0x150f7:    mov     rdi, rbp
0x150fa:    call   0x10240
0x150ff:    cmp     byte ptr [rbp + 0x6d], 0
0x15103:    je      0x15110
0x15105:    mov     rdi, rbp
0x15108:    xor     eax, eax
0x1510a:    call   0x10d70
0x1510f:    nop
0x15110:    mov     rax, qword ptr [rsp + 0x48]
0x15115:    sub     rax, qword ptr fs:[0x28]
0x1511e:    jne     0x153ed
0x15124:    add     rsp, 0x58
0x15128:    pop     rbx
0x15129:    pop     rbp
0x1512a:    pop     r12
0x1512c:    pop     r13
0x1512e:    pop     r14
0x15130:    pop     r15
0x15132:    ret
0x15133:    nop     dword ptr [rax + rax]
0x15138:    mov     rdx, qword ptr [rsi + 0x18]
0x1513c:    mov     rcx, qword ptr [rip + 0x164d5]
0x15143:    lea     rsi, [rsp + 0x38]
0x15148:    lea     rdi, [rip + 0xc28d]
0x1514f:    call   0x83e0
0x15154:    test    rax, rax
0x15157:    je      0x15110
0x15159:    mov     rdi, qword ptr [rbp + 0x18]
0x1515d:    mov     rsi, qword ptr [rsp + 0x38]
0x15162:    xor     eax, eax
0x15164:    call   0x11190
0x15169:    test    eax, eax
0x1516b:    jne     0x15308
0x15171:    mov     r12, qword ptr [rsp + 0x38]
0x15176:    mov     rdi, r12
0x15179:    call   0x49b0
0x1517e:    mov     rdi, r12

```

```

0x15181:    mov     rbx, rax
0x15184:    lea     rax, [rax + 1]
0x15188:    mov     qword ptr [rsp + 0x18], rax
0x1518d:    lea     rax, [rbx + 2]
0x15191:    mov     rsi, rax
0x15194:    mov     qword ptr [rsp + 0x10], rax
0x15199:    call    0x1d690
0x1519e:    mov     qword ptr [rsp + 0x38], rax
0x151a3:    mov     byte ptr [rax + rbx], 0x2f
0x151a7:    mov     rax, qword ptr [rsp + 0x38]
0x151ac:    mov     byte ptr [rax + rbx + 1], 0
0x151b1:    mov     eax, dword ptr [rbp + 0x70]
0x151b4:    mov     dword ptr [rbp + 0x50], 0xffffffff
0x151bb:    mov     dword ptr [rbp + 0x54], eax
0x151be:    mov     rdi, rbp
0x151c1:    call    0x100e0
0x151c6:    cmp     eax, -1
0x151c9:    je      0x152ce
0x151cf:    cdqe
0x151d1:    lea     rax, [rax + rax*4]
0x151d5:    shl     rax, 4
0x151d9:    mov     qword ptr [rsp + 8], rax
0x151de:    add     rax, qword ptr [rbp + 0x38]
0x151e2:    mov     r12, qword ptr [rax]
0x151e5:    mov     r14d, dword ptr [rax + 0x10]
0x151e9:    call    0x49e0
0x151ee:    mov     ebx, eax
0x151f0:    call    0x4bf0
0x151f5:    mov     rdi, r12
0x151f8:    mov     r13d, eax
0x151fb:    call    0x49b0
0x15200:    mov     rcx, qword ptr [rsp + 0x10]
0x15205:    mov     rdi, qword ptr [rsp + 0x38]
0x1520a:    lea     rsi, [rcx + rax]
0x1520e:    call    0x1d690
0x15213:    mov     rcx, qword ptr [rsp + 0x18]
0x15218:    mov     rsi, r12
0x1521b:    mov     qword ptr [rsp + 0x38], rax
0x15220:    lea     rdi, [rax + rcx]
0x15224:    call    0x48b0
0x15229:    mov     eax, dword ptr [rip + 0x16881]
0x1522f:    test    eax, eax
0x15231:    jne     0x15338
0x15237:    mov     rdi, qword ptr [rip + 0x15252]
0x1523e:    call    0xdb60
0x15243:    xor     eax, eax
0x15245:    call    0xa820
0x1524a:    mov     rdx, qword ptr [rsp + 0x38]

```

```

0x1524f:    xor     eax, eax
0x15251:    mov     r9d, r13d
0x15254:    mov     r8d, ebx
0x15257:    mov     ecx, r14d
0x1525a:    mov     rsi, r12
0x1525d:    mov     rdi, rbp
0x15260:    call    0x11330
0x15265:    cmp     eax, 1
0x15268:    je      0x15320
0x1526e:    cmp     eax, 2
0x15271:    je      0x152ce
0x15273:    cmp     eax, 3
0x15276:    je      0x151be
0x1527c:    sub     eax, 1
0x1527f:    lea     rbx, [rip + 0x1519a]
0x15286:    xor     ecx, ecx
0x15288:    mov     rsi, r12
0x1528b:    cdqe
0x1528d:    mov     r8d, 0xe
0x15293:    lea     rdi, [rip + 0xc00f]
0x1529a:    mov     rdx, qword ptr [rbx + rax*8]
0x1529e:    xor     eax, eax
0x152a0:    call    0xf7e0
0x152a5:    test    al, al
0x152a7:    je      0x152ce
0x152a9:    xor     eax, eax
0x152ab:    mov     rdi, rbp
0x152ae:    call    0x10240
0x152b3:    mov     rdi, r15
0x152b6:    xor     eax, eax
0x152b8:    call    0x10240
0x152bd:    mov     rdi, rbp
0x152c0:    call    0x100e0
0x152c5:    cmp     eax, -1
0x152c8:    jne     0x151cf
0x152ce:    mov     rdi, qword ptr [rsp + 0x38]
0x152d3:    test    rdi, rdi
0x152d6:    je      0x152dd
0x152d8:    call    0x18d10
0x152dd:    xor     eax, eax
0x152df:    call    0xe4e0
0x152e4:    xor     eax, eax
0x152e6:    call    0xa820
0x152eb:    jmp     0x1505b
0x1505b:
0x152f0:    mov     rdi, qword ptr [rip + 0x151a1]
0x152f7:    call    0xdb60
0x152fc:    jmp     0x14f84

```

```

0x14f84
0x15301:      nop      dword ptr [rax]
0x15308:      mov      edx, 0xc
0x1530d:      xor      esi, esi
0x1530f:      lea      rdi, [rip + 0x14fea]
0x15316:      call     0x8070
0x1531b:      jmp      0x15110
0x15110
0x15320:      mov      rax, qword ptr [rbp + 0x38]
0x15324:      mov      rbx, qword ptr [rsp + 8]
0x15329:      mov      byte ptr [rax + rbx + 0x49], 0
0x1532e:      jmp      0x152a9
0x152a9
0x15333:      nop      dword ptr [rax + rax]
0x15338:      mov      dword ptr [rip + 0x1676e], 0
0x15342:      xor      eax, eax
0x15344:      call     0xc8c0
0x15349:      mov      edx, 4
0x1534e:      lea      rsi, [rip + 0xb1dd]
0x15355:      lea      rdi, [rip + 0xbe9d]
0x1535c:      mov      qword ptr [rsp + 0x28], rax
0x15361:      call     0x8070
0x15366:      mov      rdi, qword ptr [rsp + 0x28]
0x1536b:      mov      byte ptr [rsp + 0x27], al
0x1536f:      call     0xc970
0x15374:      xor      eax, eax
0x15376:      call     0xd870
0x1537b:      xor      eax, eax
0x1537d:      call     0xd740
0x15382:      xor      eax, eax
0x15384:      call     0xa820
0x15389:      movzx    eax, byte ptr [rsp + 0x27]
0x1538e:      and      eax, 0xfffffffff
0x15391:      cmp      al, 0x4e
0x15393:      je       0x15237
0x15399:      jmp      0x152ce
0x152ce
0x1539e:      nop
0x153a0:      mov      rdx, qword ptr [rsp + 0x40]
0x153a5:      lea      rdi, [rip + 0xc8a4]
0x153ac:      xor      ecx, ecx
0x153ae:      xor      eax, eax
0x153b0:      mov      r8d, 0x1e
0x153b6:      mov      rsi, r12
0x153b9:      call     0xf7e0
0x153be:      mov      rdi, qword ptr [rsp + 0x40]
0x153c3:      call     0x18d10
0x153c8:      jmp      0x15110

```

```

0x15110
0x153cd:      nop      dword ptr [rax]
0x153d0:      mov      rdi, rbp
0x153d3:      call     0x14d00
0x153d8:      jmp      0x150d1
0x150d1
0x153dd:      nop      dword ptr [rax]
0x153e0:      mov      rdi, r15
0x153e3:      call     0x14d00
0x153e8:      jmp      0x15076
0x15076
0x153ed:      call     0x49d0
0x153f2:      nop      word ptr cs:[rax + rax]
0x153fd:      nop      dword ptr [rax]
0x15400:      endbr64
0x15404:      push     r15
0x15406:      push     r14
0x15408:      push     r13
0x1540a:      push     r12
0x1540c:      push     rbp
0x1540d:      mov      rbp, rdi
0x15410:      push     rbx
0x15411:      sub      rsp, 0x28
0x15415:      mov      qword ptr [rsp + 0x10], rsi
0x1541a:      mov      esi, dword ptr [rdi + 0x70]
0x1541d:      test     esi, esi
0x1541f:      jne      0x15450
0x15421:      mov      ecx, dword ptr [rdi + 0x28]
0x15424:      test     ecx, ecx
0x15426:      jne      0x15450
0x15428:      mov      rax, qword ptr [rdi + 0x18]
0x1542c:      cmp      byte ptr [rax], 0x2f
0x1542f:      je       0x15440
0x15431:      add      rsp, 0x28
0x15435:      pop      rbx
0x15436:      pop      rbp
0x15437:      pop      r12
0x15439:      pop      r13
0x1543b:      pop      r14
0x1543d:      pop      r15
0x1543f:      ret
0x15440:      cmp      byte ptr [rax + 1], 0
0x15444:      jne      0x15431
0x15446:      nop      word ptr cs:[rax + rax]
0x15450:      xor      edx, edx
0x15452:      lea      rsi, [rip + 0xb0d9]
0x15459:      lea      rdi, [rip + 0xbf97]
0x15460:      call     0x8070

```

```

0x15465:      cmp     al, 0x79
0x15467:      jne     0x15431
0x15469:      mov     edx, dword ptr [rbp + 0x68]
0x1546c:      test    edx, edx
0x1546e:      jle     0x158fc
0x15474:      mov     rax, qword ptr [rbp + 0x38]
0x15478:      xor     ecx, ecx
0x1547a:      add     rax, 0x49
0x1547e:      jmp     0x1548b
0x1548b:
0x15480:      add     ecx, 1
0x15483:      add     rax, 0x50
0x15487:      cmp     ecx, edx
0x15489:      je      0x15490
0x1548b:      cmp     byte ptr [rax], 0
0x1548e:      je      0x15480
0x15490:      mov     dword ptr [rsp + 0x1c], ecx
0x15494:      mov     eax, dword ptr [rbp + 0x70]
0x15497:      mov     dword ptr [rbp + 0x50], 0xffffffff
0x1549e:      xor     r13d, r13d
0x154a1:      lea     r15, [rip + 0xbf69]
0x154a8:      mov     dword ptr [rsp + 0x18], 1
0x154b0:      mov     dword ptr [rbp + 0x54], eax
0x154b3:      nop     dword ptr [rax + rax]
0x154b8:      mov     rdi, rbp
0x154bb:      call    0x100e0
0x154c0:      cmp     eax, -1
0x154c3:      je      0x155c6
0x154c9:      cdqe
0x154cb:      lea     rbx, [rax + rax*4]
0x154cf:      mov     rax, qword ptr [rbp + 0x38]
0x154d3:      shl     rbx, 4
0x154d7:      mov     r12, qword ptr [rax + rbx]
0x154db:      xor     eax, eax
0x154dd:      call    0x18ab0
0x154e2:      mov     rdi, r12
0x154e5:      call    0x49b0
0x154ea:      lea     rdi, [rax + 0x20]
0x154ee:      call    0x1d670
0x154f3:      mov     r8, r12
0x154f6:      mov     rcx, r15
0x154f9:      mov     esi, 1
0x154fe:      mov     r14, rax
0x15501:      mov     rdi, rax
0x15504:      mov     rdx, -1
0x1550b:      xor     eax, eax
0x1550d:      call    0x4f10
0x15512:      mov     edx, 1

```



```

0x15517:      mov     esi, 2
0x1551c:      mov     rdi, r14
0x1551f:      call    0xe480
0x15524:      xor     eax, eax
0x15526:      call    0xa820
0x1552b:      mov     rdi, r14
0x1552e:      call    0x18d10
0x15533:      mov     edx, dword ptr [rip + 0x16577]
0x15539:      test    edx, edx
0x1553b:      jne     0x15760
0x15541:      mov     eax, dword ptr [rsp + 0x18]
0x15545:      test    eax, eax
0x15547:      jne     0x15738
0x1554d:      mov     rdi, qword ptr [rip + 0x14f54]
0x15554:      call    0xdb60
0x15559:      xor     eax, eax
0x1555b:      call    0xa820
0x15560:      cmp     r13d, 0x61
0x15564:      je      0x15728
0x1556a:      cmp     r13d, 0x73
0x1556e:      je      0x154b8
0x15574:      cmp     r13d, 0x79
0x15578:      jne     0x155c6
0x1557a:      mov     rax, qword ptr [rbp + 0x38]
0x1557e:      cmp     byte ptr [rax + rbx + 0x48], 0
0x15583:      jne     0x156e8
0x15589:      mov     rdi, qword ptr [rip + 0x14f10]
0x15590:      call    0xdb60
0x15595:      xor     eax, eax
0x15597:      call    0xa820
0x1559c:      mov     rdi, r12
0x1559f:      call    0x4990
0x155a4:      test    eax, eax
0x155a6:      jne     0x157c8
0x155ac:      mov     rax, qword ptr [rbp + 0x38]
0x155b0:      mov     rdi, rbp
0x155b3:      mov     byte ptr [rax + rbx + 0x49], 0
0x155b8:      call    0x100e0
0x155bd:      cmp     eax, -1
0x155c0:      jne     0x154c9
0x155c6:      mov     eax, dword ptr [rsp + 0x1c]
0x155ca:      cmp     dword ptr [rbp + 0x68], eax
0x155cd:      je      0x155e3
0x155cf:      mov     eax, dword ptr [rsp + 0x1c]
0x155d3:      mov     rdi, rbp
0x155d6:      mov     dword ptr [rbp + 0x28], eax
0x155d9:      xor     eax, eax
0x155db:      call    0xf110

```

```

0x155e0:    mov     dword ptr [rbp + 0x2c], eax
0x155e3:    xor     eax, eax
0x155e5:    mov     rdi, rbp
0x155e8:    call    0x10240
0x155ed:    mov     rdi, qword ptr [rsp + 0x10]
0x155f2:    xor     eax, eax
0x155f4:    call    0x10240
0x155f9:    xor     eax, eax
0x155fb:    call    0xe4e0
0x15600:    xor     eax, eax
0x15602:    call    0xa820
0x15607:    mov     rsi, qword ptr [rbp + 0x18]
0x1560b:    xor     eax, eax
0x1560d:    mov     edx, 1
0x15612:    mov     rdi, rbp
0x15615:    call    0xf880
0x1561a:    test    eax, eax
0x1561c:    je      0x158e0
0x15622:    xor     eax, eax
0x15624:    mov     rdi, rbp
0x15627:    call    0x10ec0
0x1562c:    xor     eax, eax
0x1562e:    mov     rdi, rbp
0x15631:    call    0x10160
0x15636:    cmp     byte ptr [rbp + 0x6d], 0
0x1563a:    jne     0x15890
0x15640:    xor     eax, eax
0x15642:    mov     rdi, rbp
0x15645:    call    0x10240
0x1564a:    cmp     byte ptr [rbp + 0x6d], 0
0x1564e:    jne     0x15880
0x15654:    mov     rax, qword ptr [rsp + 0x10]
0x15659:    mov     rdi, qword ptr [rbp + 0x18]
0x1565d:    mov     r12, qword ptr [rax + 0x18]
0x15661:    mov     rsi, r12
0x15664:    call    0x4b30
0x15669:    test    eax, eax
0x1566b:    jne     0x15431
0x15671:    mov     rdi, qword ptr [rsp + 0x10]
0x15676:    mov     edx, 1
0x1567b:    mov     rsi, r12
0x1567e:    call    0xf880
0x15683:    test    eax, eax
0x15685:    je      0x158ed
0x1568b:    mov     rbx, qword ptr [rsp + 0x10]
0x15690:    xor     eax, eax
0x15692:    mov     rdi, rbx
0x15695:    call    0x10ec0

```

```

0x1569a:    xor     eax, eax
0x1569c:    mov     rdi, rbx
0x1569f:    call    0x10160
0x156a4:    cmp     byte ptr [rbx + 0x6d], 0
0x156a8:    je      0x156b4
0x156aa:    mov     rdi, rbx
0x156ad:    xor     eax, eax
0x156af:    call    0xec50
0x156b4:    mov     rbx, qword ptr [rsp + 0x10]
0x156b9:    xor     eax, eax
0x156bb:    mov     rdi, rbx
0x156be:    call    0x10240
0x156c3:    cmp     byte ptr [rbx + 0x6d], 0
0x156c7:    je      0x15431
0x156cd:    add     rsp, 0x28
0x156d1:    mov     rdi, rbx
0x156d4:    xor     eax, eax
0x156d6:    pop     rbx
0x156d7:    pop     rbp
0x156d8:    pop     r12
0x156da:    pop     r13
0x156dc:    pop     r14
0x156de:    pop     r15
0x156e0:    jmp     0x10d70
0x10d70
0x156e5:    nop     dword ptr [rax]
0x156e8:    mov     rdi, r12
0x156eb:    call    0x4870
0x156f0:    test    eax, eax
0x156f2:    je      0x155ac
0x156f8:    xor     eax, eax
0x156fa:    mov     ecx, 0xe
0x156ff:    mov     rsi, r12
0x15702:    lea     rdx, [rip + 0xae29]
0x15709:    lea     rdi, [rip + 0xc5e8]
0x15710:    call    0xf760
0x15715:    cmp     al, 0x79
0x15717:    je      0x154b8
0x1571d:    jmp     0x155c6
0x155c6
0x15722:    nop     word ptr [rax + rax]
0x15728:    mov     dword ptr [rsp + 0x18], 0
0x15730:    jmp     0x1557a
0x1557a
0x15735:    nop     dword ptr [rax]
0x15738:    mov     ecx, 2
0x1573d:    lea     rdx, [rip + 0xbcd9]
0x15744:    mov     rsi, r12

```

```

0x15747:      xor     eax, eax
0x15749:      lea     rdi, [rip + 0xc528]
0x15750:      call    0xf760
0x15755:      movsx   r13d, al
0x15759:      jmp     0x1554d
0x1554d:
0x1575e:      nop
0x15760:      mov     dword ptr [rip + 0x16346], 0
0x1576a:      xor     eax, eax
0x1576c:      call    0xc8c0
0x15771:      mov     edx, 4
0x15776:      lea     rsi, [rip + 0xad5]
0x1577d:      lea     rdi, [rip + 0xba75]
0x15784:      mov     qword ptr [rsp + 8], rax
0x15789:      call    0x8070
0x1578e:      mov     rdi, qword ptr [rsp + 8]
0x15793:      mov     r14d, eax
0x15796:      call    0xc970
0x1579b:      xor     eax, eax
0x1579d:      and     r14d, 0xffffffff
0x157a1:      call    0xd870
0x157a6:      xor     eax, eax
0x157a8:      call    0xd740
0x157ad:      xor     eax, eax
0x157af:      call    0xa820
0x157b4:      cmp     r14b, 0x4e
0x157b8:      je      0x15541
0x157be:      jmp     0x155c6
0x155c6:
0x157c3:      nop     dword ptr [rax + rax]
0x157c8:      xor     eax, eax
0x157ca:      mov     ecx, 0x12
0x157cf:      mov     rsi, r12
0x157d2:      lea     rdx, [rip + 0xad59]
0x157d9:      lea     rdi, [rip + 0xc4c0]
0x157e0:      call    0xf760
0x157e5:      cmp     al, 0x79
0x157e7:      jne     0x156f8
0x157ed:      mov     rdi, r12
0x157f0:      call    0x49b0
0x157f5:      lea     rdi, [rax + 0x21]
0x157f9:      call    0x1d670
0x157fe:      mov     rdx, -1
0x15805:      mov     r8, r12
0x15808:      lea     rcx, [rip + 0xbb38]
0x1580f:      mov     r14, rax
0x15812:      mov     rdi, rax
0x15815:      mov     esi, 1

```

```

0x1581a:    xor     eax, eax
0x1581c:    call   0x4f10
0x15821:    mov     esi, 1
0x15826:    mov     rdi, r14
0x15829:    call   0x193f0
0x1582e:    mov     rdi, r14
0x15831:    mov     dword ptr [rsp + 8], eax
0x15835:    call   0x18d10
0x1583a:    mov     rdi, qword ptr [rbp + 0x18]
0x1583e:    call   0xc4b0
0x15843:    mov     eax, dword ptr [rsp + 8]
0x15847:    mov     edx, eax
0x15849:    and     edx, 0x7f
0x1584c:    add     edx, 1
0x1584f:    cmp     dl, 1
0x15852:    jg      0x158b0
0x15854:    test    al, 0x7f
0x15856:    jne     0x156f8
0x1585c:    test    ah, 0xff
0x1585f:    je      0x155ac
0x15865:    lea     rdi, [rip + 0xb37e]
0x1586c:    call   0x195e0
0x15871:    jmp     0x156f8
0x156f8
0x15876:    nop     word ptr cs:[rax + rax]
0x15880:    mov     rdi, rbp
0x15883:    xor     eax, eax
0x15885:    call   0x10d70
0x1588a:    jmp     0x15654
0x15654
0x1588f:    nop
0x15890:    xor     eax, eax
0x15892:    mov     rdi, rbp
0x15895:    call   0xec50
0x1589a:    xor     eax, eax
0x1589c:    mov     rdi, rbp
0x1589f:    call   0x10240
0x158a4:    cmp     byte ptr [rbp + 0x6d], 0
0x158a8:    je      0x15654
0x158ae:    jmp     0x15880
0x15880
0x158b0:    xor     eax, eax
0x158b2:    mov     ecx, 0xe
0x158b7:    mov     rsi, r12
0x158ba:    lea     rdx, [rip + 0xac71]
0x158c1:    lea     rdi, [rip + 0xc408]
0x158c8:    call   0xf760
0x158cd:    cmp     al, 0x79

```

```

0x158cf:      je      0x154b8
0x158d5:      jmp     0x155c6
0x155c6
0x158da:      nop     word ptr [rax + rax]
0x158e0:      mov     rdi, rbp
0x158e3:      call    0x14d00
0x158e8:      jmp     0x15622
0x15622
0x158ed:      mov     rdi, qword ptr [rsp + 0x10]
0x158f2:      call    0x14d00
0x158f7:      jmp     0x1568b
0x1568b
0x158fc:      mov     dword ptr [rsp + 0x1c], 0
0x15904:      jmp     0x15494
0x15494
0x15909:      nop     dword ptr [rax]
0x15910:      endbr64
0x15914:      push    r13
0x15916:      xor     edx, edx
0x15918:      push    r12
0x1591a:      push    rbp
0x1591b:      mov     rbp, rsi
0x1591e:      push    rbx
0x1591f:      mov     rbx, rdi
0x15922:      lea     rdi, [rip + 0xbaf9]
0x15929:      sub     rsp, 0x18
0x1592d:      mov     rcx, qword ptr [rip + 0x15cd4]
0x15934:      mov     rax, qword ptr fs:[0x28]
0x1593d:      mov     qword ptr [rsp + 8], rax
0x15942:      xor     eax, eax
0x15944:      mov     rsi, rsp
0x15947:      mov     qword ptr [rsp], 0
0x1594f:      call    0x83e0
0x15954:      test    rax, rax
0x15957:      je      0x15a0b
0x1595d:      mov     rdi, qword ptr [rsp]
0x15961:      cmp     byte ptr [rdi], 0
0x15964:      je      0x15a06
0x1596a:      xor     eax, eax
0x1596c:      call    0x19940
0x15971:      mov     rdi, qword ptr [rsp]
0x15975:      mov     r13, rax
0x15978:      call    0x18d10
0x1597d:      mov     rdi, r13
0x15980:      mov     qword ptr [rsp], r13
0x15984:      call    0x49b0
0x15989:      mov     rdi, r13
0x1598c:      lea     rsi, [rax + 2]

```

```

0x15990:      mov     r12, rax
0x15993:      call    0x4ce0
0x15998:      mov     esi, 0x41ff
0x1599d:      mov     qword ptr [rsp], rax
0x159a1:      mov     byte ptr [rax + r12], 0x2f
0x159a6:      mov     rax, qword ptr [rsp]
0x159aa:      mov     byte ptr [rax + r12 + 1], 0
0x159b0:      mov     rdi, qword ptr [rsp]
0x159b4:      xor     eax, eax
0x159b6:      call    0x124d0
0x159bb:      cmp     eax, -1
0x159be:      je       0x15a90
0x159c4:      mov     rsi, qword ptr [rbx + 0x18]
0x159c8:      xor     eax, eax
0x159ca:      mov     edx, 1
0x159cf:      mov     rdi, rbx
0x159d2:      call    0xf880
0x159d7:      test    eax, eax
0x159d9:      jne      0x15a30
0x159db:      mov     rdi, rbx
0x159de:      call    0x14d00
0x159e3:      mov     rsi, qword ptr [rbp + 0x18]
0x159e7:      mov     rdi, qword ptr [rbx + 0x18]
0x159eb:      call    0x4b30
0x159f0:      test    eax, eax
0x159f2:      je       0x15af0
0x159f8:      mov     rdi, rbp
0x159fb:      xor     eax, eax
0x159fd:      call    0x10240
0x15a02:      mov     rdi, qword ptr [rsp]
0x15a06:      call    0x18d10
0x15a0b:      mov     rax, qword ptr [rsp + 8]
0x15a10:      sub     rax, qword ptr fs:[0x28]
0x15a19:      jne      0x15ba4
0x15a1f:      add     rsp, 0x18
0x15a23:      pop     rbx
0x15a24:      pop     rbp
0x15a25:      pop     r12
0x15a27:      pop     r13
0x15a29:      ret
0x15a2a:      nop     word ptr [rax + rax]
0x15a30:      mov     rsi, qword ptr [rsp]
0x15a34:      mov     rdi, rbx
0x15a37:      xor     eax, eax
0x15a39:      call    0x122d0
0x15a3e:      mov     edx, dword ptr [rbx + 0x60]
0x15a41:      mov     esi, dword ptr [rbx + 0x68]
0x15a44:      mov     edi, 0

```

```

0x15a49:    mov     dword ptr [rbx + 0x28], eax
0x15a4c:    sub     edx, 2
0x15a4f:    mov     ecx, edx
0x15a51:    shr     ecx, 0x1f
0x15a54:    add     ecx, edx
0x15a56:    sar     ecx, 1
0x15a58:    sub     eax, ecx
0x15a5a:    lea     ecx, [rdx + rax]
0x15a5d:    cmovs   eax, edi
0x15a60:    cmp     esi, ecx
0x15a62:    jg      0x15a70
0x15a64:    mov     eax, esi
0x15a66:    sub     eax, edx
0x15a68:    mov     edx, 0
0x15a6d:    cmovs   eax, edx
0x15a70:    cmp     byte ptr [rbx + 0x6d], 0
0x15a74:    mov     dword ptr [rbx + 0x2c], eax
0x15a77:    jne     0x15ac0
0x15a79:    mov     rdi, rbx
0x15a7c:    xor     eax, eax
0x15a7e:    call    0x10240
0x15a83:    jmp     0x159e3
0x159e3:
0x15a88:    nop     dword ptr [rax + rax]
0x15a90:    mov     rsi, qword ptr [rsp]
0x15a94:    mov     ecx, 0x1c
0x15a99:    xor     edx, edx
0x15a9b:    xor     eax, eax
0x15a9d:    lea     rdi, [rip + 0xb993]
0x15aa4:    call    0xf760
0x15aa9:    mov     rdi, qword ptr [rsp]
0x15aad:    call    0x18d10
0x15ab2:    jmp     0x15a0b
0x15a0b:
0x15ab7:    nop     word ptr [rax + rax]
0x15ac0:    xor     eax, eax
0x15ac2:    mov     rdi, rbx
0x15ac5:    call    0x10d70
0x15aca:    cmp     byte ptr [rbx + 0x6d], 0
0x15ace:    je      0x15a79
0x15ad0:    mov     rdi, rbx
0x15ad3:    xor     eax, eax
0x15ad5:    call    0xec50
0x15ada:    mov     rdi, rbx
0x15add:    xor     eax, eax
0x15adf:    call    0x10240
0x15ae4:    jmp     0x159e3
0x159e3:

```



```

0x15ae9:      nop      dword ptr [rax]
0x15af0:      movsxd   rax, dword ptr [rbp + 0x28]
0x15af4:      lea      rax, [rax + rax*4]
0x15af8:      shl      rax, 4
0x15afc:      add      rax, qword ptr [rbp + 0x38]
0x15b00:      mov      rdi, qword ptr [rax]
0x15b03:      call     0x1d7a0
0x15b08:      mov      rsi, qword ptr [rbp + 0x18]
0x15b0c:      mov      edx, 1
0x15b11:      mov      rdi, rbp
0x15b14:      mov      r12, rax
0x15b17:      xor      eax, eax
0x15b19:      call     0xf880
0x15b1e:      test     eax, eax
0x15b20:      jne      0x15b40
0x15b22:      mov      rdi, rbp
0x15b25:      call     0x14d00
0x15b2a:      mov      rdi, r12
0x15b2d:      call     0x18d10
0x15b32:      jmp      0x159f8
0x159f8
0x15b37:      nop      word ptr [rax + rax]
0x15b40:      mov      rsi, r12
0x15b43:      mov      rdi, rbp
0x15b46:      xor      eax, eax
0x15b48:      call     0x122d0
0x15b4d:      mov      edx, dword ptr [rbp + 0x60]
0x15b50:      mov      esi, dword ptr [rbp + 0x68]
0x15b53:      mov      edi, 0
0x15b58:      mov      dword ptr [rbp + 0x28], eax
0x15b5b:      sub      edx, 2
0x15b5e:      mov      ecx, edx
0x15b60:      shr      ecx, 0x1f
0x15b63:      add      ecx, edx
0x15b65:      sar      ecx, 1
0x15b67:      sub      eax, ecx
0x15b69:      lea      ecx, [rdx + rax]
0x15b6c:      cmovs    eax, edi
0x15b6f:      cmp      esi, ecx
0x15b71:      jg       0x15b7f
0x15b73:      mov      eax, esi
0x15b75:      sub      eax, edx
0x15b77:      mov      edx, 0
0x15b7c:      cmovs    eax, edx
0x15b7f:      cmp      byte ptr [rbp + 0x6d], 0
0x15b83:      mov      dword ptr [rbp + 0x2c], eax
0x15b86:      je       0x15b2a
0x15b88:      xor      eax, eax

```

```

0x15b8a:    mov     rdi, rbp
0x15b8d:    call   0x10d70
0x15b92:    cmp     byte ptr [rbp + 0x6d], 0
0x15b96:    je      0x15b2a
0x15b98:    mov     rdi, rbp
0x15b9b:    xor     eax, eax
0x15b9d:    call   0xec50
0x15ba2:    jmp     0x15b2a
0x15b2a:
0x15ba4:    call   0x49d0
0x15ba9:    nop     dword ptr [rax]
0x15bb0:    endbr64
0x15bb4:    push    r15
0x15bb6:    mov     r15, rdi
0x15bb9:    push    r14
0x15bbb:    push    r13
0x15bbd:    push    r12
0x15bbf:    mov     r12, rsi
0x15bc2:    push    rbp
0x15bc3:    push    rbx
0x15bc4:    sub     rsp, 0x48
0x15bc8:    mov     edx, dword ptr [rdi + 0x70]
0x15bcb:    mov     rax, qword ptr fs:[0x28]
0x15bd4:    mov     qword ptr [rsp + 0x38], rax
0x15bd9:    xor     eax, eax
0x15bdb:    mov     qword ptr [rsp + 0x28], 0
0x15be4:    mov     qword ptr [rsp + 0x30], 0
0x15bed:    mov     dword ptr [rdi + 0x90], 1
0x15bf7:    test    edx, edx
0x15bf9:    jne     0x15ec8
0x15bff:    movsxd  rax, dword ptr [rdi + 0x28]
0x15c03:    mov     rdx, rax
0x15c06:    lea     rax, [rax + rax*4]
0x15c0a:    shl     rax, 4
0x15c0e:    add     rax, qword ptr [rdi + 0x38]
0x15c12:    mov     rbp, qword ptr [rax]
0x15c15:    test    edx, edx
0x15c17:    jne     0x15c30
0x15c19:    mov     rax, qword ptr [rdi + 0x18]
0x15c1d:    cmp     byte ptr [rax], 0x2f
0x15c20:    jne     0x15ea0
0x15c26:    cmp     byte ptr [rax + 1], 0
0x15c2a:    jne     0x15ea0
0x15c30:    mov     rdi, rbp
0x15c33:    call   0x49b0
0x15c38:    lea     rdi, [rax + 0x11]
0x15c3c:    call   0x1d670
0x15c41:    mov     edx, 0xe

```

```

0x15c46:    mov     rsi, rbp
0x15c49:    lea     rdi, [rip + 0x15990]
0x15c50:    mov     r13, rax
0x15c53:    call    0x4880
0x15c58:    lea     rcx, [rip + 0xb7ef]
0x15c5f:    mov     esi, 1
0x15c64:    mov     rdi, r13
0x15c67:    mov     r8, rax
0x15c6a:    mov     rdx, -1
0x15c71:    xor     eax, eax
0x15c73:    call    0x4f10
0x15c78:    mov     rdi, rbp
0x15c7b:    call    0x49b0
0x15c80:    mov     rdi, qword ptr [r12 + 0x18]
0x15c85:    mov     rbx, rax
0x15c88:    call    0x49b0
0x15c8d:    lea     rdi, [rbx + rax + 2]
0x15c92:    call    0x1d670
0x15c97:    mov     r8, qword ptr [r12 + 0x18]
0x15c9c:    mov     r9, rbp
0x15c9f:    lea     rcx, [rip + 0xaa5b]
0x15ca6:    mov     rdi, rax
0x15ca9:    mov     r14, rax
0x15cac:    mov     esi, 1
0x15cb1:    xor     eax, eax
0x15cb3:    mov     rdx, -1
0x15cba:    call    0x4f10
0x15cbf:    mov     rdi, r13
0x15cc2:    lea     rsi, [rsp + 0x30]
0x15cc7:    mov     rdx, r14
0x15cca:    mov     rcx, qword ptr [rip + 0x1593f]
0x15cd1:    call    0x83e0
0x15cd6:    mov     rdi, r13
0x15cd9:    test    rax, rax
0x15cdc:    je      0x16173
0x15ce2:    call    0x18d10
0x15ce7:    movsxd  rax, dword ptr [r15 + 0x28]
0x15ceb:    lea     rax, [rax + rax*4]
0x15cef:    shl     rax, 4
0x15cf3:    add     rax, qword ptr [r15 + 0x38]
0x15cf7:    mov     eax, dword ptr [rax + 0x10]
0x15cfa:    and     eax, 0xf000
0x15cff:    cmp     eax, 0x4000
0x15d04:    je      0x160a0
0x15d0a:    mov     rdi, qword ptr [rip + 0x1479f]
0x15d11:    call    0xdb60
0x15d16:    xor     eax, eax
0x15d18:    call    0xa820

```

```

0x15d1d:    mov     rdi, qword ptr [rsp + 0x30]
0x15d22:    xor     eax, eax
0x15d24:    call    0x19940
0x15d29:    mov     rdi, qword ptr [rsp + 0x30]
0x15d2e:    mov     r13, rax
0x15d31:    call    0x18d10
0x15d36:    mov     rdi, rbp
0x15d39:    xor     eax, eax
0x15d3b:    mov     rsi, r13
0x15d3e:    mov     qword ptr [rsp + 0x30], r13
0x15d43:    call    0x11190
0x15d48:    mov     rdi, r14
0x15d4b:    mov     ebx, eax
0x15d4d:    call    0x18d10
0x15d52:    test    ebx, ebx
0x15d54:    jne     0x16150
0x15d5a:    movsxd  rax, dword ptr [r15 + 0x28]
0x15d5e:    mov     rdx, qword ptr [rsp + 0x30]
0x15d63:    mov     rsi, rbp
0x15d66:    mov     rdi, r15
0x15d69:    lea     rax, [rax + rax*4]
0x15d6d:    shl     rax, 4
0x15d71:    add     rax, qword ptr [r15 + 0x38]
0x15d75:    mov     ecx, dword ptr [rax + 0x10]
0x15d78:    xor     eax, eax
0x15d7a:    call    0x11d10
0x15d7f:    cmp     eax, 1
0x15d82:    je      0x15db6
0x15d84:    cmp     eax, 2
0x15d87:    je      0x161a0
0x15d8d:    sub     eax, 1
0x15d90:    lea     rdx, [rip + 0x14629]
0x15d97:    xor     ecx, ecx
0x15d99:    mov     rsi, rbp
0x15d9c:    cdqe
0x15d9e:    mov     r8d, 0xe
0x15da4:    lea     rdi, [rip + 0xb6b0]
0x15dab:    mov     rdx, qword ptr [rdx + rax*8]
0x15daf:    xor     eax, eax
0x15db1:    call    0xf7e0
0x15db6:    mov     rdi, qword ptr [rsp + 0x30]
0x15dbb:    call    0x18d10
0x15dc0:    xor     eax, eax
0x15dc2:    call    0xe4e0
0x15dc7:    mov     rdi, r15
0x15dca:    xor     eax, eax
0x15dcc:    call    0x10240
0x15dd1:    xor     eax, eax

```

```

0x15dd3:    mov     rdi, r12
0x15dd6:    call   0x10240
0x15ddb:    xor     eax, eax
0x15ddd:    call   0xa820
0x15de2:    mov     rsi, qword ptr [r12 + 0x18]
0x15de7:    xor     eax, eax
0x15de9:    mov     edx, 1
0x15dee:    mov     rdi, r12
0x15df1:    call   0xf880
0x15df6:    test    eax, eax
0x15df8:    je      0x16190
0x15dfe:    xor     eax, eax
0x15e00:    mov     rdi, r12
0x15e03:    call   0x10ec0
0x15e08:    xor     eax, eax
0x15e0a:    mov     rdi, r12
0x15e0d:    call   0x10160
0x15e12:    cmp     byte ptr [r12 + 0x6d], 0
0x15e18:    je      0x15e24
0x15e1a:    mov     rdi, r12
0x15e1d:    xor     eax, eax
0x15e1f:    call   0xec50
0x15e24:    xor     eax, eax
0x15e26:    mov     rdi, r12
0x15e29:    call   0x10240
0x15e2e:    cmp     byte ptr [r12 + 0x6d], 0
0x15e34:    je      0x15e40
0x15e36:    mov     rdi, r12
0x15e39:    xor     eax, eax
0x15e3b:    call   0x10d70
0x15e40:    mov     rsi, qword ptr [r15 + 0x18]
0x15e44:    xor     eax, eax
0x15e46:    mov     edx, 1
0x15e4b:    mov     rdi, r15
0x15e4e:    call   0xf880
0x15e53:    test    eax, eax
0x15e55:    je      0x16180
0x15e5b:    xor     eax, eax
0x15e5d:    mov     rdi, r15
0x15e60:    call   0x10ec0
0x15e65:    xor     eax, eax
0x15e67:    mov     rdi, r15
0x15e6a:    call   0x10160
0x15e6f:    cmp     byte ptr [r15 + 0x6d], 0
0x15e74:    je      0x15e80
0x15e76:    mov     rdi, r15
0x15e79:    xor     eax, eax
0x15e7b:    call   0xec50

```

```

0x15e80:    xor     eax, eax
0x15e82:    mov     rdi, r15
0x15e85:    call    0x10240
0x15e8a:    cmp     byte ptr [r15 + 0x6d], 0
0x15e8f:    je      0x15ea0
0x15e91:    mov     rdi, r15
0x15e94:    xor     eax, eax
0x15e96:    call    0x10d70
0x15e9b:    nop     dword ptr [rax + rax]
0x15ea0:    mov     rax, qword ptr [rsp + 0x38]
0x15ea5:    sub     rax, qword ptr fs:[0x28]
0x15eae:    jne     0x161bd
0x15eb4:    add     rsp, 0x48
0x15eb8:    pop     rbx
0x15eb9:    pop     rbp
0x15eba:    pop     r12
0x15ebc:    pop     r13
0x15ebe:    pop     r14
0x15ec0:    pop     r15
0x15ec2:    ret
0x15ec3:    nop     dword ptr [rax + rax]
0x15ec8:    mov     rdx, qword ptr [rsi + 0x18]
0x15ecc:    mov     rcx, qword ptr [rip + 0x1573d]
0x15ed3:    lea     rsi, [rsp + 0x28]
0x15ed8:    lea     rdi, [rip + 0xb591]
0x15edf:    call    0x83e0
0x15ee4:    test    rax, rax
0x15ee7:    je      0x15ea0
0x15ee9:    mov     rdi, qword ptr [r15 + 0x18]
0x15eed:    mov     rsi, qword ptr [rsp + 0x28]
0x15ef2:    xor     eax, eax
0x15ef4:    call    0x11190
0x15ef9:    mov     r13d, eax
0x15efc:    test    eax, eax
0x15efe:    jne     0x160b8
0x15f04:    mov     rbp, qword ptr [rsp + 0x28]
0x15f09:    mov     rdi, rbp
0x15f0c:    call    0x49b0
0x15f11:    mov     rdi, rbp
0x15f14:    mov     rbx, rax
0x15f17:    lea     rax, [rax + 1]
0x15f1b:    lea     r14, [rbx + 2]
0x15f1f:    mov     qword ptr [rsp + 8], rax
0x15f24:    mov     rsi, r14
0x15f27:    call    0x1d690
0x15f2c:    mov     qword ptr [rsp + 0x28], rax
0x15f31:    mov     byte ptr [rax + rbx], 0x2f
0x15f35:    mov     rax, qword ptr [rsp + 0x28]

```

```

0x15f3a:    mov     byte ptr [rax + rbx + 1], 0
0x15f3f:    mov     edx, dword ptr [r15 + 0x68]
0x15f43:    test    edx, edx
0x15f45:    jle     0x15f6a
0x15f47:    mov     rax, qword ptr [r15 + 0x38]
0x15f4b:    add     rax, 0x49
0x15f4f:    jmp     0x15f65
0x15f65:
0x15f51:    nop     dword ptr [rax]
0x15f58:    add     r13d, 1
0x15f5c:    add     rax, 0x50
0x15f60:    cmp     r13d, edx
0x15f63:    je      0x15f6a
0x15f65:    cmp     byte ptr [rax], 0
0x15f68:    je      0x15f58
0x15f6a:    mov     eax, dword ptr [r15 + 0x70]
0x15f6e:    mov     dword ptr [r15 + 0x50], 0xffffffff
0x15f76:    mov     dword ptr [r15 + 0x54], eax
0x15f7a:    nop     word ptr [rax + rax]
0x15f80:    mov     rdi, r15
0x15f83:    call    0x100e0
0x15f88:    cmp     eax, -1
0x15f8b:    je      0x16060
0x15f91:    cdqe
0x15f93:    lea     rbx, [rax + rax*4]
0x15f97:    mov     rax, qword ptr [r15 + 0x38]
0x15f9b:    shl     rbx, 4
0x15f9f:    mov     rbp, qword ptr [rax + rbx]
0x15fa3:    xor     eax, eax
0x15fa5:    call    0x18ab0
0x15faa:    mov     rdi, rbp
0x15fad:    call    0x49b0
0x15fb2:    mov     rdi, qword ptr [rsp + 0x28]
0x15fb7:    lea     rsi, [r14 + rax]
0x15fbb:    call    0x1d690
0x15fc0:    mov     rcx, qword ptr [rsp + 8]
0x15fc5:    mov     rsi, rbp
0x15fc8:    mov     qword ptr [rsp + 0x28], rax
0x15fcd:    lea     rdi, [rax + rcx]
0x15fd1:    call    0x48b0
0x15fd6:    mov     eax, dword ptr [rip + 0x15ad4]
0x15fdc:    test    eax, eax
0x15fde:    jne     0x160e0
0x15fe4:    mov     rdi, qword ptr [rip + 0x144cd]
0x15feb:    call    0xdb60
0x15ff0:    xor     eax, eax
0x15ff2:    call    0xa820
0x15ff7:    mov     rax, qword ptr [r15 + 0x38]

```

```

0x15ffb:    mov     rsi, rbp
0x15ffe:    mov     rdi, r15
0x16001:    mov     rdx, qword ptr [rsp + 0x28]
0x16006:    mov     ecx, dword ptr [rax + rbx + 0x10]
0x1600a:    xor     eax, eax
0x1600c:    call    0x11d10
0x16011:    cmp     eax, 1
0x16014:    je      0x160d0
0x1601a:    cmp     eax, 2
0x1601d:    je      0x16060
0x1601f:    cmp     eax, 3
0x16022:    je      0x15f80
0x16028:    sub     eax, 1
0x1602b:    lea     rdx, [rip + 0x1438e]
0x16032:    xor     ecx, ecx
0x16034:    mov     rsi, rbp
0x16037:    cdq     rsi
0x16039:    mov     r8d, 0xe
0x1603f:    lea     rdi, [rip + 0xb415]
0x16046:    mov     rdx, qword ptr [rdx + rax*8]
0x1604a:    xor     eax, eax
0x1604c:    call    0xf7e0
0x16051:    test    al, al
0x16053:    jne     0x15f80
0x16059:    nop     dword ptr [rax]
0x16060:    mov     rdi, qword ptr [rsp + 0x28]
0x16065:    test    rdi, rdi
0x16068:    je      0x1606f
0x1606a:    call    0x18d10
0x1606f:    cmp     dword ptr [r15 + 0x68], r13d
0x16073:    je      0x16087
0x16075:    mov     dword ptr [r15 + 0x28], r13d
0x16079:    mov     rdi, r15
0x1607c:    xor     eax, eax
0x1607e:    call    0xf110
0x16083:    mov     dword ptr [r15 + 0x2c], eax
0x16087:    xor     eax, eax
0x16089:    call    0xe4e0
0x1608e:    xor     eax, eax
0x16090:    call    0xa820
0x16095:    jmp     0x15de2
0x15de2:
0x1609a:    nop     word ptr [rax + rax]
0x160a0:    mov     rdi, qword ptr [rip + 0x14419]
0x160a7:    call    0xdb60
0x160ac:    jmp     0x15d16
0x15d16:
0x160b1:    nop     dword ptr [rax]

```



```

0x160b8:      mov     edx, 0xc
0x160bd:      xor     esi, esi
0x160bf:      lea     rdi, [rip + 0x1423a]
0x160c6:      call    0x8070
0x160cb:      jmp     0x15ea0
0x15ea0
0x160d0:      mov     rax, qword ptr [r15 + 0x38]
0x160d4:      mov     byte ptr [rax + rbx + 0x49], 0
0x160d9:      jmp     0x15f80
0x15f80
0x160de:      nop
0x160e0:      mov     dword ptr [rip + 0x159c6], 0
0x160ea:      xor     eax, eax
0x160ec:      call    0xc8c0
0x160f1:      mov     edx, 4
0x160f6:      lea     rsi, [rip + 0xa435]
0x160fd:      lea     rdi, [rip + 0xb0f5]
0x16104:      mov     qword ptr [rsp + 0x18], rax
0x16109:      call    0x8070
0x1610e:      mov     rdi, qword ptr [rsp + 0x18]
0x16113:      mov     byte ptr [rsp + 0x17], al
0x16117:      call    0xc970
0x1611c:      xor     eax, eax
0x1611e:      call    0xd870
0x16123:      xor     eax, eax
0x16125:      call    0xd740
0x1612a:      xor     eax, eax
0x1612c:      call    0xa820
0x16131:      movzx   eax, byte ptr [rsp + 0x17]
0x16136:      and     eax, 0xffffffff
0x16139:      cmp     al, 0x4e
0x1613b:      je      0x15fe4
0x16141:      jmp     0x16060
0x16060
0x16146:      nop     word ptr cs:[rax + rax]
0x16150:      mov     rdx, qword ptr [rsp + 0x30]
0x16155:      lea     rdi, [rip + 0xbaf4]
0x1615c:      xor     ecx, ecx
0x1615e:      xor     eax, eax
0x16160:      mov     r8d, 0x1e
0x16166:      mov     rsi, rbp
0x16169:      call    0xf7e0
0x1616e:      mov     rdi, qword ptr [rsp + 0x30]
0x16173:      call    0x18d10
0x16178:      jmp     0x15ea0
0x15ea0
0x1617d:      nop     dword ptr [rax]
0x16180:      mov     rdi, r15

```

```

0x16183:      call    0x14d00
0x16188:      jmp     0x15e5b
0x15e5b
0x1618d:      nop     dword ptr [rax]
0x16190:      mov     rdi, r12
0x16193:      call    0x14d00
0x16198:      jmp     0x15dfe
0x15dfe
0x1619d:      nop     dword ptr [rax]
0x161a0:      mov     rdi, qword ptr [rsp + 0x30]
0x161a5:      call    0x18d10
0x161aa:      xor     eax, eax
0x161ac:      call    0xe4e0
0x161b1:      xor     eax, eax
0x161b3:      call    0xa820
0x161b8:      jmp     0x15ea0
0x15ea0
0x161bd:      call    0x49d0
0x161c2:      nop     word ptr cs:[rax + rax]
0x161cd:      nop     dword ptr [rax]
0x161d0:      endbr64
0x161d4:      push    r13
0x161d6:      xor     r13d, r13d
0x161d9:      push    r12
0x161db:      lea     r12, [rip + 0xc203]
0x161e2:      push    rbp
0x161e3:      mov     rbp, rdi
0x161e6:      push    rbx
0x161e7:      mov     rbx, rsi
0x161ea:      sub     rsp, 8
0x161ee:      mov     rdx, qword ptr [rdi + 0x38]
0x161f2:      test    rdx, rdx
0x161f5:      je      0x1621a
0x161f7:      movsxd  rax, dword ptr [rdi + 0x28]
0x161fb:      lea     rax, [rax + rax*4]
0x161ff:      shl     rax, 4
0x16203:      mov     rdi, qword ptr [rdx + rax]
0x16207:      test    rdi, rdi
0x1620a:      je      0x1621a
0x1620c:      call    0x1d7a0
0x16211:      mov     r13d, 1
0x16217:      mov     r12, rax
0x1621a:      xor     edx, edx
0x1621c:      mov     rsi, qword ptr [rbp + 0x18]
0x16220:      cmp     rbx, -1
0x16224:      mov     rdi, rbp
0x16227:      sete    dl
0x1622a:      xor     eax, eax

```

```

0x1622c:      call    0xf880
0x16231:      test    eax, eax
0x16233:      je      0x16278
0x16235:      cmp     rbx, -1
0x16239:      je      0x16288
0x1623b:      mov     dword ptr [rbp + 0x28], 0
0x16242:      mov     rdi, rbp
0x16245:      xor     eax, eax
0x16247:      call    0xf110
0x1624c:      mov     dword ptr [rbp + 0x2c], eax
0x1624f:      test    r13d, r13d
0x16252:      jne     0x16260
0x16254:      add     rsp, 8
0x16258:      pop     rbx
0x16259:      pop     rbp
0x1625a:      pop     r12
0x1625c:      pop     r13
0x1625e:      ret
0x1625f:      nop
0x16260:      add     rsp, 8
0x16264:      mov     rdi, r12
0x16267:      pop     rbx
0x16268:      pop     rbp
0x16269:      pop     r12
0x1626b:      pop     r13
0x1626d:      jmp     0x18d10
0x18d10
0x16272:      nop     word ptr [rax + rax]
0x16278:      mov     rdi, rbp
0x1627b:      xor     eax, eax
0x1627d:      call    0x14d00
0x16282:      jmp     0x1624f
0x1624f
0x16284:      nop     dword ptr [rax]
0x16288:      mov     rsi, r12
0x1628b:      mov     rdi, rbp
0x1628e:      xor     eax, eax
0x16290:      call    0x122d0
0x16295:      mov     rdi, rbp
0x16298:      mov     dword ptr [rbp + 0x28], eax
0x1629b:      xor     eax, eax
0x1629d:      call    0xf110
0x162a2:      mov     dword ptr [rbp + 0x2c], eax
0x162a5:      jmp     0x1624f
0x1624f
0x162a7:      nop     word ptr [rax + rax]
0x162b0:      endbr64
0x162b4:      push    r15

```

```

0x162b6:    push    r14
0x162b8:    push    r13
0x162ba:    push    r12
0x162bc:    mov     r12, rdi
0x162bf:    push    rbp
0x162c0:    movsxd  rax, dword ptr [rdi + 0x28]
0x162c4:    mov     rbp, rsi
0x162c7:    lea     rax, [rax + rax*4]
0x162cb:    shl     rax, 4
0x162cf:    add     rax, qword ptr [rdi + 0x38]
0x162d3:    movzx   edx, byte ptr [rax + 0x48]
0x162d7:    mov     r14, qword ptr [rax]
0x162da:    test    dl, dl
0x162dc:    je      0x16300
0x162de:    cmp     dl, 1
0x162e1:    jne     0x162ed
0x162e3:    cmp     byte ptr [rax + 0x4a], 0
0x162e7:    jne     0x16420
0x162ed:    xor     r13d, r13d
0x162f0:    pop     rbp
0x162f1:    mov     eax, r13d
0x162f4:    pop     r12
0x162f6:    pop     r13
0x162f8:    pop     r14
0x162fa:    pop     r15
0x162fc:    ret
0x162fd:    nop     dword ptr [rax]
0x16300:    cmp     byte ptr [r14], 0x2e
0x16304:    jne     0x163c0
0x1630a:    cmp     byte ptr [r14 + 1], 0x2e
0x1630f:    jne     0x163c0
0x16315:    cmp     byte ptr [r14 + 2], 0
0x1631a:    jne     0x163c0
0x16320:    mov     rax, qword ptr [rdi + 0x18]
0x16324:    cmp     byte ptr [rax], 0x2f
0x16327:    jne     0x16330
0x16329:    cmp     byte ptr [rax + 1], 0
0x1632d:    je      0x162ed
0x1632f:    nop
0x16330:    mov     rax, qword ptr [r12 + 0x78]
0x16335:    lea     rdi, [rax + 1]
0x16339:    call    0x1d670
0x1633e:    mov     rsi, qword ptr [r12 + 0x18]
0x16343:    mov     rdi, rax
0x16346:    mov     r15, rax
0x16349:    call    0x48b0
0x1634e:    mov     edx, 1
0x16353:    mov     rsi, r14

```

```

0x16356:      mov     rdi, r12
0x16359:      xor     eax, eax
0x1635b:      call    0xf880
0x16360:      mov     r13d, eax
0x16363:      test    eax, eax
0x16365:      jne     0x165c0
0x1636b:      xor     eax, eax
0x1636d:      mov     rdi, r12
0x16370:      call    0x14d00
0x16375:      xor     eax, eax
0x16377:      mov     rdi, r12
0x1637a:      call    0x10ec0
0x1637f:      xor     eax, eax
0x16381:      mov     rdi, r12
0x16384:      call    0x10160
0x16389:      cmp     byte ptr [r12 + 0x6d], 0
0x1638f:      jne     0x165b0
0x16395:      xor     eax, eax
0x16397:      mov     rdi, r12
0x1639a:      call    0x10240
0x1639f:      cmp     byte ptr [r12 + 0x6d], 0
0x163a5:      je      0x162ed
0x163ab:      mov     rdi, r12
0x163ae:      xor     eax, eax
0x163b0:      call    0x10d70
0x163b5:      jmp     0x162f0
0x162f0:
0x163ba:      nop     word ptr [rax + rax]
0x163c0:      mov     rax, qword ptr [r12 + 0x78]
0x163c5:      lea     rdi, [rax + 1]
0x163c9:      call    0x1d670
0x163ce:      mov     rsi, qword ptr [r12 + 0x18]
0x163d3:      mov     rdi, rax
0x163d6:      mov     r15, rax
0x163d9:      call    0x48b0
0x163de:      mov     edx, 1
0x163e3:      mov     rsi, r14
0x163e6:      mov     rdi, r12
0x163e9:      xor     eax, eax
0x163eb:      call    0xf880
0x163f0:      mov     r13d, eax
0x163f3:      test    eax, eax
0x163f5:      jne     0x164b0
0x163fb:      mov     ecx, 0x1c
0x16400:      xor     edx, edx
0x16402:      mov     rsi, r14
0x16405:      xor     eax, eax
0x16407:      lea     rdi, [rip + 0xaf94]

```

```

0x1640e:      call    0xf760
0x16413:      jmp     0x162f0
0x162f0:
0x16418:      nop     dword ptr [rax + rax]
0x16420:      mov     rdi, r14
0x16423:      call    0x49b0
0x16428:      lea     rdi, [rax + 0x21]
0x1642c:      call    0x1d670
0x16431:      mov     r8, r14
0x16434:      mov     esi, 1
0x16439:      lea     rcx, [rip + 0xb04b]
0x16440:      mov     rdx, -1
0x16447:      mov     r13, rax
0x1644a:      mov     rdi, rax
0x1644d:      xor     eax, eax
0x1644f:      call    0x4f10
0x16454:      xor     esi, esi
0x16456:      mov     rdi, r13
0x16459:      call    0x193f0
0x1645e:      mov     rdi, r13
0x16461:      mov     r13d, 1
0x16467:      mov     dword ptr [rip + 0x1446b], 1
0x16471:      call    0x18d10
0x16476:      xor     eax, eax
0x16478:      call    0xa480
0x1647d:      mov     rdi, qword ptr [rip + 0x14444]
0x16484:      call    0xbb10
0x16489:      mov     rdi, r14
0x1648c:      mov     dword ptr [r12 + 0x14], 0x3fffffff
0x16495:      mov     dword ptr [rbp + 0x14], 0x3fffffff
0x1649c:      call    0xd610
0x164a1:      mov     rdi, qword ptr [r12 + 0x18]
0x164a6:      call    0xc4b0
0x164ab:      jmp     0x162f0
0x162f0:
0x164b0:      mov     dword ptr [r12 + 0x28], 0
0x164b9:      mov     rdi, r12
0x164bc:      xor     eax, eax
0x164be:      call    0xf110
0x164c3:      mov     dword ptr [r12 + 0x2c], eax
0x164c8:      mov     rdi, r15
0x164cb:      call    0x18d10
0x164d0:      xor     eax, eax
0x164d2:      mov     rdi, r12
0x164d5:      call    0x10ec0
0x164da:      xor     eax, eax
0x164dc:      mov     rdi, r12
0x164df:      call    0x10160

```

```

0x164e4:    cmp     byte ptr [r12 + 0x6d], 0
0x164ea:    je      0x164f6
0x164ec:    mov     rdi, r12
0x164ef:    xor     eax, eax
0x164f1:    call    0xec50
0x164f6:    xor     eax, eax
0x164f8:    mov     rdi, r12
0x164fb:    call    0x10240
0x16500:    cmp     byte ptr [r12 + 0x6d], 0
0x16506:    je      0x16512
0x16508:    mov     rdi, r12
0x1650b:    xor     eax, eax
0x1650d:    call    0x10d70
0x16512:    mov     rsi, qword ptr [rbp + 0x18]
0x16516:    mov     rdi, qword ptr [r12 + 0x18]
0x1651b:    call    0x4b30
0x16520:    mov     r13d, eax
0x16523:    test    eax, eax
0x16525:    jne     0x16588
0x16527:    mov     r8d, 1
0x1652d:    mov     rdx, r12
0x16530:    mov     esi, 0x22
0x16535:    mov     rdi, rbp
0x16538:    mov     rcx, -1
0x1653f:    call    0x13e30
0x16544:    xor     eax, eax
0x16546:    mov     rdi, rbp
0x16549:    call    0x10ec0
0x1654e:    xor     eax, eax
0x16550:    mov     rdi, rbp
0x16553:    call    0x10160
0x16558:    cmp     byte ptr [rbp + 0x6d], 0
0x1655c:    jne     0x165a0
0x1655e:    xor     eax, eax
0x16560:    mov     rdi, rbp
0x16563:    call    0x10240
0x16568:    cmp     byte ptr [rbp + 0x6d], 0
0x1656c:    je      0x162ed
0x16572:    mov     rdi, rbp
0x16575:    xor     eax, eax
0x16577:    call    0x10d70
0x1657c:    jmp     0x162f0
0x162f0:
0x16581:    nop     dword ptr [rax]
0x16588:    mov     rdi, rbp
0x1658b:    xor     eax, eax
0x1658d:    call    0x10240
0x16592:    jmp     0x162ed

```

```

0x162ed
0x16597:      nop      word ptr [rax + rax]
0x165a0:      mov      rdi, rbp
0x165a3:      xor      eax, eax
0x165a5:      call     0xec50
0x165aa:      jmp      0x1655e
0x1655e
0x165ac:      nop      dword ptr [rax]
0x165b0:      mov      rdi, r12
0x165b3:      xor      eax, eax
0x165b5:      call     0xec50
0x165ba:      jmp      0x16395
0x16395
0x165bf:      nop
0x165c0:      mov      rdi, r15
0x165c3:      mov      esi, 0x2f
0x165c8:      call     0x4a20
0x165cd:      mov      rdi, r12
0x165d0:      lea      rsi, [rax + 1]
0x165d4:      xor      eax, eax
0x165d6:      call     0x122d0
0x165db:      mov      rdi, r12
0x165de:      mov      dword ptr [r12 + 0x28], eax
0x165e3:      xor      eax, eax
0x165e5:      call     0xf110
0x165ea:      jmp      0x164c3
0x164c3
0x165ef:      nop
0x165f0:      endbr64
0x165f4:      push     r13
0x165f6:      push     r12
0x165f8:      mov      r12, rdx
0x165fb:      push     rbp
0x165fc:      mov      rbp, rsi
0x165ff:      push     rbx
0x16600:      mov      rbx, rdi
0x16603:      mov      rdi, rdx
0x16606:      sub      rsp, 8
0x1660a:      call     0x49b0
0x1660f:      cmp      byte ptr [r12], 0x2f
0x16614:      je       0x166f0
0x1661a:      add      rax, qword ptr [rbx + 0x78]
0x1661e:      mov      rdi, qword ptr [rbx + 0x18]
0x16622:      lea      rdx, [rax + 1]
0x16626:      lea      rsi, [rax + 2]
0x1662a:      mov      qword ptr [rbx + 0x78], rdx
0x1662e:      call     0x1d690
0x16633:      mov      qword ptr [rbx + 0x18], rax

```



```

0x16637:      mov     rdi, rax
0x1663a:      mov     r13, rax
0x1663d:      call    0x49b0
0x16642:      mov     edx, 0x2f
0x16647:      mov     rsi, r12
0x1664a:      mov     word ptr [r13 + rax], dx
0x16650:      mov     rdi, qword ptr [rbx + 0x18]
0x16654:      call    0x4df0
0x16659:      mov     rdi, qword ptr [rbx + 0x18]
0x1665d:      call    0x18040
0x16662:      mov     dword ptr [rbx + 0x28], 0
0x16669:      mov     rdi, rbx
0x1666c:      xor     eax, eax
0x1666e:      mov     r8d, 1
0x16674:      call    0xf110
0x16679:      xor     ecx, ecx
0x1667b:      xor     edx, edx
0x1667d:      mov     esi, 0x22
0x16682:      mov     dword ptr [rbx + 0x2c], eax
0x16685:      mov     rdi, rbx
0x16688:      call    0x13e30
0x1668d:      xor     eax, eax
0x1668f:      mov     rdi, rbx
0x16692:      call    0x10ec0
0x16697:      xor     eax, eax
0x16699:      mov     rdi, rbx
0x1669c:      call    0x10160
0x166a1:      cmp     byte ptr [rbx + 0x6d], 0
0x166a5:      je      0x166b1
0x166a7:      mov     rdi, rbx
0x166aa:      xor     eax, eax
0x166ac:      call    0xec50
0x166b1:      xor     eax, eax
0x166b3:      mov     rdi, rbx
0x166b6:      call    0x10240
0x166bb:      cmp     byte ptr [rbx + 0x6d], 0
0x166bf:      je      0x166cb
0x166c1:      mov     rdi, rbx
0x166c4:      xor     eax, eax
0x166c6:      call    0x10d70
0x166cb:      mov     rsi, qword ptr [rbp + 0x18]
0x166cf:      mov     rdi, qword ptr [rbx + 0x18]
0x166d3:      call    0x4b30
0x166d8:      test    eax, eax
0x166da:      je      0x16718
0x166dc:      add     rsp, 8
0x166e0:      pop     rbx
0x166e1:      pop     rbp

```

```

0x166e2:      pop     r12
0x166e4:      pop     r13
0x166e6:      ret
0x166e7:      nop     word ptr [rax + rax]
0x166f0:      mov     qword ptr [rbx + 0x78], rax
0x166f4:      mov     rdi, qword ptr [rbx + 0x18]
0x166f8:      lea     rsi, [rax + 1]
0x166fc:      call    0x1d690
0x16701:      mov     rsi, r12
0x16704:      mov     qword ptr [rbx + 0x18], rax
0x16708:      mov     rdi, rax
0x1670b:      call    0x48b0
0x16710:      jmp     0x16662
0x16662
0x16715:      nop     dword ptr [rax]
0x16718:      xor     edx, edx
0x1671a:      mov     r8d, 1
0x16720:      mov     esi, 0x22
0x16725:      mov     rdi, rbp
0x16728:      mov     rcx, -1
0x1672f:      call    0x13e30
0x16734:      xor     eax, eax
0x16736:      mov     rdi, rbp
0x16739:      call    0x10ec0
0x1673e:      xor     eax, eax
0x16740:      mov     rdi, rbp
0x16743:      call    0x10160
0x16748:      cmp     byte ptr [rbp + 0x6d], 0
0x1674c:      jne     0x16780
0x1674e:      xor     eax, eax
0x16750:      mov     rdi, rbp
0x16753:      call    0x10240
0x16758:      cmp     byte ptr [rbp + 0x6d], 0
0x1675c:      je      0x166dc
0x16762:      add     rsp, 8
0x16766:      mov     rdi, rbp
0x16769:      xor     eax, eax
0x1676b:      pop     rbx
0x1676c:      pop     rbp
0x1676d:      pop     r12
0x1676f:      pop     r13
0x16771:      jmp     0x10d70
0x10d70
0x16776:      nop     word ptr cs:[rax + rax]
0x16780:      xor     eax, eax
0x16782:      mov     rdi, rbp
0x16785:      call    0xec50
0x1678a:      xor     eax, eax

```

```

0x1678c:      mov     rdi, rbp
0x1678f:      call    0x10240
0x16794:      cmp     byte ptr [rbp + 0x6d], 0
0x16798:      je      0x166dc
0x1679e:      jmp     0x16762
0x16762:
0x167a0:      endbr64
0x167a4:      push    r15
0x167a6:      lea     r15, [rip + 0x13bd3]
0x167ad:      push    r14
0x167af:      lea     r14, [rip + 0xacdc]
0x167b6:      push    r13
0x167b8:      mov     r13, rsi
0x167bb:      push    r12
0x167bd:      mov     r12d, edx
0x167c0:      push    rbp
0x167c1:      mov     rbp, rdi
0x167c4:      push    rbx
0x167c5:      sub     rsp, 8
0x167c9:      mov     rdi, qword ptr [rip + 0x13d08]
0x167d0:      call    0xdb60
0x167d5:      xor     eax, eax
0x167d7:      call    0xa820
0x167dc:      mov     eax, dword ptr [rbp + 0x70]
0x167df:      mov     dword ptr [rbp + 0x50], 0xffffffff
0x167e6:      mov     dword ptr [rbp + 0x90], 1
0x167f0:      mov     dword ptr [rbp + 0x54], eax
0x167f3:      nop     dword ptr [rax + rax]
0x167f8:      mov     rdi, rbp
0x167fb:      call    0x100e0
0x16800:      movsxd  rbx, eax
0x16803:      cmp     ebx, -1
0x16806:      je      0x1686b
0x16808:      xor     eax, eax
0x1680a:      call    0x18ab0
0x1680f:      xor     eax, eax
0x16811:      mov     edx, r12d
0x16814:      mov     esi, ebx
0x16816:      mov     rdi, rbp
0x16819:      call    0x134d0
0x1681e:      cmp     eax, 1
0x16821:      je      0x168f0
0x16827:      cmp     eax, 2
0x1682a:      je      0x1686b
0x1682c:      cmp     eax, 3
0x1682f:      je      0x167f8
0x16831:      sub     eax, 1
0x16834:      xor     ecx, ecx

```

```

0x16836:      mov     rdi, r14
0x16839:      mov     r8d, 0x1e
0x1683f:      cdqe
0x16841:      mov     rdx, qword ptr [r15 + rax*8]
0x16845:      lea     rax, [rbx + rbx*4]
0x16849:      shl     rax, 4
0x1684d:      add     rax, qword ptr [rbp + 0x38]
0x16851:      mov     rsi, qword ptr [rax]
0x16854:      xor     eax, eax
0x16856:      call    0xf7e0
0x1685b:      mov     rdi, rbp
0x1685e:      call    0x100e0
0x16863:      movsxd   rbx, eax
0x16866:      cmp     ebx, -1
0x16869:      jne     0x16808
0x1686b:      xor     eax, eax
0x1686d:      call    0xe4e0
0x16872:      xor     eax, eax
0x16874:      call    0xa820
0x16879:      xor     ecx, ecx
0x1687b:      xor     edx, edx
0x1687d:      mov     r8d, 1
0x16883:      mov     esi, 0x22
0x16888:      mov     rdi, rbp
0x1688b:      call    0x13e30
0x16890:      xor     eax, eax
0x16892:      mov     rdi, rbp
0x16895:      call    0x10ec0
0x1689a:      xor     eax, eax
0x1689c:      mov     rdi, rbp
0x1689f:      call    0x10160
0x168a4:      cmp     byte ptr [rbp + 0x6d], 0
0x168a8:      je      0x168b4
0x168aa:      mov     rdi, rbp
0x168ad:      xor     eax, eax
0x168af:      call    0xec50
0x168b4:      xor     eax, eax
0x168b6:      mov     rdi, rbp
0x168b9:      call    0x10240
0x168be:      cmp     byte ptr [rbp + 0x6d], 0
0x168c2:      je      0x168ce
0x168c4:      mov     rdi, rbp
0x168c7:      xor     eax, eax
0x168c9:      call    0x10d70
0x168ce:      mov     rsi, qword ptr [r13 + 0x18]
0x168d2:      mov     rdi, qword ptr [rbp + 0x18]
0x168d6:      call    0x4b30
0x168db:      test    eax, eax

```

```

0x168dd:      je      0x16908
0x168df:      add     rsp, 8
0x168e3:      pop     rbx
0x168e4:      pop     rbp
0x168e5:      pop     r12
0x168e7:      pop     r13
0x168e9:      pop     r14
0x168eb:      pop     r15
0x168ed:      ret
0x168ee:      nop
0x168f0:      lea     rax, [rbx + rbx*4]
0x168f4:      shl     rax, 4
0x168f8:      add     rax, qword ptr [rbp + 0x38]
0x168fc:      mov     byte ptr [rax + 0x49], 0
0x16900:      jmp     0x167f8
0x167f8
0x16905:      nop     dword ptr [rax]
0x16908:      xor     ecx, ecx
0x1690a:      xor     edx, edx
0x1690c:      mov     r8d, 1
0x16912:      mov     esi, 0x22
0x16917:      mov     rdi, r13
0x1691a:      call    0x13e30
0x1691f:      xor     eax, eax
0x16921:      mov     rdi, r13
0x16924:      call    0x10ec0
0x16929:      xor     eax, eax
0x1692b:      mov     rdi, r13
0x1692e:      call    0x10160
0x16933:      cmp     byte ptr [r13 + 0x6d], 0
0x16938:      jne     0x16968
0x1693a:      xor     eax, eax
0x1693c:      mov     rdi, r13
0x1693f:      call    0x10240
0x16944:      cmp     byte ptr [r13 + 0x6d], 0
0x16949:      je      0x168df
0x1694b:      add     rsp, 8
0x1694f:      mov     rdi, r13
0x16952:      xor     eax, eax
0x16954:      pop     rbx
0x16955:      pop     rbp
0x16956:      pop     r12
0x16958:      pop     r13
0x1695a:      pop     r14
0x1695c:      pop     r15
0x1695e:      jmp     0x10d70
0x10d70
0x16963:      nop     dword ptr [rax + rax]

```

```

0x16968:      mov     rdi, r13
0x1696b:      xor     eax, eax
0x1696d:      call    0xec50
0x16972:      jmp     0x1693a
0x1693a:
0x16974:      nop     word ptr cs:[rax + rax]
0x1697e:      nop
0x16980:      push    r15
0x16982:      push    r14
0x16984:      push    r13
0x16986:      push    r12
0x16988:      push    rbp
0x16989:      push    rbx
0x1698a:      sub     rsp, 0x8c8
0x16991:      mov     rdi, qword ptr [rip + 0x14ca8]
0x16998:      mov     r13, qword ptr [rip + 0x13cc1]
0x1699f:      mov     rax, qword ptr fs:[0x28]
0x169a8:      mov     qword ptr [rsp + 0x8b8], rax
0x169b0:      xor     eax, eax
0x169b2:      lea     rax, [rip + 0x14c8f]
0x169b9:      mov     qword ptr [rip + 0x13ca0], rax
0x169c0:      test    rdi, rdi
0x169c3:      je      0x16a78
0x169c9:      lea     rsi, [rsp + 0x20]
0x169ce:      call    0x18f40
0x169d3:      cmp     eax, -1
0x169d6:      je      0x16a78
0x169dc:      mov     rax, qword ptr [rsp + 0x68]
0x169e1:      mov     rdi, qword ptr [rip + 0x14c58]
0x169e8:      lea     rsi, [rip + 0xb5da]
0x169ef:      mov     qword ptr [rsp + 0x10], rax
0x169f4:      mov     rax, qword ptr [rsp + 0x78]
0x169f9:      mov     qword ptr [rsp + 0x18], rax
0x169fe:      call    0x4da0
0x16a03:      mov     r14, rax
0x16a06:      test    rax, rax
0x16a09:      je      0x16a78
0x16a0b:      mov     dword ptr [rsp + 4], 0
0x16a13:      xor     ebp, ebp
0x16a15:      xor     ebx, ebx
0x16a17:      xor     r12d, r12d
0x16a1a:      lea     r15, [rsp + 0xb0]
0x16a22:      nop     word ptr [rax + rax]
0x16a28:      mov     rdx, r14
0x16a2b:      mov     esi, 0x800
0x16a30:      mov     rdi, r15
0x16a33:      call    0x4b00
0x16a38:      test    rax, rax

```

```

0x16a3b:    je      0x16ad0
0x16a41:    cmp     byte ptr [rsp + 0xb0], 0
0x16a49:    je      0x16aa8
0x16a4b:    cmp     dword ptr [r15], 0x6d6f7246
0x16a52:    je      0x16ab0
0x16a54:    test    ebp, ebp
0x16a56:    jne     0x16a28
0x16a58:    test    ebx, ebx
0x16a5a:    je      0x16a28
0x16a5c:    cmp     dword ptr [r15], 0x74617453
0x16a63:    je      0x16b60
0x16a69:    xor     ebp, ebp
0x16a6b:    mov     ebx, 1
0x16a70:    jmp     0x16a28
0x16a28:
0x16a72:    nop     word ptr [rax + rax]
0x16a78:    xor     eax, eax
0x16a7a:    mov     rdx, qword ptr [rsp + 0x8b8]
0x16a82:    sub     rdx, qword ptr fs:[0x28]
0x16a8b:    jne     0x16c02
0x16a91:    add     rsp, 0x8c8
0x16a98:    pop     rbx
0x16a99:    pop     rbp
0x16a9a:    pop     r12
0x16a9c:    pop     r13
0x16a9e:    pop     r14
0x16aa0:    pop     r15
0x16aa2:    ret
0x16aa3:    nop     dword ptr [rax + rax]
0x16aa8:    xor     ebx, ebx
0x16aaa:    jmp     0x16a28
0x16a28:
0x16aaf:    nop
0x16ab0:    cmp     byte ptr [r15 + 4], 0x20
0x16ab5:    jne     0x16a54
0x16ab7:    add     r12d, 1
0x16abb:    xor     ebp, ebp
0x16abd:    mov     ebx, 1
0x16ac2:    jmp     0x16a28
0x16a28:
0x16ac7:    nop     word ptr [rax + rax]
0x16ad0:    mov     rdi, r14
0x16ad3:    call    0x4940
0x16ad8:    mov     rdi, qword ptr [rip + 0x14b61]
0x16adf:    lea     rsi, [rsp + 0x10]
0x16ae4:    call    0x47f0
0x16ae9:    mov     rbp, qword ptr [rip + 0x13b70]
0x16af0:    test    r12d, r12d

```

```

0x16af3:      je      0x16b0a
0x16af5:      cmp     r12d, dword ptr [rsp + 4]
0x16afa:      je      0x16b48
0x16afc:      lea     rbp, [rip + 0x13b1d]
0x16b03:      mov     qword ptr [rip + 0x13b56], rbp
0x16b0a:      mov     rdi, rbp
0x16b0d:      call    0x49b0
0x16b12:      mov     rsi, r13
0x16b15:      mov     rdi, rbp
0x16b18:      mov     r8, rax
0x16b1b:      mov     eax, dword ptr [rip + 0x14b2f]
0x16b21:      add     eax, dword ptr [rip + 0x14b2d]
0x16b27:      lea     eax, [rax + r8 + 0x15]
0x16b2c:      mov     dword ptr [rip + 0x14b1a], eax
0x16b32:      call    0x4b30
0x16b37:      test    eax, eax
0x16b39:      setne   al
0x16b3c:      movzx   eax, al
0x16b3f:      jmp     0x16a7a
0x16a7a
0x16b44:      nop     dword ptr [rax]
0x16b48:      lea     rbp, [rip + 0x13adc]
0x16b4f:      mov     qword ptr [rip + 0x13b0a], rbp
0x16b56:      jmp     0x16b0a
0x16b0a
0x16b58:      nop     dword ptr [rax + rax]
0x16b60:      cmp     word ptr [r15 + 4], 0x7375
0x16b67:      jne     0x16a69
0x16b6d:      cmp     byte ptr [r15 + 6], 0x3a
0x16b72:      jne     0x16a69
0x16b78:      mov     rdi, r15
0x16b7b:      mov     esi, 0x3a
0x16b80:      call    0x4a00
0x16b85:      movsx   rbx, byte ptr [rax + 1]
0x16b8a:      lea     rdi, [rax + 1]
0x16b8e:      test    bl, bl
0x16b90:      je      0x16bf8
0x16b92:      mov     qword ptr [rsp + 8], rdi
0x16b97:      call    0x4f00
0x16b9c:      mov     rdi, qword ptr [rsp + 8]
0x16ba1:      mov     rax, qword ptr [rax]
0x16ba4:      jmp     0x16bbd
0x16bbd
0x16ba6:      nop     word ptr cs:[rax + rax]
0x16bb0:      movsx   rbx, byte ptr [rdi + 1]
0x16bb5:      add     rdi, 1
0x16bb9:      test    bl, bl
0x16bbb:      je      0x16bf8

```



```

0x16bbd:    test    byte ptr [rax + rbx*2 + 1], 0x20
0x16bc2:    jne     0x16bb0
0x16bc4:    mov     esi, 0x52
0x16bc9:    mov     ebp, 1
0x16bce:    mov     ebx, 1
0x16bd3:    call    0x4a00
0x16bd8:    test    rax, rax
0x16bdb:    je      0x16a28
0x16be1:    add     dword ptr [rsp + 4], 1
0x16be6:    mov     ebp, 1
0x16beb:    mov     ebx, 1
0x16bf0:    jmp     0x16a28
0x16a28:
0x16bf5:    nop     dword ptr [rax]
0x16bf8:    mov     ebx, 1
0x16bfd:    jmp     0x16a28
0x16a28:
0x16c02:    call    0x49d0
0x16c07:    nop     word ptr [rax + rax]
0x16c10:    endbr64
0x16c14:    sub     rsp, 0xa8
0x16c1b:    mov     edi, 0x1d
0x16c20:    mov     rax, qword ptr fs:[0x28]
0x16c29:    mov     qword ptr [rsp + 0x98], rax
0x16c31:    xor     eax, eax
0x16c33:    call    0x1d670
0x16c38:    lea     r9, [rip + 0x9950]
0x16c3f:    lea     r8, [rip + 0x991f]
0x16c46:    mov     rdx, -1
0x16c4d:    lea     rcx, [rip + 0xb219]
0x16c54:    mov     esi, 1
0x16c59:    mov     rdi, rax
0x16c5c:    mov     qword ptr [rip + 0x149fd], rax
0x16c63:    xor     eax, eax
0x16c65:    call    0x4f10
0x16c6a:    mov     rdi, qword ptr [rip + 0x149ef]
0x16c71:    call    0x49b0
0x16c76:    mov     rdi, qword ptr [rip + 0x14e1b]
0x16c7d:    mov     dword ptr [rip + 0x149d5], eax
0x16c83:    call    0x49b0
0x16c88:    mov     rdi, qword ptr [rip + 0x13d01]
0x16c8f:    mov     dword ptr [rip + 0x149bf], eax
0x16c95:    call    0x49b0
0x16c9a:    lea     rdi, [rip + 0x9999]
0x16ca1:    mov     dword ptr [rip + 0x149a9], eax
0x16ca7:    call    0x47e0
0x16cac:    mov     qword ptr [rip + 0x1498d], rax
0x16cb3:    test    rax, rax

```

```

0x16cb6:    je      0x16cc3
0x16cb8:    mov     rdi, rax
0x16cbb:    mov     rsi, rsp
0x16cbe:    call    0x18f40
0x16cc3:    xor     eax, eax
0x16cc5:    call    0x16980
0x16cca:    mov     rdi, qword ptr [rip + 0x1398f]
0x16cd1:    call    0x49b0
0x16cd6:    mov     r8, rax
0x16cd9:    mov     eax, dword ptr [rip + 0x14971]
0x16cdf:    add     eax, dword ptr [rip + 0x1496f]
0x16ce5:    lea     eax, [rax + r8 + 0x15]
0x16cea:    mov     dword ptr [rip + 0x1495c], eax
0x16cf0:    xor     eax, eax
0x16cf2:    call    0xc590
0x16cf7:    mov     qword ptr [rip + 0x1493a], rax
0x16cfe:    mov     rax, qword ptr [rsp + 0x98]
0x16d06:    sub     rax, qword ptr fs:[0x28]
0x16d0f:    jne     0x16d19
0x16d11:    add     rsp, 0xa8
0x16d18:    ret
0x16d19:    call    0x49d0
0x16d1e:    nop
0x16d20:    endbr64
0x16d24:    mov     rdi, qword ptr [rip + 0x1490d]
0x16d2b:    jmp     0xc5c0
0xc5c0
0x16d30:    endbr64
0x16d34:    mov     r8d, edi
0x16d37:    mov     rdi, qword ptr [rip + 0x148fa]
0x16d3e:    mov     edx, esi
0x16d40:    mov     ecx, 1
0x16d45:    xor     esi, esi
0x16d47:    jmp     0xc5e0
0xc5e0
0x16d4c:    nop     dword ptr [rax]
0x16d50:    endbr64
0x16d54:    push    r14
0x16d56:    push    r13
0x16d58:    push    r12
0x16d5a:    push    rbp
0x16d5b:    push    rbx
0x16d5c:    sub     rsp, 0x10
0x16d60:    mov     rax, qword ptr fs:[0x28]
0x16d69:    mov     qword ptr [rsp + 8], rax
0x16d6e:    xor     eax, eax
0x16d70:    lea     rbp, [rsp + 7]
0x16d75:    mov     rdi, rbp

```

```

0x16d78:      call    0xa6a0
0x16d7d:      mov     edx, dword ptr [rip + 0x138c1]
0x16d83:      mov     esi, dword ptr [rip + 0x138b7]
0x16d89:      mov     edi, dword ptr [rip + 0x138b9]
0x16d8f:      call    0xa570
0x16d94:      mov     rdi, qword ptr [rip + 0x1489d]
0x16d9b:      xor     edx, edx
0x16d9d:      xor     esi, esi
0x16d9f:      call    0xc680
0x16da4:      mov     edx, dword ptr [rip + 0x148ae]
0x16daa:      mov     rsi, qword ptr [rip + 0x148af]
0x16db1:      mov     rdi, qword ptr [rip + 0x14880]
0x16db8:      call    0xc600
0x16dbd:      mov     rax, qword ptr [rip + 0x14874]
0x16dc4:      mov     edi, dword ptr [rax + 0xc]
0x16dc7:      add     edi, 1
0x16dca:      movsxd  rdi, edi
0x16dcd:      call    0x1d670
0x16dd2:      mov     r13, rax
0x16dd5:      xor     eax, eax
0x16dd7:      call    0x16980
0x16ddc:      mov     r14, qword ptr [rip + 0x14855]
0x16de3:      mov     eax, dword ptr [rip + 0x1486f]
0x16de9:      mov     ecx, dword ptr [rip + 0x1485d]
0x16def:      mov     edx, dword ptr [r14 + 0xc]
0x16df3:      lea     esi, [rax + rcx + 2]
0x16df7:      cmp     esi, edx
0x16df9:      jl      0x16e58
0x16dfb:      cmp     eax, edx
0x16dfd:      jl      0x16e30
0x16dff:      mov     rdi, r13
0x16e02:      call    0x18d10
0x16e07:      mov     rdi, rbp
0x16e0a:      call    0xa6b0
0x16e0f:      mov     rax, qword ptr [rsp + 8]
0x16e14:      sub     rax, qword ptr fs:[0x28]
0x16e1d:      jne     0x16fc7
0x16e23:      add     rsp, 0x10
0x16e27:      pop     rbx
0x16e28:      pop     rbp
0x16e29:      pop     r12
0x16e2b:      pop     r13
0x16e2d:      pop     r14
0x16e2f:      ret
0x16e30:      sub     edx, eax
0x16e32:      mov     rdi, r13
0x16e35:      mov     esi, 0x20
0x16e3a:      mov     r12d, edx

```

```

0x16e3d:      movsxd  rdx, edx
0x16e40:      call   0x4a50
0x16e45:      mov     edx, r12d
0x16e48:      mov     rsi, r13
0x16e4b:      mov     rdi, r14
0x16e4e:      call   0xc600
0x16e53:      jmp     0x16dff
0x16dff:
0x16e55:      nop     dword ptr [rax]
0x16e58:      sub     edx, eax
0x16e5a:      mov     rdi, r13
0x16e5d:      mov     esi, 0x20
0x16e62:      mov     ebx, edx
0x16e64:      sub     ebx, ecx
0x16e66:      movsxd  rdx, ebx
0x16e69:      call   0x4a50
0x16e6e:      mov     edx, ebx
0x16e70:      mov     rsi, r13
0x16e73:      mov     rdi, r14
0x16e76:      call   0xc600
0x16e7b:      add     ebx, dword ptr [rip + 0x147d7]
0x16e81:      mov     rdi, qword ptr [rip + 0x147b0]
0x16e88:      xor     esi, esi
0x16e8a:      mov     edx, ebx
0x16e8c:      call   0xc680
0x16e91:      mov     rdi, qword ptr [rip + 0x147a0]
0x16e98:      mov     edx, 5
0x16e9d:      lea     rsi, [rip + 0x13793]
0x16ea4:      call   0xc600
0x16ea9:      mov     rdi, qword ptr [rip + 0x14788]
0x16eb0:      mov     esi, 0x20
0x16eb5:      call   0xc650
0x16eba:      mov     edi, dword ptr [rip + 0x1378c]
0x16ec0:      call   0xa4f0
0x16ec5:      mov     edx, dword ptr [rip + 0x14789]
0x16ecb:      mov     rsi, qword ptr [rip + 0x14bc6]
0x16ed2:      mov     rdi, qword ptr [rip + 0x1475f]
0x16ed9:      call   0xc600
0x16ede:      mov     rdi, qword ptr [rip + 0x14753]
0x16ee5:      mov     esi, 0x20
0x16eea:      call   0xc650
0x16eef:      mov     r12, qword ptr [rip + 0x1376a]
0x16ef6:      mov     rdi, r12
0x16ef9:      call   0x49b0
0x16efe:      mov     rdi, qword ptr [rip + 0x14733]
0x16f05:      mov     rsi, r12
0x16f08:      mov     edx, eax
0x16f0a:      call   0xc600

```

```

0x16f0f:      mov     rdi, qword ptr [rip + 0x14722]
0x16f16:      mov     esi, 0x20
0x16f1b:      call    0xc650
0x16f20:      mov     edi, dword ptr [rip + 0x1371a]
0x16f26:      call    0xa4f0
0x16f2b:      mov     rdi, qword ptr [rip + 0x14706]
0x16f32:      mov     edx, 4
0x16f37:      lea     rsi, [rip + 0x136f4]
0x16f3e:      call    0xc600
0x16f43:      mov     rdi, qword ptr [rip + 0x146ee]
0x16f4a:      mov     esi, 0x20
0x16f4f:      call    0xc650
0x16f54:      mov     edi, dword ptr [rip + 0x136f6]
0x16f5a:      call    0xa4f0
0x16f5f:      mov     edx, dword ptr [rip + 0x146eb]
0x16f65:      mov     rsi, qword ptr [rip + 0x13a24]
0x16f6c:      mov     rdi, qword ptr [rip + 0x146c5]
0x16f73:      call    0xc600
0x16f78:      mov     edi, dword ptr [rip + 0x136c2]
0x16f7e:      call    0xa4f0
0x16f83:      mov     rdi, qword ptr [rip + 0x146ae]
0x16f8a:      mov     esi, 0x20
0x16f8f:      call    0xc650
0x16f94:      xor     edi, edi
0x16f96:      xor     eax, eax
0x16f98:      call    0x16fd0
0x16f9d:      mov     rdi, qword ptr [rip + 0x14694]
0x16fa4:      xor     esi, esi
0x16fa6:      mov     eax, dword ptr [rdi + 0xc]
0x16fa9:      lea     edx, [rax - 1]
0x16fac:      call    0xc680
0x16fb1:      mov     rdi, qword ptr [rip + 0x14680]
0x16fb8:      mov     esi, 0x20
0x16fbd:      call    0xc650
0x16fc2:      jmp     0x16dff
0x16dff:
0x16fc7:      call    0x49d0
0x16fcc:      nop     dword ptr [rax]
0x16fd0:      endbr64
0x16fd4:      push    r12
0x16fd6:      push    rbp
0x16fd7:      push    rbx
0x16fd8:      mov     ebx, edi
0x16fda:      sub     rsp, 0x30
0x16fde:      mov     rax, qword ptr fs:[0x28]
0x16fe7:      mov     qword ptr [rsp + 0x28], rax
0x16fec:      xor     eax, eax
0x16fee:      call    0x7870

```

```

0x16ff3:    test    eax, eax
0x16ff5:    jne     0x17114
0x16ffb:    mov     eax, dword ptr [rip + 0x1464b]
0x17001:    mov     edx, dword ptr [rip + 0x14651]
0x17007:    lea     edx, [rdx + rax + 2]
0x1700b:    mov     rax, qword ptr [rip + 0x14626]
0x17012:    cmp     edx, dword ptr [rax + 0xc]
0x17015:    jge     0x17114
0x1701b:    test    ebx, ebx
0x1701d:    jne     0x17140
0x17023:    lea     r12, [rsp + 7]
0x17028:    xor     eax, eax
0x1702a:    call    0x18330
0x1702f:    mov     rdi, r12
0x17032:    mov     rbp, rax
0x17035:    call    0xa6a0
0x1703a:    lea     rsi, [rsp + 0xc]
0x1703f:    lea     rdi, [rsp + 8]
0x17044:    call    0xa4d0
0x17049:    xor     edi, edi
0x1704b:    call    0xa660
0x17050:    movsxd  r8, dword ptr [rbp + 8]
0x17054:    mov     r9d, dword ptr [rbp + 4]
0x17058:    lea     rbp, [rsp + 0x10]
0x1705d:    lea     rcx, [rip + 0xa0d4]
0x17064:    mov     esi, 1
0x17069:    mov     rdi, rbp
0x1706c:    mov     rax, r8
0x1706f:    imul    r8, r8, 0x2aaaaaab
0x17076:    cdq
0x17077:    sar     r8, 0x21
0x1707b:    sub     r8d, edx
0x1707e:    lea     edx, [r8 + r8*2]
0x17082:    mov     r8d, eax
0x17085:    shl     edx, 2
0x17088:    sub     r8d, edx
0x1708b:    mov     edx, 0xc
0x17090:    cmovbe  r8d, edx
0x17094:    cmp     eax, 0xc
0x17097:    mov     edx, 0x70
0x1709c:    mov     eax, 0x61
0x170a1:    cmovge  eax, edx
0x170a4:    sub     rsp, 8
0x170a8:    mov     edx, 0x10
0x170ad:    push    rax
0x170ae:    xor     eax, eax
0x170b0:    call    0x4f10
0x170b5:    mov     rdi, qword ptr [rip + 0x1457c]

```

```

0x170bc:    xor     esi, esi
0x170be:    mov     eax, dword ptr [rdi + 0xc]
0x170c1:    lea     edx, [rax - 7]
0x170c4:    call    0xc680
0x170c9:    mov     edx, dword ptr [rip + 0x13589]
0x170cf:    mov     esi, dword ptr [rip + 0x1357f]
0x170d5:    mov     edi, dword ptr [rip + 0x13581]
0x170db:    call    0xa570
0x170e0:    mov     rdi, rbp
0x170e3:    call    0x49b0
0x170e8:    mov     rdi, qword ptr [rip + 0x14549]
0x170ef:    mov     rsi, rbp
0x170f2:    mov     edx, eax
0x170f4:    call    0xc600
0x170f9:    mov     esi, dword ptr [rsp + 0x1c]
0x170fd:    mov     edi, dword ptr [rsp + 0x18]
0x17101:    call    0xa4b0
0x17106:    mov     rdi, r12
0x17109:    call    0xa6b0
0x1710e:    pop     rax
0x1710f:    pop     rdx
0x17110:    test    ebx, ebx
0x17112:    jne     0x17130
0x17114:    mov     rax, qword ptr [rsp + 0x28]
0x17119:    sub     rax, qword ptr fs:[0x28]
0x17122:    jne     0x1715b
0x17124:    add     rsp, 0x30
0x17128:    pop     rbx
0x17129:    pop     rbp
0x1712a:    pop     r12
0x1712c:    ret
0x1712d:    nop     dword ptr [rax]
0x17130:    xor     eax, eax
0x17132:    call    0xa820
0x17137:    jmp     0x17114
0x17114:
0x17139:    nop     dword ptr [rax]
0x17140:    xor     eax, eax
0x17142:    call    0x16980
0x17147:    test    eax, eax
0x17149:    je      0x17023
0x1714f:    xor     eax, eax
0x17151:    call    0x16d50
0x17156:    jmp     0x17023
0x17023:
0x1715b:    call    0x49d0
0x17160:    push    rbp
0x17161:    mov     esi, 0x400

```

```

0x17166:    push    rbx
0x17167:    lea     rbx, [rip + 0x14512]
0x1716e:    mov     rdi, rbx
0x17171:    sub     rsp, 8
0x17175:    mov     rdx, qword ptr [rip + 0x1490c]
0x1717c:    call    0x4b00
0x17181:    test    rax, rax
0x17184:    je      0x171d8
0x17186:    mov     rdi, rbx
0x17189:    call    0x49b0
0x1718e:    mov     rbp, rax
0x17191:    cmp     rax, 0x3ff
0x17197:    je      0x171e8
0x17199:    mov     esi, 0x23
0x1719e:    mov     rdi, rbx
0x171a1:    call    0x4a00
0x171a6:    test    rax, rax
0x171a9:    je      0x171c0
0x171ab:    mov     byte ptr [rax], 0
0x171ae:    mov     eax, 1
0x171b3:    add     rsp, 8
0x171b7:    pop     rbx
0x171b8:    pop     rbp
0x171b9:    ret
0x171ba:    nop     word ptr [rax + rax]
0x171c0:    cmp     byte ptr [rbx + rbp - 1], 0xa
0x171c5:    mov     eax, 1
0x171ca:    jne     0x171b3
0x171cc:    mov     byte ptr [rbx + rbp - 1], 0
0x171d1:    jmp     0x171b3
0x171b3:
0x171d3:    nop     dword ptr [rax + rax]
0x171d8:    add     rsp, 8
0x171dc:    xor     eax, eax
0x171de:    pop     rbx
0x171df:    pop     rbp
0x171e0:    ret
0x171e1:    nop     dword ptr [rax]
0x171e8:    mov     rcx, qword ptr [rip + 0x136f9]
0x171ef:    mov     rdi, qword ptr [rip + 0x1368a]
0x171f6:    lea     rdx, [rip + 0xaceb]
0x171fd:    xor     eax, eax
0x171ff:    mov     esi, 1
0x17204:    call    0x4e60
0x17209:    nop     dword ptr [rax]
0x17210:    mov     rdi, qword ptr [rip + 0x14871]
0x17217:    call    0x4a90
0x1721c:    cmp     eax, 0xa

```



```

0x1721f:      je      0x17199
0x17225:      cmp     eax, -1
0x17228:      jne     0x17210
0x1722a:      jmp     0x17199
0x17199
0x1722f:      nop
0x17230:      endbr64
0x17234:      test    rdi, rdi
0x17237:      je      0x17270
0x17239:      sub     rsp, 8
0x1723d:      lea     rsi, [rip + 0xad85]
0x17244:      call    0x4da0
0x17249:      mov     qword ptr [rip + 0x14838], rax
0x17250:      test    rax, rax
0x17253:      je      0x17269
0x17255:      mov     dword ptr [rip + 0x14821], 0xffffffff
0x1725f:      mov     eax, 1
0x17264:      add     rsp, 8
0x17268:      ret
0x17269:      xor     eax, eax
0x1726b:      add     rsp, 8
0x1726f:      ret
0x17270:      xor     eax, eax
0x17272:      ret
0x17273:      nop     word ptr cs:[rax + rax]
0x1727e:      nop
0x17280:      endbr64
0x17284:      mov     rdi, qword ptr [rip + 0x147fd]
0x1728b:      test    rdi, rdi
0x1728e:      je      0x17298
0x17290:      jmp     0x4940
0x4940
0x17295:      nop     dword ptr [rax]
0x17298:      ret
0x17299:      nop     dword ptr [rax]
0x172a0:      endbr64
0x172a4:      push    rbp
0x172a5:      xor     edx, edx
0x172a7:      xor     esi, esi
0x172a9:      lea     rbp, [rip + 0x143d0]
0x172b0:      push    rbx
0x172b1:      mov     rbx, rdi
0x172b4:      sub     rsp, 8
0x172b8:      mov     rdi, qword ptr [rip + 0x147c9]
0x172bf:      call    0x4c80
0x172c4:      jmp     0x172df
0x172df
0x172c6:      nop     word ptr cs:[rax + rax]

```

```

0x172d0:    mov     rsi, rbp
0x172d3:    mov     rdi, rbx
0x172d6:    call    0x4b30
0x172db:    test    eax, eax
0x172dd:    je      0x17300
0x172df:    xor     eax, eax
0x172e1:    call    0x17160
0x172e6:    test    eax, eax
0x172e8:    jne     0x172d0
0x172ea:    mov     eax, 0xffffffff
0x172ef:    mov     dword ptr [rip + 0x1478b], eax
0x172f5:    add     rsp, 8
0x172f9:    pop     rbx
0x172fa:    pop     rbp
0x172fb:    ret
0x172fc:    nop     dword ptr [rax]
0x17300:    mov     rdi, qword ptr [rip + 0x14781]
0x17307:    call    0x4b80
0x1730c:    mov     dword ptr [rip + 0x1476e], eax
0x17312:    add     rsp, 8
0x17316:    pop     rbx
0x17317:    pop     rbp
0x17318:    ret
0x17319:    nop     dword ptr [rax]
0x17320:    endbr64
0x17324:    push    r15
0x17326:    push    r14
0x17328:    push    r13
0x1732a:    mov     r13, rdi
0x1732d:    push    r12
0x1732f:    push    rbp
0x17330:    push    rbx
0x17331:    mov     ebx, ecx
0x17333:    sub     rsp, 0x438
0x1733a:    mov     qword ptr [rsp + 0x10], rsi
0x1733f:    mov     dword ptr [rsp + 0xc], edx
0x17343:    mov     rax, qword ptr fs:[0x28]
0x1734c:    mov     qword ptr [rsp + 0x428], rax
0x17354:    xor     eax, eax
0x17356:    test    ecx, ecx
0x17358:    jne     0x17618
0x1735e:    cmp     dword ptr [rsp + 0xc], 1
0x17363:    jne     0x175e0
0x17369:    mov     rax, qword ptr [rsp + 0x10]
0x1736e:    mov     qword ptr [rax], 0
0x17375:    lea     rbp, [rip + 0x14304]
0x1737c:    lea     r12, [rsp + 0x20]
0x17381:    lea     r14, [rip + 0xa092]

```

```

0x17388:    nop    dword ptr [rax + rax]
0x17390:    xor    eax, eax
0x17392:    call   0x17160
0x17397:    test   eax, eax
0x17399:    je     0x17600
0x1739f:    cmp    byte ptr [rbp], 0
0x173a3:    je     0x17600
0x173a9:    mov     rdx, r12
0x173ac:    mov     rsi, r14
0x173af:    mov     rdi, rbp
0x173b2:    xor     eax, eax
0x173b4:    mov     byte ptr [rsp + 0x20], 0
0x173b9:    call   0x4c60
0x173be:    call   0x4f00
0x173c3:    mov     r15, rax
0x173c6:    test    ebx, ebx
0x173c8:    jne     0x17410
0x173ca:    movsx   rcx, byte ptr [rsp + 0x20]
0x173d0:    mov     rax, qword ptr [rax]
0x173d3:    mov     byte ptr [rsp + 0x40], 0
0x173d8:    test    byte ptr [rax + rcx*2 + 1], 0x40
0x173dd:    jne     0x17433
0x173df:    mov     rax, qword ptr [rsp + 0x428]
0x173e7:    sub     rax, qword ptr fs:[0x28]
0x173f0:    jne     0x17657
0x173f6:    add     rsp, 0x438
0x173fd:    pop     rbx
0x173fe:    pop     rbp
0x173ff:    pop     r12
0x17401:    pop     r13
0x17403:    pop     r14
0x17405:    pop     r15
0x17407:    ret
0x17408:    nop    dword ptr [rax + rax]
0x17410:    movsx   rcx, byte ptr [rsp + 0x20]
0x17416:    mov     rax, qword ptr [rax]
0x17419:    test    byte ptr [rax + rcx*2 + 1], 0x40
0x1741e:    je      0x173df
0x17420:    mov     rsi, r12
0x17423:    mov     rdi, r13
0x17426:    call   0x4b30
0x1742b:    test    eax, eax
0x1742d:    jne     0x17390
0x17433:    mov     esi, 0x3d
0x17438:    mov     rdi, rbp
0x1743b:    call   0x4a00
0x17440:    test    rax, rax
0x17443:    je      0x174e3

```

```

0x17449:      cmp     byte ptr [rax + 1], 0
0x1744d:      je      0x174e3
0x17453:      mov     rdx, qword ptr [rsp + 0x10]
0x17458:      lea     rdi, [rax + 1]
0x1745c:      mov     qword ptr [rdx], rdi
0x1745f:      movzx   ecx, byte ptr [rax + 1]
0x17463:      test    cl, cl
0x17465:      je      0x1763f
0x1746b:      cmp     dword ptr [rsp + 0xc], 1
0x17470:      jle     0x1763f
0x17476:      mov     r8d, 1
0x1747c:      jmp     0x1748e
0x1748e:
0x1747e:      nop
0x17480:      mov     ecx, eax
0x17482:      cmp     r8d, dword ptr [rsp + 0xc]
0x17487:      jge     0x174d1
0x17489:      test    sil, sil
0x1748c:      je      0x174d1
0x1748e:      movzx   eax, byte ptr [rdi + 1]
0x17492:      test    al, al
0x17494:      setne   sil
0x17498:      add     rdi, 1
0x1749c:      cmp     cl, 0x3b
0x1749f:      jne     0x17480
0x174a1:      mov     byte ptr [rdi - 1], 0
0x174a5:      cmp     al, 0x3b
0x174a7:      je      0x17638
0x174ad:      test    sil, sil
0x174b0:      je      0x17638
0x174b6:      mov     rdx, qword ptr [rsp + 0x10]
0x174bb:      movsxd  rax, r8d
0x174be:      mov     qword ptr [rdx + rax*8], rdi
0x174c2:      movzx   ecx, byte ptr [rdi]
0x174c5:      test    cl, cl
0x174c7:      setne   sil
0x174cb:      add     r8d, 1
0x174cf:      jmp     0x17482
0x17482:
0x174d1:      mov     esi, 0x3b
0x174d6:      call    0x4a00
0x174db:      test    rax, rax
0x174de:      je      0x174ef
0x174e0:      mov     byte ptr [rax], 0
0x174e3:      mov     eax, dword ptr [rsp + 0xc]
0x174e7:      test    eax, eax
0x174e9:      jle     0x175c0
0x174ef:      mov     rsi, qword ptr [rsp + 0x10]

```

```

0x174f4:    xor     r8d, r8d
0x174f7:    jmp     0x17513
0x17513:
0x174f9:    nop     dword ptr [rax]
0x17500:    add     r8d, 1
0x17504:    add     rsi, 8
0x17508:    cmp     dword ptr [rsp + 0xc], r8d
0x1750d:    jle     0x175c0
0x17513:    mov     rbp, qword ptr [rsi]
0x17516:    test    rbp, rbp
0x17519:    je      0x17500
0x1751b:    mov     r14, qword ptr [r15]
0x1751e:    movsx   rax, byte ptr [rbp]
0x17523:    test    byte ptr [r14 + rax*2 + 1], 0x20
0x17529:    je      0x17547
0x1752b:    lea     rax, [rbp + 1]
0x1752f:    nop
0x17530:    mov     rbp, rax
0x17533:    mov     qword ptr [rsi], rax
0x17536:    add     rax, 1
0x1753a:    movsx   rcx, byte ptr [rbp]
0x1753f:    test    byte ptr [r14 + rcx*2 + 1], 0x20
0x17545:    jne     0x17530
0x17547:    mov     rdi, rbp
0x1754a:    mov     qword ptr [rsp + 0x18], rsi
0x1754f:    mov     dword ptr [rsp + 0x10], r8d
0x17554:    call    0x49b0
0x17559:    mov     r8d, dword ptr [rsp + 0x10]
0x1755e:    mov     rsi, qword ptr [rsp + 0x18]
0x17563:    lea     rax, [rbp + rax - 1]
0x17568:    movsx   rcx, byte ptr [rax]
0x1756c:    test    byte ptr [r14 + rcx*2 + 1], 0x20
0x17572:    jne     0x17591
0x17574:    jmp     0x17596
0x17596:
0x17576:    nop     word ptr cs:[rax + rax]
0x17580:    movsx   rcx, byte ptr [rax - 1]
0x17585:    sub     rax, 1
0x17589:    test    byte ptr [r14 + rcx*2 + 1], 0x20
0x1758f:    je      0x17596
0x17591:    cmp     rax, rbp
0x17594:    jae     0x17580
0x17596:    mov     byte ptr [rax + 1], 0
0x1759a:    mov     rax, qword ptr [rsi]
0x1759d:    cmp     byte ptr [rax], 0
0x175a0:    jne     0x17500
0x175a6:    mov     qword ptr [rsi], 0
0x175ad:    add     r8d, 1

```

```

0x175b1:      add     rsi, 8
0x175b5:      cmp     dword ptr [rsp + 0xc], r8d
0x175ba:      jg       0x17513
0x175c0:      test    ebx, ebx
0x175c2:      jne     0x173df
0x175c8:      mov     rsi, r12
0x175cb:      mov     rdi, r13
0x175ce:      call    0x48b0
0x175d3:      jmp     0x173df
0x173df
0x175d8:      nop     dword ptr [rax + rax]
0x175e0:      movsxd  rdx, dword ptr [rsp + 0xc]
0x175e5:      mov     rdi, qword ptr [rsp + 0x10]
0x175ea:      xor     esi, esi
0x175ec:      shl     rdx, 3
0x175f0:      call    0x4a50
0x175f5:      jmp     0x17375
0x17375
0x175fa:      nop     word ptr [rax + rax]
0x17600:      test    ebx, ebx
0x17602:      jne     0x173df
0x17608:      mov     byte ptr [r13], 0
0x1760d:      jmp     0x173df
0x173df
0x17612:      nop     word ptr [rax + rax]
0x17618:      movsxd  rsi, dword ptr [rip + 0x14461]
0x1761f:      mov     rdi, qword ptr [rip + 0x14462]
0x17626:      xor     edx, edx
0x17628:      call    0x4c80
0x1762d:      jmp     0x1735e
0x1735e
0x17632:      nop     word ptr [rax + rax]
0x17638:      mov     ecx, eax
0x1763a:      jmp     0x174cb
0x174cb
0x1763f:      mov     esi, 0x3b
0x17644:      call    0x4a00
0x17649:      test    rax, rax
0x1764c:      jne     0x174e0
0x17652:      jmp     0x174e3
0x174e3
0x17657:      call    0x49d0
0x1765c:      nop     dword ptr [rax]
0x17660:      endbr64
0x17664:      push    r12
0x17666:      mov     edi, 0xf
0x1766b:      lea     r12, [rip + 0xa8b4]
0x17672:      call    0x1d670

```

```

0x17677:      mov     rcx, r12
0x1767a:      lea     r9, [rip + 0xa895]
0x17681:      lea     r8, [rip + 0xa893]
0x17688:      mov     rdi, rax
0x1768b:      mov     rdx, -1
0x17692:      mov     esi, 1
0x17697:      mov     qword ptr [rip + 0x14402], rax
0x1769e:      xor     eax, eax
0x176a0:      call    0x4f10
0x176a5:      mov     edi, 0x17
0x176aa:      call    0x1d670
0x176af:      mov     rcx, r12
0x176b2:      pop     r12
0x176b4:      lea     r9, [rip + 0xa870]
0x176bb:      mov     qword ptr [rip + 0x143e6], rax
0x176c2:      mov     rdi, rax
0x176c5:      xor     eax, eax
0x176c7:      lea     r8, [rip + 0xa84d]
0x176ce:      mov     rdx, -1
0x176d5:      mov     esi, 1
0x176da:      jmp     0x4f10
0x4f10
0x176df:      nop
0x176e0:      endbr64
0x176e4:      push    rbp
0x176e5:      mov     rbp, rdi
0x176e8:      xor     edi, edi
0x176ea:      push    rbx
0x176eb:      sub     rsp, 0x18
0x176ef:      mov     rax, qword ptr fs:[0x28]
0x176f8:      mov     qword ptr [rsp + 8], rax
0x176fd:      xor     eax, eax
0x176ff:      call    0x4be0
0x17704:      mov     rdi, rsp
0x17707:      mov     qword ptr [rsp], rax
0x1770b:      call    0x4840
0x17710:      mov     rbx, rax
0x17713:      call    0x4920
0x17718:      movsxd  rcx, dword ptr [rbx + 0x10]
0x1771c:      sub     rsp, 8
0x17720:      lea     rdx, [rip + 0x12fd9]
0x17727:      mov     r9d, dword ptr [rbx + 8]
0x1772b:      mov     r8d, dword ptr [rbx + 0xc]
0x1772f:      mov     esi, 1
0x17734:      mov     rcx, qword ptr [rdx + rcx*8]
0x17738:      push    rbp
0x17739:      lea     rdx, [rip + 0xa8d0]
0x17740:      push    rax

```

```

0x17741:    mov     rdi, qword ptr [rip + 0x13138]
0x17748:    push    qword ptr [rip + 0x1319a]
0x1774e:    mov     eax, dword ptr [rbx]
0x17750:    push    rax
0x17751:    mov     eax, dword ptr [rbx + 4]
0x17754:    push    rax
0x17755:    xor     eax, eax
0x17757:    call    0x4e60
0x1775c:    add     rsp, 0x30
0x17760:    mov     rax, qword ptr [rsp + 8]
0x17765:    sub     rax, qword ptr fs:[0x28]
0x1776e:    jne     0x17777
0x17770:    add     rsp, 0x18
0x17774:    pop     rbx
0x17775:    pop     rbp
0x17776:    ret
0x17777:    call    0x49d0
0x1777c:    nop     dword ptr [rax]
0x17780:    endbr64
0x17784:    push    rbp
0x17785:    xor     eax, eax
0x17787:    mov     ebp, edi
0x17789:    call    0x8ad0
0x1778e:    cmp     ebp, 0xb
0x17791:    je      0x177d3
0x17793:    jg      0x177c0
0x17795:    lea     eax, [rbp - 1]
0x17798:    cmp     eax, 1
0x1779b:    ja      0x17802
0x1779d:    cmp     ebp, 1
0x177a0:    lea     rdi, [rip + 0xa791]
0x177a7:    lea     rax, [rip + 0xa78e]
0x177ae:    cmovne  rdi, rax
0x177b2:    call    0x176e0
0x177b7:    mov     edi, ebp
0x177b9:    call    0x4e20
0x177be:    nop
0x177c0:    cmp     ebp, 0xf
0x177c3:    jne     0x17802
0x177c5:    lea     rdi, [rip + 0x8dbe]
0x177cc:    call    0x176e0
0x177d1:    jmp     0x177b7
0x177b7:
0x177d3:    lea     rdi, [rip + 0xa766]
0x177da:    call    0x176e0
0x177df:    mov     rcx, qword ptr [rip + 0x13102]
0x177e6:    mov     rdi, qword ptr [rip + 0x13093]
0x177ed:    lea     rdx, [rip + 0xa88c]

```



```

0x177f4:      xor     eax, eax
0x177f6:      mov     esi, 1
0x177fb:      call    0x4e60
0x17800:      jmp     0x177b7
0x177b7:
0x17802:      mov     rcx, qword ptr [rip + 0x130df]
0x17809:      mov     rdi, qword ptr [rip + 0x13070]
0x17810:      mov     r8d, ebp
0x17813:      xor     eax, eax
0x17815:      lea     rdx, [rip + 0xa82c]
0x1781c:      mov     esi, 1
0x17821:      call    0x4e60
0x17826:      jmp     0x177df
0x177df:
0x17828:      nop     dword ptr [rax + rax]
0x17830:      endbr64
0x17834:      push    rbx
0x17835:      mov     rcx, qword ptr [rip + 0x130ac]
0x1783c:      mov     rbx, rdi
0x1783f:      mov     esi, 1
0x17844:      mov     rdi, qword ptr [rip + 0x13035]
0x1784b:      lea     rdx, [rip + 0xa6f3]
0x17852:      xor     eax, eax
0x17854:      call    0x4e60
0x17859:      mov     r8, rbx
0x1785c:      mov     rcx, qword ptr [rip + 0x13085]
0x17863:      pop     rbx
0x17864:      mov     rdi, qword ptr [rip + 0x13015]
0x1786b:      lea     rdx, [rip + 0xa836]
0x17872:      mov     esi, 1
0x17877:      xor     eax, eax
0x17879:      jmp     0x4e60
0x4e60:
0x1787e:      nop
0x17880:      endbr64
0x17884:      sub     rsp, 8
0x17888:      mov     r8, rdi
0x1788b:      mov     rcx, qword ptr [rip + 0x13056]
0x17892:      xor     eax, eax
0x17894:      mov     rdi, qword ptr [rip + 0x12fe5]
0x1789b:      lea     rdx, [rip + 0xa836]
0x178a2:      mov     esi, 1
0x178a7:      call    0x4e60
0x178ac:      mov     rcx, qword ptr [rip + 0x13035]
0x178b3:      mov     esi, 1
0x178b8:      xor     eax, eax
0x178ba:      mov     rdi, qword ptr [rip + 0x12fbf]
0x178c1:      lea     rdx, [rip + 0xa840]

```

```

0x178c8:      call    0x4e60
0x178cd:      mov     rcx, qword ptr [rip + 0x13014]
0x178d4:      mov     esi, 1
0x178d9:      xor     eax, eax
0x178db:      mov     rdi, qword ptr [rip + 0x12f9e]
0x178e2:      lea     rdx, [rip + 0xa84f]
0x178e9:      call    0x4e60
0x178ee:      mov     rcx, qword ptr [rip + 0x12ff3]
0x178f5:      mov     esi, 1
0x178fa:      xor     eax, eax
0x178fc:      mov     rdi, qword ptr [rip + 0x12f7d]
0x17903:      lea     rdx, [rip + 0xa86e]
0x1790a:      call    0x4e60
0x1790f:      mov     rcx, qword ptr [rip + 0x12fd2]
0x17916:      mov     esi, 1
0x1791b:      xor     eax, eax
0x1791d:      mov     rdi, qword ptr [rip + 0x12f5c]
0x17924:      lea     rdx, [rip + 0xa885]
0x1792b:      call    0x4e60
0x17930:      mov     rcx, qword ptr [rip + 0x12fb1]
0x17937:      mov     esi, 1
0x1793c:      xor     eax, eax
0x1793e:      mov     rdi, qword ptr [rip + 0x12f3b]
0x17945:      lea     rdx, [rip + 0xa8a4]
0x1794c:      call    0x4e60
0x17951:      mov     rcx, qword ptr [rip + 0x12f90]
0x17958:      mov     esi, 1
0x1795d:      xor     eax, eax
0x1795f:      mov     rdi, qword ptr [rip + 0x12f1a]
0x17966:      lea     rdx, [rip + 0xa8c3]
0x1796d:      call    0x4e60
0x17972:      mov     rcx, qword ptr [rip + 0x12f6f]
0x17979:      mov     esi, 1
0x1797e:      xor     eax, eax
0x17980:      mov     rdi, qword ptr [rip + 0x12ef9]
0x17987:      lea     rdx, [rip + 0xa8ea]
0x1798e:      add     rsp, 8
0x17992:      jmp     0x4e60
0x4e60
0x17997:      nop     word ptr [rax + rax]
0x179a0:      endbr64
0x179a4:      push    r12
0x179a6:      lea     r12, [rip + 0x12d33]
0x179ad:      push    rbp
0x179ae:      sub     rsp, 8
0x179b2:      mov     rdi, qword ptr [rip + 0x140ef]
0x179b9:      call    0x49b0
0x179be:      mov     rdi, r12

```

```

0x179c1:    mov     rbp, rax
0x179c4:    call   0x49b0
0x179c9:    lea     rdi, [rbp + rax + 9]
0x179ce:    call   0x1d670
0x179d3:    mov     rsi, qword ptr [rip + 0x140ce]
0x179da:    mov     rdi, rax
0x179dd:    mov     rbp, rax
0x179e0:    call   0x4980
0x179e5:    mov     rsi, r12
0x179e8:    mov     rdi, rax
0x179eb:    call   0x4980
0x179f0:    mov     edx, 0x6e6f
0x179f5:    mov     rdi, rbp
0x179f8:    mov     dword ptr [rax], 0x6d6d6f63
0x179fe:    mov     word ptr [rax + 4], dx
0x17a02:    mov     byte ptr [rax + 6], 0
0x17a06:    call   0x17230
0x17a0b:    test    eax, eax
0x17a0d:    je      0x17a17
0x17a0f:    add     rsp, 8
0x17a13:    pop     rbp
0x17a14:    pop     r12
0x17a16:    ret
0x17a17:    mov     rdi, rbp
0x17a1a:    call   0x17830
0x17a1f:    mov     edi, 1
0x17a24:    call   0x4e20
0x17a29:    nop     dword ptr [rax]
0x17a30:    endbr64
0x17a34:    push    r13
0x17a36:    push    r12
0x17a38:    lea     r12, [rip + 0x12c91]
0x17a3f:    push    rbp
0x17a40:    push    rbx
0x17a41:    sub     rsp, 8
0x17a45:    mov     rdi, qword ptr [rip + 0x12ea4]
0x17a4c:    call   0x49b0
0x17a51:    mov     rdi, r12
0x17a54:    mov     rbx, rax
0x17a57:    call   0x49b0
0x17a5c:    mov     rdi, qword ptr [rip + 0x12eed]
0x17a63:    mov     rbp, rax
0x17a66:    call   0x49b0
0x17a6b:    lea     rdi, [rbx + rbp + 2]
0x17a70:    add     rdi, rax
0x17a73:    call   0x1d670
0x17a78:    mov     rsi, qword ptr [rip + 0x12e71]
0x17a7f:    mov     rdi, rax

```

```

0x17a82:    mov     rbp, rax
0x17a85:    call   0x4980
0x17a8a:    mov     rsi, r12
0x17a8d:    mov     rdi, rax
0x17a90:    call   0x4980
0x17a95:    mov     rsi, qword ptr [rip + 0x12eb4]
0x17a9c:    mov     rdi, rax
0x17a9f:    call   0x48b0
0x17aa4:    mov     rdi, rbp
0x17aa7:    call   0x17230
0x17aac:    test    eax, eax
0x17aae:    jne     0x17b2a
0x17ab0:    mov     rdi, rbp
0x17ab3:    lea     r12, [rip + 0x12c06]
0x17aba:    call   0x18d10
0x17abf:    mov     rdi, qword ptr [rip + 0x12e2a]
0x17ac6:    call   0x49b0
0x17acb:    mov     rdi, r12
0x17ace:    mov     rbp, rax
0x17ad1:    call   0x49b0
0x17ad6:    mov     rdi, qword ptr [rip + 0x12e73]
0x17add:    mov     rbx, rax
0x17ae0:    call   0x49b0
0x17ae5:    lea     rdi, [rbp + rbx + 2]
0x17aea:    add     rdi, rax
0x17aed:    call   0x1d670
0x17af2:    mov     rsi, qword ptr [rip + 0x12df7]
0x17af9:    mov     rdi, rax
0x17afc:    mov     rbp, rax
0x17aff:    call   0x4980
0x17b04:    mov     rsi, r12
0x17b07:    mov     rdi, rax
0x17b0a:    call   0x4980
0x17b0f:    mov     rsi, qword ptr [rip + 0x12e3a]
0x17b16:    mov     rdi, rax
0x17b19:    call   0x48b0
0x17b1e:    mov     rdi, rbp
0x17b21:    call   0x17230
0x17b26:    test    eax, eax
0x17b28:    je      0x17b50
0x17b2a:    mov     rdi, rbp
0x17b2d:    mov     r12d, 1
0x17b33:    call   0x18d10
0x17b38:    add     rsp, 8
0x17b3c:    mov     eax, r12d
0x17b3f:    pop     rbx
0x17b40:    pop     rbp
0x17b41:    pop     r12

```

```

0x17b43:    pop     r13
0x17b45:    ret
0x17b46:    nop     word ptr cs:[rax + rax]
0x17b50:    mov     rdi, rbp
0x17b53:    lea     r13, [rip + 0x12b86]
0x17b5a:    call    0x18d10
0x17b5f:    mov     rdi, qword ptr [rip + 0x13f42]
0x17b66:    call    0x49b0
0x17b6b:    mov     rdi, r13
0x17b6e:    mov     rbp, rax
0x17b71:    call    0x49b0
0x17b76:    mov     rdi, qword ptr [rip + 0x12dd3]
0x17b7d:    mov     rbx, rax
0x17b80:    call    0x49b0
0x17b85:    lea     rdi, [rbp + rbx + 2]
0x17b8a:    add     rdi, rax
0x17b8d:    call    0x1d670
0x17b92:    mov     rsi, qword ptr [rip + 0x13f0f]
0x17b99:    mov     rdi, rax
0x17b9c:    mov     rbp, rax
0x17b9f:    call    0x4980
0x17ba4:    mov     rsi, r13
0x17ba7:    mov     rdi, rax
0x17baa:    call    0x4980
0x17baf:    mov     rsi, qword ptr [rip + 0x12d9a]
0x17bb6:    mov     rdi, rax
0x17bb9:    call    0x48b0
0x17bbe:    mov     rdi, rbp
0x17bc1:    call    0x17230
0x17bc6:    mov     r12d, eax
0x17bc9:    test    eax, eax
0x17bcb:    jne     0x17b2a
0x17bd1:    xor     eax, eax
0x17bd3:    mov     rdi, rbp
0x17bd6:    call    0x17880
0x17bdb:    mov     rdi, rbp
0x17bde:    call    0x18d10
0x17be3:    mov     rdi, qword ptr [rip + 0x13ebe]
0x17bea:    call    0x49b0
0x17bef:    mov     rdi, r13
0x17bf2:    mov     rbx, rax
0x17bf5:    call    0x49b0
0x17bfa:    lea     rdi, [rbx + rax + 0xa]
0x17bff:    call    0x1d670
0x17c04:    mov     rsi, qword ptr [rip + 0x13e9d]
0x17c0b:    mov     rdi, rax
0x17c0e:    mov     rbp, rax
0x17c11:    call    0x4980

```

```

0x17c16:    mov     rsi, r13
0x17c19:    mov     rdi, rax
0x17c1c:    call    0x4980
0x17c21:    mov     rdi, rbp
0x17c24:    movabs  rdx, 0x636972656e6567
0x17c2e:    mov     qword ptr [rax], rdx
0x17c31:    call    0x17230
0x17c36:    test    eax, eax
0x17c38:    jne     0x17b38
0x17c3e:    mov     rdi, rbp
0x17c41:    call    0x17830
0x17c46:    mov     edi, 1
0x17c4b:    call    0x4e20
0x17c50:    endbr64
0x17c54:    push    rbp
0x17c55:    push    rbx
0x17c56:    mov     rbx, rdi
0x17c59:    sub     rsp, 8
0x17c5d:    call    0x172a0
0x17c62:    cmp     eax, -1
0x17c65:    je      0x17c70
0x17c67:    add     rsp, 8
0x17c6b:    pop     rbx
0x17c6c:    pop     rbp
0x17c6d:    ret
0x17c6e:    nop
0x17c70:    mov     edx, 7
0x17c75:    lea     rsi, [rip + 0xa2e5]
0x17c7c:    mov     rdi, rbx
0x17c7f:    call    0x4890
0x17c84:    test    eax, eax
0x17c86:    jne     0x17cd4
0x17c88:    lea     rdi, [rbx + 5]
0x17c8c:    call    0x49b0
0x17c91:    lea     rdi, [rax + 6]
0x17c95:    call    0x1d670
0x17c9a:    lea     rsi, [rbx + 7]
0x17c9e:    mov     dword ptr [rax], 0x5449475b
0x17ca4:    mov     rbp, rax
0x17ca7:    lea     rdi, [rax + 5]
0x17cab:    mov     byte ptr [rax + 4], 0x2d
0x17caf:    call    0x48b0
0x17cb4:    mov     rdi, rbp
0x17cb7:    call    0x172a0
0x17cbc:    mov     rdi, rbp
0x17cbf:    cmp     eax, -1
0x17cc2:    je      0x17ccf
0x17cc4:    add     rsp, 8

```

```

0x17cc8:      pop     rbx
0x17cc9:      pop     rbp
0x17cca:      jmp     0x18d10
0x18d10:
0x17ccf:      call    0x18d10
0x17cd4:      mov     rdi, qword ptr [rip + 0x12ba5]
0x17cdb:      mov     rcx, qword ptr [rip + 0x12c06]
0x17ce2:      mov     r8, rbx
0x17ce5:      xor     eax, eax
0x17ce7:      lea     rdx, [rip + 0xa5ba]
0x17cee:      mov     esi, 1
0x17cf3:      call    0x4e60
0x17cf8:      mov     edi, 1
0x17cfd:      call    0x4e20
0x17d02:      nop     word ptr cs:[rax + rax]
0x17d0d:      nop     dword ptr [rax]
0x17d10:      endbr64
0x17d14:      push    rbx
0x17d15:      mov     ecx, 1
0x17d1a:      mov     ebx, esi
0x17d1c:      mov     edx, 1
0x17d21:      sub     rsp, 0x10
0x17d25:      mov     rax, qword ptr fs:[0x28]
0x17d2e:      mov     qword ptr [rsp + 8], rax
0x17d33:      xor     eax, eax
0x17d35:      mov     rsi, rsp
0x17d38:      call    0x17320
0x17d3d:      mov     rdi, qword ptr [rsp]
0x17d41:      mov     eax, ebx
0x17d43:      test    rdi, rdi
0x17d46:      je      0x17d54
0x17d48:      mov     edx, 0xa
0x17d4d:      xor     esi, esi
0x17d4f:      call    0x4ba0
0x17d54:      mov     rdx, qword ptr [rsp + 8]
0x17d59:      sub     rdx, qword ptr fs:[0x28]
0x17d62:      jne     0x17d6a
0x17d64:      add     rsp, 0x10
0x17d68:      pop     rbx
0x17d69:      ret
0x17d6a:      call    0x49d0
0x17d6f:      nop
0x17d70:      endbr64
0x17d74:      push    r15
0x17d76:      push    r14
0x17d78:      mov     r14, rdi
0x17d7b:      push    r13
0x17d7d:      mov     r13d, ecx

```

```

0x17d80:    mov     ecx, 1
0x17d85:    push    r12
0x17d87:    mov     r12d, r13d
0x17d8a:    push    rbp
0x17d8b:    mov     ebp, edx
0x17d8d:    mov     edx, 1
0x17d92:    push    rbx
0x17d93:    mov     rbx, rsi
0x17d96:    sub     rsp, 0x18
0x17d9a:    mov     rax, qword ptr fs:[0x28]
0x17da3:    mov     qword ptr [rsp + 8], rax
0x17da8:    xor     eax, eax
0x17daa:    mov     rsi, rsp
0x17dad:    call    0x17320
0x17db2:    mov     r15, qword ptr [rsp]
0x17db6:    test    r15, r15
0x17db9:    je      0x17de4
0x17dbb:    test    ebp, ebp
0x17dbd:    jle     0x17e3c
0x17dbf:    xor     r12d, r12d
0x17dc2:    jmp     0x17dd5
0x17dd5:
0x17dc4:    nop     dword ptr [rax]
0x17dc8:    add     r12d, 1
0x17dcc:    add     rbx, 8
0x17dd0:    cmp     ebp, r12d
0x17dd3:    je      0x17e10
0x17dd5:    mov     rsi, qword ptr [rbx]
0x17dd8:    mov     rdi, r15
0x17ddb:    call    0x4b30
0x17de0:    test    eax, eax
0x17de2:    jne     0x17dc8
0x17de4:    mov     rax, qword ptr [rsp + 8]
0x17de9:    sub     rax, qword ptr fs:[0x28]
0x17df2:    jne     0x17e43
0x17df4:    add     rsp, 0x18
0x17df8:    mov     eax, r12d
0x17dfb:    pop     rbx
0x17dfc:    pop     rbp
0x17dfd:    pop     r12
0x17dff:    pop     r13
0x17e01:    pop     r14
0x17e03:    pop     r15
0x17e05:    ret
0x17e06:    nop     word ptr cs:[rax + rax]
0x17e10:    mov     rcx, qword ptr [rip + 0x12ad1]
0x17e17:    mov     r9, r15
0x17e1a:    mov     r8, r14

```



```

0x17e1d:    xor     eax, eax
0x17e1f:    mov     rdi, qword ptr [rip + 0x12a5a]
0x17e26:    lea     rdx, [rip + 0xa13c]
0x17e2d:    mov     esi, 1
0x17e32:    mov     r12d, r13d
0x17e35:    call    0x4e60
0x17e3a:    jmp     0x17de4
0x17de4:
0x17e3c:    je      0x17e10
0x17e3e:    xor     r12d, r12d
0x17e41:    jmp     0x17de4
0x17de4:
0x17e43:    call    0x49d0
0x17e48:    nop     dword ptr [rax + rax]
0x17e50:    endbr64
0x17e54:    push    r12
0x17e56:    mov     ecx, 1
0x17e5b:    mov     edx, 1
0x17e60:    mov     r12, rdi
0x17e63:    push    rbp
0x17e64:    push    rbx
0x17e65:    mov     ebx, esi
0x17e67:    sub     rsp, 0x10
0x17e6b:    mov     rax, qword ptr fs:[0x28]
0x17e74:    mov     qword ptr [rsp + 8], rax
0x17e79:    xor     eax, eax
0x17e7b:    mov     rsi, rsp
0x17e7e:    call    0x17320
0x17e83:    mov     rbp, qword ptr [rsp]
0x17e87:    mov     eax, ebx
0x17e89:    test    rbp, rbp
0x17e8c:    je      0x17ea7
0x17e8e:    cmp     byte ptr [rbp], 0x4f
0x17e92:    je      0x17ec0
0x17e94:    lea     rsi, [rip + 0x8e4b]
0x17e9b:    mov     rdi, rbp
0x17e9e:    call    0x4b30
0x17ea3:    test    eax, eax
0x17ea5:    jne     0x17ed8
0x17ea7:    mov     rdx, qword ptr [rsp + 8]
0x17eac:    sub     rdx, qword ptr fs:[0x28]
0x17eb5:    jne     0x17f03
0x17eb7:    add     rsp, 0x10
0x17ebb:    pop     rbx
0x17ebc:    pop     rbp
0x17ebd:    pop     r12
0x17ebf:    ret
0x17ec0:    cmp     byte ptr [rbp + 1], 0x4e

```

```

0x17ec4:    jne     0x17e94
0x17ec6:    cmp     byte ptr [rbp + 2], 0
0x17eca:    mov     eax, 1
0x17ecf:    je      0x17ea7
0x17ed1:    jmp     0x17e94
0x17e94:
0x17ed3:    nop     dword ptr [rax + rax]
0x17ed8:    mov     rcx, qword ptr [rip + 0x12a09]
0x17edf:    mov     r9, rbp
0x17ee2:    mov     r8, r12
0x17ee5:    xor     eax, eax
0x17ee7:    mov     rdi, qword ptr [rip + 0x12992]
0x17eee:    lea     rdx, [rip + 0xa074]
0x17ef5:    mov     esi, 1
0x17efa:    call    0x4e60
0x17eff:    mov     eax, ebx
0x17f01:    jmp     0x17ea7
0x17ea7:
0x17f03:    call    0x49d0
0x17f08:    nop     dword ptr [rax + rax]
0x17f10:    endbr64
0x17f14:    push    r12
0x17f16:    mov     ecx, 1
0x17f1b:    mov     r12, rsi
0x17f1e:    mov     edx, 1
0x17f23:    sub     rsp, 0x10
0x17f27:    mov     rax, qword ptr fs:[0x28]
0x17f30:    mov     qword ptr [rsp + 8], rax
0x17f35:    xor     eax, eax
0x17f37:    mov     rsi, rsp
0x17f3a:    call    0x17320
0x17f3f:    mov     rdi, qword ptr [rsp]
0x17f43:    test    rdi, rdi
0x17f46:    je      0x17f50
0x17f48:    call    0x1d7a0
0x17f4d:    mov     r12, rax
0x17f50:    mov     rax, qword ptr [rsp + 8]
0x17f55:    sub     rax, qword ptr fs:[0x28]
0x17f5e:    jne     0x17f6a
0x17f60:    add     rsp, 0x10
0x17f64:    mov     rax, r12
0x17f67:    pop     r12
0x17f69:    ret
0x17f6a:    call    0x49d0
0x17f6f:    nop
0x17f70:    endbr64
0x17f74:    push    r14
0x17f76:    push    r13

```

```

0x17f78:    push    r12
0x17f7a:    push    rbp
0x17f7b:    push    rbx
0x17f7c:    sub     rsp, 0x10
0x17f80:    mov     rax, qword ptr fs:[0x28]
0x17f89:    mov     qword ptr [rsp + 8], rax
0x17f8e:    xor     eax, eax
0x17f90:    test    edx, edx
0x17f92:    jle     0x17fe8
0x17f94:    lea     eax, [rdx - 1]
0x17f97:    mov     rbx, rsi
0x17f9a:    mov     rbp, rdi
0x17f9d:    mov     r12, rsp
0x17fa0:    lea     r13, [rsi + rax*8 + 8]
0x17fa5:    lea     r14, [rip + 0x9fbd]
0x17fac:    nop     dword ptr [rax]
0x17fb0:    mov     rdi, qword ptr [rbx]
0x17fb3:    mov     ecx, 1
0x17fb8:    mov     edx, 1
0x17fbd:    mov     rsi, r12
0x17fc0:    call    0x17320
0x17fc5:    mov     rdi, qword ptr [rsp]
0x17fc9:    test    rdi, rdi
0x17fcc:    je      0x17fdb
0x17fce:    call    0xbd00
0x17fd3:    cmp     eax, -1
0x17fd6:    je      0x18008
0x17fd8:    mov     dword ptr [rbp], eax
0x17fdb:    add     rbx, 8
0x17fdf:    add     rbp, 4
0x17fe3:    cmp     rbx, r13
0x17fe6:    jne     0x17fb0
0x17fe8:    mov     rax, qword ptr [rsp + 8]
0x17fed:    sub     rax, qword ptr fs:[0x28]
0x17ff6:    jne     0x1802e
0x17ff8:    add     rsp, 0x10
0x17ffc:    pop     rbx
0x17ffd:    pop     rbp
0x17ffe:    pop     r12
0x18000:    pop     r13
0x18002:    pop     r14
0x18004:    ret
0x18005:    nop     dword ptr [rax]
0x18008:    mov     r9, qword ptr [rsp]
0x1800c:    mov     r8, qword ptr [rbx]
0x1800f:    mov     rdx, r14
0x18012:    mov     esi, 1
0x18017:    mov     rcx, qword ptr [rip + 0x128ca]

```

```

0x1801e:    mov     rdi, qword ptr [rip + 0x1285b]
0x18025:    xor     eax, eax
0x18027:    call    0x4e60
0x1802c:    jmp     0x17fdb
0x17fdb:
0x1802e:    call    0x49d0
0x18033:    nop     word ptr cs:[rax + rax]
0x1803e:    nop
0x18040:    endbr64
0x18044:    push    r12
0x18046:    cmp     byte ptr [rdi], 0x2f
0x18049:    mov     r12, rdi
0x1804c:    lea     rax, [rdi + 1]
0x18050:    je      0x18061
0x18052:    lea     rdi, [rip + 0x9f26]
0x18059:    call    0x8b00
0x1805e:    mov     rax, r12
0x18061:    movzx   edx, byte ptr [rax]
0x18064:    mov     rsi, rax
0x18067:    test    dl, dl
0x18069:    jne     0x1808c
0x1806b:    jmp     0x180f0
0x180f0:
0x18070:    cmp     dl, 0x2e
0x18073:    jne     0x18118
0x18079:    movzx   ecx, byte ptr [rax + 1]
0x1807d:    cmp     cl, 0x2f
0x18080:    je      0x18086
0x18082:    test    cl, cl
0x18084:    jne     0x180a0
0x18086:    add     rax, 1
0x1808a:    mov     edx, ecx
0x1808c:    cmp     dl, 0x2f
0x1808f:    jne     0x18070
0x18091:    movzx   ecx, byte ptr [rax + 1]
0x18095:    jmp     0x18086
0x18086:
0x18097:    nop     word ptr [rax + rax]
0x180a0:    cmp     cl, 0x2e
0x180a3:    jne     0x180c0
0x180a5:    movzx   edx, byte ptr [rax + 2]
0x180a9:    cmp     dl, 0x2f
0x180ac:    je      0x18120
0x180ae:    test    dl, dl
0x180b0:    je      0x18120
0x180b2:    mov     edx, 0x2e
0x180b7:    nop     word ptr [rax + rax]
0x180c0:    mov     rdi, rax

```

```

0x180c3:    mov     rcx, rsi
0x180c6:    add     rax, 1
0x180ca:    add     rsi, 1
0x180ce:    mov     byte ptr [rsi - 1], dl
0x180d1:    movzx   edx, byte ptr [rax]
0x180d4:    test    dl, dl
0x180d6:    je      0x180f0
0x180d8:    cmp     dl, 0x2f
0x180db:    jne     0x180c0
0x180dd:    test    dl, dl
0x180df:    je      0x180f0
0x180e1:    mov     byte ptr [rsi], 0x2f
0x180e4:    movzx   edx, byte ptr [rdi + 1]
0x180e8:    lea     rsi, [rcx + 2]
0x180ec:    test    dl, dl
0x180ee:    jne     0x1808c
0x180f0:    cmp     byte ptr [rsi - 1], 0x2f
0x180f4:    jne     0x18109
0x180f6:    mov     rax, rsi
0x180f9:    sub     rax, r12
0x180fc:    cmp     rax, 1
0x18100:    setg    al
0x18103:    movzx   eax, al
0x18106:    sub     rsi, rax
0x18109:    mov     rax, r12
0x1810c:    mov     byte ptr [rsi], 0
0x1810f:    pop     r12
0x18111:    ret
0x18112:    nop     word ptr [rax + rax]
0x18118:    test    dl, dl
0x1811a:    jne     0x180c0
0x1811c:    jmp     0x180f0
0x180f0
0x1811e:    nop
0x18120:    lea     rdi, [rsi - 1]
0x18124:    cmp     rdi, r12
0x18127:    jne     0x1812f
0x18129:    add     rax, 2
0x1812d:    jmp     0x180ec
0x180ec
0x1812f:    cmp     byte ptr [rsi - 2], 0x2f
0x18133:    lea     rcx, [rsi - 2]
0x18137:    je      0x1814e
0x18139:    nop     dword ptr [rax]
0x18140:    mov     rsi, rcx
0x18143:    sub     rcx, 1
0x18147:    cmp     byte ptr [rcx], 0x2f
0x1814a:    jne     0x18140

```

```

0x1814c:      jmp      0x18129
0x18129
0x1814e:      mov      rsi, rdi
0x18151:      jmp      0x18129
0x18129
0x18153:      nop      word ptr cs:[rax + rax]
0x1815e:      nop
0x18160:      endbr64
0x18164:      push     rbx
0x18165:      call     0x4a60
0x1816a:      mov      edi, eax
0x1816c:      mov      ebx, eax
0x1816e:      call     0x4970
0x18173:      test     rax, rax
0x18176:      je       0x18189
0x18178:      mov      rdi, qword ptr [rax]
0x1817b:      call     0x1d7a0
0x18180:      pop      rbx
0x18181:      mov      qword ptr [rip + 0x13910], rax
0x18188:      ret
0x18189:      mov      rcx, qword ptr [rip + 0x12758]
0x18190:      mov      r8d, ebx
0x18193:      mov      rdi, qword ptr [rip + 0x126e6]
0x1819a:      lea      rdx, [rip + 0xa13f]
0x181a1:      mov      esi, 1
0x181a6:      call     0x4e60
0x181ab:      mov      rcx, qword ptr [rip + 0x12736]
0x181b2:      xor      eax, eax
0x181b4:      mov      rdi, qword ptr [rip + 0x126c5]
0x181bb:      lea      r8, [rip + 0x9dd7]
0x181c2:      lea      rdx, [rip + 0xa14f]
0x181c9:      mov      esi, 1
0x181ce:      call     0x4e60
0x181d3:      mov      rdi, qword ptr [rip + 0x126a6]
0x181da:      mov      esi, 1
0x181df:      xor      eax, eax
0x181e1:      mov      rcx, qword ptr [rip + 0x12700]
0x181e8:      lea      rdx, [rip + 0xa161]
0x181ef:      call     0x4e60
0x181f4:      mov      edi, 1
0x181f9:      call     0x4e20
0x181fe:      nop
0x18200:      endbr64
0x18204:      push     r12
0x18206:      push     rbp
0x18207:      mov      rbp, rsi
0x1820a:      push     rbx
0x1820b:      movsxd   rbx, edx

```

```

0x1820e:    cmp     ebx, 2
0x18211:    je      0x18250
0x18213:    mov     r12, rdi
0x18216:    jg      0x18230
0x18218:    test    ebx, ebx
0x1821a:    je      0x18224
0x1821c:    cmp     ebx, 1
0x1821f:    jne     0x18260
0x18221:    mov     byte ptr [rsi], 0x20
0x18224:    pop     rbx
0x18225:    pop     rbp
0x18226:    pop     r12
0x18228:    ret
0x18229:    nop     dword ptr [rax]
0x18230:    cmp     ebx, 3
0x18233:    jne     0x18260
0x18235:    mov     edx, 0x2020
0x1823a:    mov     byte ptr [rsi + 2], 0x20
0x1823e:    mov     word ptr [rsi], dx
0x18241:    pop     rbx
0x18242:    pop     rbp
0x18243:    pop     r12
0x18245:    ret
0x18246:    nop     word ptr cs:[rax + rax]
0x18250:    mov     ecx, 0x2020
0x18255:    mov     word ptr [rsi], cx
0x18258:    pop     rbx
0x18259:    pop     rbp
0x1825a:    pop     r12
0x1825c:    ret
0x1825d:    nop     dword ptr [rax]
0x18260:    mov     rdi, r12
0x18263:    call    0x49b0
0x18268:    movsxd  rdx, eax
0x1826b:    cmp     ebx, eax
0x1826d:    jge     0x182a0
0x1826f:    mov     eax, 0x2e2e
0x18274:    lea     r8d, [rbx - 3]
0x18278:    mov     byte ptr [rbp + 2], 0x2e
0x1827c:    sub     rdx, rbx
0x1827f:    mov     word ptr [rbp], ax
0x18283:    movsxd  r8, r8d
0x18286:    lea     rsi, [r12 + rdx + 3]
0x1828b:    pop     rbx
0x1828c:    lea     rdi, [rbp + 3]
0x18290:    mov     rdx, r8
0x18293:    pop     rbp
0x18294:    pop     r12

```

```

0x18296:      jmp      0x4bb0
0x4bb0
0x1829b:      nop      dword ptr [rax + rax]
0x182a0:      pop      rbx
0x182a1:      mov      rsi, r12
0x182a4:      mov      rdi, rbp
0x182a7:      pop      rbp
0x182a8:      pop      r12
0x182aa:      jmp      0x4bb0
0x4bb0
0x182af:      nop
0x182b0:      endbr64
0x182b4:      push     r12
0x182b6:      mov      r12, rsi
0x182b9:      push     rbp
0x182ba:      mov      rbp, rdi
0x182bd:      push     rbx
0x182be:      lea      ebx, [rdx - 1]
0x182c1:      mov      edx, ebx
0x182c3:      call     0x18200
0x182c8:      mov      rdi, rbp
0x182cb:      call     0x49b0
0x182d0:      movsxd   rcx, ebx
0x182d3:      cmp      ebx, eax
0x182d5:      movsxd   rdx, eax
0x182d8:      mov      rax, rcx
0x182db:      pop      rbx
0x182dc:      cmovg    rax, rdx
0x182e0:      pop      rbp
0x182e1:      mov      byte ptr [r12 + rax], 0
0x182e6:      mov      rax, r12
0x182e9:      pop      r12
0x182eb:      ret
0x182ec:      nop      dword ptr [rax]
0x182f0:      endbr64
0x182f4:      push     r13
0x182f6:      mov      edx, 1
0x182fb:      xor      esi, esi
0x182fd:      push     r12
0x182ff:      push     rbp
0x18300:      mov      ebp, edi
0x18302:      call     0x4e50
0x18307:      mov      edi, ebp
0x18309:      mov      edx, 2
0x1830e:      xor      esi, esi
0x18310:      mov      r13, rax
0x18313:      call     0x4e50
0x18318:      mov      rsi, r13

```



```

0x1831b:    mov     edi, ebp
0x1831d:    xor     edx, edx
0x1831f:    mov     r12, rax
0x18322:    call    0x4e50
0x18327:    mov     rax, r12
0x1832a:    pop     rbp
0x1832b:    pop     r12
0x1832d:    pop     r13
0x1832f:    ret
0x18330:    endbr64
0x18334:    sub     rsp, 0x18
0x18338:    xor     edi, edi
0x1833a:    mov     rax, qword ptr fs:[0x28]
0x18343:    mov     qword ptr [rsp + 8], rax
0x18348:    xor     eax, eax
0x1834a:    call    0x4be0
0x1834f:    mov     rdi, rsp
0x18352:    mov     qword ptr [rsp], rax
0x18356:    call    0x4840
0x1835b:    mov     rdx, qword ptr [rsp + 8]
0x18360:    sub     rdx, qword ptr fs:[0x28]
0x18369:    jne     0x18370
0x1836b:    add     rsp, 0x18
0x1836f:    ret
0x18370:    call    0x49d0
0x18375:    nop     word ptr cs:[rax + rax]
0x18380:    endbr64
0x18384:    push    r12
0x18386:    push    rbp
0x18387:    mov     rbp, rsi
0x1838a:    push    rbx
0x1838b:    mov     rbx, rdi
0x1838e:    call    0x49b0
0x18393:    mov     rdi, rbp
0x18396:    mov     r12, rax
0x18399:    call    0x49b0
0x1839e:    lea     rdi, [r12 + rax + 2]
0x183a3:    call    0x1d670
0x183a8:    mov     r9, rbp
0x183ab:    mov     r8, rbx
0x183ae:    mov     esi, 1
0x183b3:    mov     r12, rax
0x183b6:    mov     rdi, rax
0x183b9:    xor     eax, eax
0x183bb:    mov     rdx, -1
0x183c2:    lea     rcx, [rip + 0x9be5]
0x183c9:    call    0x4f10
0x183ce:    mov     rdi, r12

```

```

0x183d1:    call    0x4b20
0x183d6:    mov     r12d, eax
0x183d9:    cmp     eax, -1
0x183dc:    je      0x183f0
0x183de:    mov     eax, r12d
0x183e1:    pop     rbx
0x183e2:    pop     rbp
0x183e3:    pop     r12
0x183e5:    ret
0x183e6:    nop     word ptr cs:[rax + rax]
0x183f0:    mov     rcx, qword ptr [rip + 0x124f1]
0x183f7:    mov     rdi, qword ptr [rip + 0x12482]
0x183fe:    mov     r8, rbx
0x18401:    xor     eax, eax
0x18403:    lea     rdx, [rip + 0x9f6e]
0x1840a:    mov     esi, 1
0x1840f:    call    0x4e60
0x18414:    mov     eax, r12d
0x18417:    pop     rbx
0x18418:    pop     rbp
0x18419:    pop     r12
0x1841b:    ret
0x1841c:    nop     dword ptr [rax]
0x18420:    endbr64
0x18424:    push    r13
0x18426:    push    r12
0x18428:    lea     r12, [rip + 0x9b85]
0x1842f:    push    rbp
0x18430:    mov     rdi, r12
0x18433:    push    rbx
0x18434:    sub     rsp, 8
0x18438:    call    0x47e0
0x1843d:    mov     r13, qword ptr [rip + 0x1365c]
0x18444:    test    rax, rax
0x18447:    je      0x184b0
0x18449:    mov     rdi, rax
0x1844c:    mov     rbx, rax
0x1844f:    call    0x49b0
0x18454:    mov     rdi, r13
0x18457:    mov     rbp, rax
0x1845a:    call    0x49b0
0x1845f:    lea     rdi, [rbp + rax + 2]
0x18464:    call    0x1d670
0x18469:    mov     r9, qword ptr [rip + 0x13630]
0x18470:    mov     r8, rbx
0x18473:    lea     rcx, [rip + 0x9b3f]
0x1847a:    mov     rbp, rax
0x1847d:    mov     rdi, rax

```

```

0x18480:    mov     esi, 1
0x18485:    xor     eax, eax
0x18487:    mov     rdx, -1
0x1848e:    call    0x4f10
0x18493:    mov     rdi, r12
0x18496:    mov     rsi, rbp
0x18499:    call    0x18380
0x1849e:    add     rsp, 8
0x184a2:    mov     rdi, rbp
0x184a5:    pop     rbx
0x184a6:    pop     rbp
0x184a7:    pop     r12
0x184a9:    pop     r13
0x184ab:    jmp     0x18d10
0x18d10:
0x184b0:    add     rsp, 8
0x184b4:    mov     rsi, r13
0x184b7:    mov     rdi, r12
0x184ba:    pop     rbx
0x184bb:    pop     rbp
0x184bc:    pop     r12
0x184be:    pop     r13
0x184c0:    jmp     0x18380
0x18380:
0x184c5:    nop     word ptr cs:[rax + rax]
0x184d0:    endbr64
0x184d4:    test    rsi, rsi
0x184d7:    je      0x18520
0x184d9:    push    rbp
0x184da:    mov     rbp, rsi
0x184dd:    push    rbx
0x184de:    mov     rbx, rdi
0x184e1:    sub     rsp, 8
0x184e5:    call    0x4f00
0x184ea:    mov     rdi, rbx
0x184ed:    lea     rsi, [rbx + rbp]
0x184f1:    nop     dword ptr [rax]
0x184f8:    movsx   rcx, byte ptr [rdi]
0x184fc:    mov     rdx, qword ptr [rax]
0x184ff:    test    byte ptr [rdx + rcx*2 + 1], 0x40
0x18504:    jne     0x18509
0x18506:    mov     byte ptr [rdi], 0x3f
0x18509:    add     rdi, 1
0x1850d:    cmp     rsi, rdi
0x18510:    jne     0x184f8
0x18512:    add     rsp, 8
0x18516:    pop     rbx
0x18517:    pop     rbp

```

```

0x18518:      ret
0x18519:      nop     dword ptr [rax]
0x18520:      ret
0x18521:      nop     word ptr cs:[rax + rax]
0x1852c:      nop     dword ptr [rax]
0x18530:      endbr64
0x18534:      test    rsi, rsi
0x18537:      je      0x185c2
0x1853d:      push    rbp
0x1853e:      mov     rbp, rsi
0x18541:      push    rbx
0x18542:      mov     rbx, rdi
0x18545:      sub     rsp, 8
0x18549:      call    0x4f00
0x1854e:      mov     rdi, rbx
0x18551:      lea     rsi, [rbx + rbp]
0x18555:      movabs  r9, 0x68000000c0000001
0x1855f:      movabs  r8, 0x2400f82000000000
0x18569:      mov     rcx, qword ptr [rax]
0x1856c:      nop     dword ptr [rax]
0x18570:      movsx   rdx, byte ptr [rdi]
0x18574:      mov     rax, rdx
0x18577:      test    byte ptr [rcx + rdx*2], 8
0x1857b:      jne     0x185b0
0x1857d:      cmp     dl, 0x3d
0x18580:      jg      0x185a0
0x18582:      cmp     dl, 0x24
0x18585:      jle     0x1858d
0x18587:      bt      r8, rdx
0x1858b:      jb      0x185b0
0x1858d:      add     rsp, 8
0x18591:      mov     eax, 1
0x18596:      pop     rbx
0x18597:      pop     rbp
0x18598:      ret
0x18599:      nop     dword ptr [rax]
0x185a0:      sub     eax, 0x40
0x185a3:      cmp     al, 0x3e
0x185a5:      ja      0x1858d
0x185a7:      bt      r9, rax
0x185ab:      jae     0x1858d
0x185ad:      nop     dword ptr [rax]
0x185b0:      add     rdi, 1
0x185b4:      cmp     rsi, rdi
0x185b7:      jne     0x18570
0x185b9:      add     rsp, 8
0x185bd:      xor     eax, eax
0x185bf:      pop     rbx

```

```

0x185c0:    pop     rbp
0x185c1:    ret
0x185c2:    xor     eax, eax
0x185c4:    ret
0x185c5:    nop     word ptr cs:[rax + rax]
0x185d0:    endbr64
0x185d4:    push    rbx
0x185d5:    mov     rbx, rdi
0x185d8:    call    0x49b0
0x185dd:    sub     eax, 1
0x185e0:    js      0x18618
0x185e2:    cdqe
0x185e4:    jmp     0x18602
0x18602:
0x185e6:    nop     word ptr cs:[rax + rax]
0x185f0:    cmp     dl, 0x20
0x185f3:    je      0x185fa
0x185f5:    cmp     dl, 9
0x185f8:    jne     0x18618
0x185fa:    sub     rax, 1
0x185fe:    test    eax, eax
0x18600:    js      0x18618
0x18602:    movzx   edx, byte ptr [rbx + rax]
0x18606:    cmp     dl, 0x26
0x18609:    jne     0x185f0
0x1860b:    mov     eax, 1
0x18610:    pop     rbx
0x18611:    ret
0x18612:    nop     word ptr [rax + rax]
0x18618:    xor     eax, eax
0x1861a:    pop     rbx
0x1861b:    ret
0x1861c:    nop     dword ptr [rax]
0x18620:    endbr64
0x18624:    jmp     0x1863c
0x1863c:
0x18626:    nop     word ptr cs:[rax + rax]
0x18630:    cmp     al, 0x20
0x18632:    je      0x18638
0x18634:    cmp     al, 0x3b
0x18636:    jne     0x18650
0x18638:    add     rdi, 1
0x1863c:    movzx   eax, byte ptr [rdi]
0x1863f:    test    al, al
0x18641:    jne     0x18630
0x18643:    mov     eax, 1
0x18648:    ret
0x18649:    nop     dword ptr [rax]

```

```

0x18650:    xor     eax, eax
0x18652:    ret
0x18653:    nop     word ptr cs:[rax + rax]
0x1865e:    nop
0x18660:    endbr64
0x18664:    push    r15
0x18666:    push    r14
0x18668:    push    r13
0x1866a:    push    r12
0x1866c:    push    rbp
0x1866d:    push    rbx
0x1866e:    sub     rsp, 0x98
0x18675:    mov     rax, qword ptr fs:[0x28]
0x1867e:    mov     qword ptr [rsp + 0x88], rax
0x18686:    xor     eax, eax
0x18688:    lea     r15, [rsp + 0x10]
0x1868d:    lea     r13, [rsp + 0x30]
0x18692:    mov     qword ptr [rsp + 8], 0
0x1869b:    mov     qword ptr [rsp], 0
0x186a3:    jmp     0x186d3
0x186d3:
0x186a5:    nop     dword ptr [rax]
0x186a8:    mov     rax, qword ptr [rsp]
0x186ac:    mov     qword ptr [rax + 0x18], r14
0x186b0:    mov     rdi, r13
0x186b3:    call    0x1d7a0
0x186b8:    mov     dword ptr [r14 + 8], r12d
0x186bc:    mov     qword ptr [r14], rax
0x186bf:    mov     dword ptr [r14 + 0xc], ebp
0x186c3:    mov     dword ptr [r14 + 0x10], ebx
0x186c7:    mov     qword ptr [r14 + 0x18], 0
0x186cf:    mov     qword ptr [rsp], r14
0x186d3:    xor     ecx, ecx
0x186d5:    mov     edx, 3
0x186da:    mov     rsi, r15
0x186dd:    mov     rdi, r13
0x186e0:    call    0x17320
0x186e5:    cmp     byte ptr [rsp + 0x30], 0
0x186ea:    je      0x18758
0x186ec:    mov     rdi, qword ptr [rsp + 0x10]
0x186f1:    mov     r12d, 0xffffffff
0x186f7:    test    rdi, rdi
0x186fa:    je      0x18704
0x186fc:    call    0xbd00
0x18701:    mov     r12d, eax
0x18704:    mov     rdi, qword ptr [rsp + 0x18]
0x18709:    mov     ebp, 0xffffffff
0x1870e:    test    rdi, rdi

```

```

0x18711:      je      0x1871a
0x18713:      call   0xbd00
0x18718:      mov     ebp, eax
0x1871a:      mov     rdi, qword ptr [rsp + 0x20]
0x1871f:      mov     ebx, 0xffffffff
0x18724:      test    rdi, rdi
0x18727:      je      0x18730
0x18729:      call   0xbd00
0x1872e:      mov     ebx, eax
0x18730:      mov     edi, 0x20
0x18735:      call   0x1d670
0x1873a:      cmp     qword ptr [rsp + 8], 0
0x18740:      mov     r14, rax
0x18743:      jne     0x186a8
0x18749:      mov     qword ptr [rsp + 8], rax
0x1874e:      jmp     0x186b0
0x186b0:
0x18753:      nop     dword ptr [rax + rax]
0x18758:      mov     rdx, qword ptr [rsp + 8]
0x1875d:      test    rdx, rdx
0x18760:      je      0x1877d
0x18762:      mov     rax, qword ptr [rip + 0x13327]
0x18769:      test    rax, rax
0x1876c:      je      0x187a8
0x1876e:      mov     rcx, qword ptr [rsp]
0x18772:      mov     qword ptr [rip + 0x13317], rdx
0x18779:      mov     qword ptr [rcx + 0x18], rax
0x1877d:      mov     rax, qword ptr [rsp + 0x88]
0x18785:      sub     rax, qword ptr fs:[0x28]
0x1878e:      jne     0x187b6
0x18790:      add     rsp, 0x98
0x18797:      pop     rbx
0x18798:      pop     rbp
0x18799:      pop     r12
0x1879b:      pop     r13
0x1879d:      pop     r14
0x1879f:      pop     r15
0x187a1:      ret
0x187a2:      nop     word ptr [rax + rax]
0x187a8:      mov     rax, qword ptr [rsp + 8]
0x187ad:      mov     qword ptr [rip + 0x132dc], rax
0x187b4:      jmp     0x1877d
0x1877d:
0x187b6:      call   0x49d0
0x187bb:      nop     dword ptr [rax + rax]
0x187c0:      endbr64
0x187c4:      sub     rsp, 8
0x187c8:      lea     rsi, [rip - 0xf]

```

```

0x187cf:      call    0x4b50
0x187d4:      mov     dword ptr [rip + 0x132d2], 1
0x187de:      add     rsp, 8
0x187e2:      ret
0x187e3:      nop     word ptr cs:[rax + rax]
0x187ee:      nop
0x187f0:      cmp     edi, 0x14
0x187f3:      je      0x18850
0x187f5:      jg      0x18818
0x187f7:      cmp     edi, 0xe
0x187fa:      je      0x18838
0x187fc:      cmp     edi, 0x12
0x187ff:      jne     0x18861
0x18801:      lea     rsi, [rip + 0x148]
0x18808:      mov     edi, 0x12
0x1880d:      jmp     0x4b50
0x4b50
0x18812:      nop     word ptr [rax + rax]
0x18818:      cmp     edi, 0x1c
0x1881b:      jne     0x18830
0x1881d:      lea     rsi, [rip + 0x1cc]
0x18824:      mov     edi, 0x1c
0x18829:      jmp     0x4b50
0x4b50
0x1882e:      nop
0x18830:      ret
0x18831:      nop     dword ptr [rax]
0x18838:      lea     rsi, [rip + 0x31]
0x1883f:      mov     edi, 0xe
0x18844:      jmp     0x4b50
0x4b50
0x18849:      nop     dword ptr [rax]
0x18850:      lea     rsi, [rip + 0x89]
0x18857:      mov     edi, 0x14
0x1885c:      jmp     0x4b50
0x4b50
0x18861:      ret
0x18862:      nop     word ptr cs:[rax + rax]
0x1886d:      nop     dword ptr [rax]
0x18870:      endbr64
0x18874:      mov     edx, dword ptr [rip + 0x1323a]
0x1887a:      push    rbp
0x1887b:      mov     eax, 1
0x18880:      mov     ebp, edi
0x18882:      test    edx, edx
0x18884:      jne     0x188b0
0x18886:      mov     edi, ebp
0x18888:      mov     dword ptr [rip + 0x1322e], eax

```



```

0x1888e:      call    0x187f0
0x18893:      xor     eax, eax
0x18895:      call    0x18330
0x1889a:      mov     edi, 0x3c
0x1889f:      pop     rbp
0x188a0:      sub     edi, dword ptr [rax]
0x188a2:      jmp     0x4aa0
0x4aa0
0x188a7:      nop     word ptr [rax + rax]
0x188b0:      xor     eax, eax
0x188b2:      call    0x18330
0x188b7:      mov     eax, dword ptr [rax]
0x188b9:      test    eax, eax
0x188bb:      je      0x188d0
0x188bd:      xor     eax, eax
0x188bf:      call    0xb4e0
0x188c4:      xor     eax, eax
0x188c6:      jmp     0x18886
0x18886
0x188c8:      nop     dword ptr [rax + rax]
0x188d0:      mov     edi, ebp
0x188d2:      call    0x16fd0
0x188d7:      jmp     0x188bd
0x188bd
0x188d9:      nop     dword ptr [rax]
0x188e0:      endbr64
0x188e4:      mov     eax, dword ptr [rip + 0x131ca]
0x188ea:      push    rbp
0x188eb:      mov     ebp, edi
0x188ed:      test    eax, eax
0x188ef:      jne     0x18908
0x188f1:      mov     eax, 1
0x188f6:      mov     edi, ebp
0x188f8:      pop     rbp
0x188f9:      mov     dword ptr [rip + 0x131c5], eax
0x188ff:      jmp     0x187f0
0x187f0
0x18904:      nop     dword ptr [rax]
0x18908:      xor     eax, eax
0x1890a:      call    0x9c90
0x1890f:      cmp     eax, 1
0x18912:      sete    al
0x18915:      movzx   eax, al
0x18918:      mov     dword ptr [rip + 0x1319a], eax
0x1891e:      xor     eax, eax
0x18920:      call    0x9610
0x18925:      call    0x4920
0x1892a:      mov     esi, 0x13

```

```

0x1892f:      mov     edi, eax
0x18931:      call    0x4bd0
0x18936:      xor     eax, eax
0x18938:      mov     edi, ebp
0x1893a:      pop     rbp
0x1893b:      mov     dword ptr [rip + 0x13183], eax
0x18941:      jmp     0x187f0
0x187f0
0x18946:      nop     word ptr cs:[rax + rax]
0x18950:      endbr64
0x18954:      mov     eax, dword ptr [rip + 0x1315e]
0x1895a:      push    rbx
0x1895b:      mov     ebx, edi
0x1895d:      test    eax, eax
0x1895f:      jne     0x189a0
0x18961:      cmp     ebx, 0x14
0x18964:      je      0x189b4
0x18966:      jg      0x18988
0x18968:      cmp     ebx, 0xe
0x1896b:      je      0x189d8
0x1896d:      cmp     ebx, 0x12
0x18970:      jne     0x189d0
0x18972:      lea     rsi, [rip - 0x29]
0x18979:      mov     edi, 0x12
0x1897e:      pop     rbx
0x1897f:      jmp     0x4b50
0x4b50
0x18984:      nop     dword ptr [rax]
0x18988:      cmp     ebx, 0x1c
0x1898b:      jne     0x189d0
0x1898d:      lea     rsi, [rip + 0x5c]
0x18994:      mov     edi, 0x1c
0x18999:      pop     rbx
0x1899a:      jmp     0x4b50
0x4b50
0x1899f:      nop
0x189a0:      call    0x7bd0
0x189a5:      mov     dword ptr [rip + 0x13111], 0
0x189af:      cmp     ebx, 0x14
0x189b2:      jne     0x18966
0x189b4:      lea     rsi, [rip - 0xdb]
0x189bb:      mov     edi, 0x14
0x189c0:      pop     rbx
0x189c1:      jmp     0x4b50
0x4b50
0x189c6:      nop     word ptr cs:[rax + rax]
0x189d0:      pop     rbx
0x189d1:      ret

```

```

0x189d2:      nop      word ptr [rax + rax]
0x189d8:      lea       rsi, [rip - 0x16f]
0x189df:      mov       edi, 0xe
0x189e4:      pop       rbx
0x189e5:      jmp       0x4b50
0x4b50
0x189ea:      nop      word ptr [rax + rax]
0x189f0:      endbr64
0x189f4:      mov       eax, dword ptr [rip + 0x130ba]
0x189fa:      push      rbx
0x189fb:      mov       ebx, edi
0x189fd:      test      eax, eax
0x189ff:      jne       0x18a50
0x18a01:      mov       eax, 1
0x18a06:      mov       dword ptr [rip + 0x130b4], eax
0x18a0c:      cmp       ebx, 0x14
0x18a0f:      je        0x18a72
0x18a11:      jg        0x18a30
0x18a13:      cmp       ebx, 0xe
0x18a16:      je        0x18a90
0x18a18:      cmp       ebx, 0x12
0x18a1b:      jne       0x18a88
0x18a1d:      lea       rsi, [rip - 0xd4]
0x18a24:      mov       edi, 0x12
0x18a29:      pop       rbx
0x18a2a:      jmp       0x4b50
0x4b50
0x18a2f:      nop
0x18a30:      cmp       ebx, 0x1c
0x18a33:      jne       0x18a88
0x18a35:      lea       rsi, [rip - 0x4c]
0x18a3c:      mov       edi, 0x1c
0x18a41:      pop       rbx
0x18a42:      jmp       0x4b50
0x4b50
0x18a47:      nop      word ptr [rax + rax]
0x18a50:      xor       eax, eax
0x18a52:      call      0xa6c0
0x18a57:      xor       eax, eax
0x18a59:      call      0x9d80
0x18a5e:      mov       edi, ebx
0x18a60:      call      0x7bd0
0x18a65:      xor       eax, eax
0x18a67:      mov       dword ptr [rip + 0x13053], eax
0x18a6d:      cmp       ebx, 0x14
0x18a70:      jne       0x18a11
0x18a72:      lea       rsi, [rip - 0x199]
0x18a79:      mov       edi, 0x14

```

```

0x18a7e:    pop     rbx
0x18a7f:    jmp     0x4b50
0x4b50
0x18a84:    nop     dword ptr [rax]
0x18a88:    pop     rbx
0x18a89:    ret
0x18a8a:    nop     word ptr [rax + rax]
0x18a90:    lea     rsi, [rip - 0x227]
0x18a97:    mov     edi, 0xe
0x18a9c:    pop     rbx
0x18a9d:    jmp     0x4b50
0x4b50
0x18aa2:    nop     word ptr cs:[rax + rax]
0x18aad:    nop     dword ptr [rax]
0x18ab0:    endbr64
0x18ab4:    sub     rsp, 8
0x18ab8:    mov     ecx, dword ptr [rip + 0x13006]
0x18abe:    test    ecx, ecx
0x18ac0:    jne     0x18af0
0x18ac2:    mov     edx, dword ptr [rip + 0x12ff8]
0x18ac8:    test    edx, edx
0x18aca:    jne     0x18b30
0x18acc:    mov     eax, dword ptr [rip + 0x12fea]
0x18ad2:    test    eax, eax
0x18ad4:    jne     0x18ae0
0x18ad6:    add     rsp, 8
0x18ada:    ret
0x18adb:    nop     dword ptr [rax + rax]
0x18ae0:    mov     dword ptr [rip + 0x12fd2], 0
0x18aea:    add     rsp, 8
0x18aee:    ret
0x18aef:    nop
0x18af0:    xor     eax, eax
0x18af2:    call    0x9610
0x18af7:    call    0x4920
0x18afc:    mov     esi, 0x13
0x18b01:    mov     edi, eax
0x18b03:    call    0x4bd0
0x18b08:    mov     dword ptr [rip + 0x12fb2], 0
0x18b12:    mov     dword ptr [rip + 0x12fa4], 0
0x18b1c:    mov     dword ptr [rip + 0x12f96], 0
0x18b26:    add     rsp, 8
0x18b2a:    ret
0x18b2b:    nop     dword ptr [rax + rax]
0x18b30:    xor     eax, eax
0x18b32:    call    0xa6c0
0x18b37:    xor     eax, eax
0x18b39:    call    0x9d80

```

```

0x18b3e:    mov     edi, 0x1c
0x18b43:    call    0x7bd0
0x18b48:    mov     dword ptr [rip + 0x12f6e], 0
0x18b52:    mov     dword ptr [rip + 0x12f60], 0
0x18b5c:    add     rsp, 8
0x18b60:    ret
0x18b61:    nop     word ptr cs:[rax + rax]
0x18b6c:    nop     dword ptr [rax]
0x18b70:    endbr64
0x18b74:    mov     dword ptr [rip + 0x12f3a], edi
0x18b7a:    test    edi, edi
0x18b7c:    jne     0x18b80
0x18b7e:    ret
0x18b7f:    nop
0x18b80:    xor     eax, eax
0x18b82:    jmp     0x18ab0
0x18ab0:
0x18b87:    nop     word ptr [rax + rax]
0x18b90:    endbr64
0x18b94:    sub     rsp, 8
0x18b98:    mov     eax, dword ptr [rip + 0x11bfa]
0x18b9e:    cmp     edi, 1
0x18ba1:    je      0x18bf0
0x18ba3:    test    eax, eax
0x18ba5:    jne     0x18bd0
0x18ba7:    mov     esi, 1
0x18bac:    mov     edi, 0x1c
0x18bb1:    call    0x4b50
0x18bb6:    mov     esi, 1
0x18bbb:    mov     edi, 0xe
0x18bc0:    add     rsp, 8
0x18bc4:    jmp     0x4b50
0x4b50:
0x18bc9:    nop     dword ptr [rax]
0x18bd0:    mov     esi, 1
0x18bd5:    mov     edi, 0x14
0x18bda:    call    0x4b50
0x18bdf:    mov     esi, 1
0x18be4:    mov     edi, 0x12
0x18be9:    call    0x4b50
0x18bee:    jmp     0x18ba7
0x18ba7:
0x18bf0:    test    eax, eax
0x18bf2:    jne     0x18c20
0x18bf4:    lea     rsi, [rip - 0x20b]
0x18bfb:    mov     edi, 0x1c
0x18c00:    call    0x4b50
0x18c05:    lea     rsi, [rip - 0x39c]

```

```

0x18c0c:      mov     edi, 0xe
0x18c11:      add     rsp, 8
0x18c15:      jmp     0x4b50
0x4b50
0x18c1a:      nop     word ptr [rax + rax]
0x18c20:      lea     rsi, [rip - 0x347]
0x18c27:      mov     edi, 0x14
0x18c2c:      call    0x4b50
0x18c31:      lea     rsi, [rip - 0x2e8]
0x18c38:      mov     edi, 0x12
0x18c3d:      call    0x4b50
0x18c42:      jmp     0x18bf4
0x18bf4
0x18c44:      nop     word ptr cs:[rax + rax]
0x18c4f:      nop
0x18c50:      endbr64
0x18c54:      push    rbp
0x18c55:      mov     esi, 1
0x18c5a:      mov     edi, 0x14
0x18c5f:      call    0x4b50
0x18c64:      mov     esi, 1
0x18c69:      mov     edi, 0x12
0x18c6e:      cmp     rax, 1
0x18c72:      setne   al
0x18c75:      movzx   eax, al
0x18c78:      mov     dword ptr [rip + 0x11b1a], eax
0x18c7e:      call    0x4b50
0x18c83:      mov     esi, 1
0x18c88:      mov     edi, 0x1c
0x18c8d:      call    0x4b50
0x18c92:      lea     rbp, [rip - 0x1519]
0x18c99:      mov     edi, 0xb
0x18c9e:      mov     rsi, rbp
0x18ca1:      call    0x4b50
0x18ca6:      mov     rsi, rbp
0x18ca9:      mov     edi, 1
0x18cae:      call    0x4b50
0x18cb3:      mov     rsi, rbp
0x18cb6:      mov     edi, 0xf
0x18cbb:      call    0x4b50
0x18cc0:      mov     esi, 1
0x18cc5:      mov     edi, 3
0x18cca:      call    0x4b50
0x18ccf:      lea     rsi, [rip - 0x516]
0x18cd6:      mov     edi, 2
0x18cdb:      pop     rbp
0x18cdc:      jmp     0x4b50
0x4b50

```

```

0x18ce1:      nop      word ptr cs:[rax + rax]
0x18ceb:      nop      dword ptr [rax + rax]
0x18cf0:      endbr64
0x18cf4:      push     rax
0x18cf5:      pop      rax
0x18cf6:      lea      rdi, [rip + 0x96a8]
0x18cfd:      sub      rsp, 8
0x18d01:      call     0x8b00
0x18d06:      call     0x4850
0x18d0b:      nop      dword ptr [rax + rax]
0x18d10:      endbr64
0x18d14:      test     rdi, rdi
0x18d17:      je       0x18d20
0x18d19:      jmp      0x4800
0x4800
0x18d1e:      nop
0x18d20:      lea      rdi, [rip + 0x9697]
0x18d27:      jmp      0x8b00
0x8b00
0x18d2c:      nop      dword ptr [rax]
0x18d30:      endbr64
0x18d34:      xor      r8d, r8d
0x18d37:      test     rdx, rdx
0x18d3a:      je       0x18d90
0x18d3c:      push     r14
0x18d3e:      push     r13
0x18d40:      push     r12
0x18d42:      mov      r12, rsi
0x18d45:      push     rbp
0x18d46:      mov      ebp, edi
0x18d48:      push     rbx
0x18d49:      mov      rbx, rdx
0x18d4c:      call     0x4860
0x18d51:      mov      r13d, dword ptr [rax]
0x18d54:      mov      r14, rax
0x18d57:      jmp      0x18d66
0x18d66
0x18d59:      nop      dword ptr [rax]
0x18d60:      cmp      dword ptr [r14], 4
0x18d64:      jne      0x18d7d
0x18d66:      mov      dword ptr [r14], r13d
0x18d69:      mov      rdx, rbx
0x18d6c:      mov      rsi, r12
0x18d6f:      mov      edi, ebp
0x18d71:      call     0x47a0
0x18d76:      mov      r8d, eax
0x18d79:      test     eax, eax
0x18d7b:      js       0x18d60

```

```

0x18d7d:    pop     rbx
0x18d7e:    mov     eax, r8d
0x18d81:    pop     rbp
0x18d82:    pop     r12
0x18d84:    pop     r13
0x18d86:    pop     r14
0x18d88:    ret
0x18d89:    nop     dword ptr [rax]
0x18d90:    mov     eax, r8d
0x18d93:    ret
0x18d94:    nop     word ptr cs:[rax + rax]
0x18d9f:    nop
0x18da0:    endbr64
0x18da4:    xor     r8d, r8d
0x18da7:    test    rdx, rdx
0x18daa:    je      0x18e00
0x18dac:    push    r14
0x18dae:    push    r13
0x18db0:    push    r12
0x18db2:    mov     r12, rsi
0x18db5:    push    rbp
0x18db6:    mov     ebp, edi
0x18db8:    push    rbx
0x18db9:    mov     rbx, rdx
0x18dbc:    call    0x4860
0x18dc1:    mov     r13d, dword ptr [rax]
0x18dc4:    mov     r14, rax
0x18dc7:    jmp     0x18dd6
0x18dd6:
0x18dc9:    nop     dword ptr [rax]
0x18dd0:    cmp     dword ptr [r14], 4
0x18dd4:    jne     0x18ded
0x18dd6:    mov     dword ptr [r14], r13d
0x18dd9:    mov     rdx, rbx
0x18ddc:    mov     rsi, r12
0x18ddf:    mov     edi, ebp
0x18de1:    call    0x4790
0x18de6:    mov     r8d, eax
0x18de9:    test    eax, eax
0x18deb:    js      0x18dd0
0x18ded:    pop     rbx
0x18dee:    mov     eax, r8d
0x18df1:    pop     rbp
0x18df2:    pop     r12
0x18df4:    pop     r13
0x18df6:    pop     r14
0x18df8:    ret
0x18df9:    nop     dword ptr [rax]

```



```

0x18e00:    mov     eax, r8d
0x18e03:    ret
0x18e04:    nop     word ptr cs:[rax + rax]
0x18e0f:    nop
0x18e10:    endbr64
0x18e14:    xor     r8d, r8d
0x18e17:    test    rdx, rdx
0x18e1a:    je      0x18e70
0x18e1c:    push    r14
0x18e1e:    push    r13
0x18e20:    push    r12
0x18e22:    mov     r12, rsi
0x18e25:    push    rbp
0x18e26:    mov     rbp, rdi
0x18e29:    push    rbx
0x18e2a:    mov     rbx, rdx
0x18e2d:    call    0x4860
0x18e32:    mov     r13d, dword ptr [rax]
0x18e35:    mov     r14, rax
0x18e38:    jmp     0x18e46
0x18e46:
0x18e3a:    nop     word ptr [rax + rax]
0x18e40:    cmp     dword ptr [r14], 4
0x18e44:    jne     0x18e5e
0x18e46:    mov     dword ptr [r14], r13d
0x18e49:    mov     rdx, rbx
0x18e4c:    mov     rsi, r12
0x18e4f:    mov     rdi, rbp
0x18e52:    call    0x4910
0x18e57:    mov     r8d, eax
0x18e5a:    test    eax, eax
0x18e5c:    js      0x18e40
0x18e5e:    pop     rbx
0x18e5f:    mov     eax, r8d
0x18e62:    pop     rbp
0x18e63:    pop     r12
0x18e65:    pop     r13
0x18e67:    pop     r14
0x18e69:    ret
0x18e6a:    nop     word ptr [rax + rax]
0x18e70:    mov     eax, r8d
0x18e73:    ret
0x18e74:    nop     word ptr cs:[rax + rax]
0x18e7f:    nop
0x18e80:    endbr64
0x18e84:    push    r13
0x18e86:    mov     r13, rdi
0x18e89:    push    r12

```

```

0x18e8b:    push    rbp
0x18e8c:    push    rbx
0x18e8d:    mov     ebx, 0x64
0x18e92:    sub     rsp, 0x18
0x18e96:    jmp     0x18eaa
0x18eaa:
0x18e98:    nop     dword ptr [rax + rax]
0x18ea0:    mov     rdi, rbp
0x18ea3:    add     ebx, ebx
0x18ea5:    call    0x18d10
0x18eaa:    movsxd  r12, ebx
0x18ead:    mov     rdi, r12
0x18eb0:    call    0x1d670
0x18eb5:    mov     rdx, r12
0x18eb8:    mov     rdi, r13
0x18ebb:    mov     rbp, rax
0x18ebe:    mov     rsi, rax
0x18ec1:    xor     eax, eax
0x18ec3:    call    0x18e10
0x18ec8:    cmp     ebx, eax
0x18eca:    jle     0x18ea0
0x18ecc:    mov     rdi, rbp
0x18ecf:    mov     dword ptr [rsp + 0xc], eax
0x18ed3:    call    0x18d10
0x18ed8:    mov     eax, dword ptr [rsp + 0xc]
0x18edc:    add     rsp, 0x18
0x18ee0:    pop     rbx
0x18ee1:    pop     rbp
0x18ee2:    pop     r12
0x18ee4:    pop     r13
0x18ee6:    ret
0x18ee7:    nop     word ptr [rax + rax]
0x18ef0:    endbr64
0x18ef4:    push    r13
0x18ef6:    push    r12
0x18ef8:    mov     r12d, edi
0x18efb:    push    rbp
0x18efc:    mov     rbp, rsi
0x18eff:    push    rbx
0x18f00:    sub     rsp, 8
0x18f04:    call    0x4860
0x18f09:    mov     r13d, dword ptr [rax]
0x18f0c:    mov     rbx, rax
0x18f0f:    jmp     0x18f1d
0x18f1d:
0x18f11:    nop     dword ptr [rax]
0x18f18:    cmp     dword ptr [rbx], 4
0x18f1b:    jne     0x18f2f

```

```

0x18f1d:    mov     dword ptr [rbx], r13d
0x18f20:    mov     rsi, rbp
0x18f23:    mov     edi, r12d
0x18f26:    call    0x4eb0
0x18f2b:    test    eax, eax
0x18f2d:    js      0x18f18
0x18f2f:    add     rsp, 8
0x18f33:    pop     rbx
0x18f34:    pop     rbp
0x18f35:    pop     r12
0x18f37:    pop     r13
0x18f39:    ret
0x18f3a:    nop     word ptr [rax + rax]
0x18f40:    endbr64
0x18f44:    push    r13
0x18f46:    push    r12
0x18f48:    mov     r12, rdi
0x18f4b:    push    rbp
0x18f4c:    mov     rbp, rsi
0x18f4f:    push    rbx
0x18f50:    sub     rsp, 8
0x18f54:    call    0x4860
0x18f59:    mov     r13d, dword ptr [rax]
0x18f5c:    mov     rbx, rax
0x18f5f:    jmp     0x18f6d
0x18f6d:
0x18f61:    nop     dword ptr [rax]
0x18f68:    cmp     dword ptr [rbx], 4
0x18f6b:    jne     0x18f7f
0x18f6d:    mov     dword ptr [rbx], r13d
0x18f70:    mov     rsi, rbp
0x18f73:    mov     rdi, r12
0x18f76:    call    0x4b90
0x18f7b:    test    eax, eax
0x18f7d:    js      0x18f68
0x18f7f:    add     rsp, 8
0x18f83:    pop     rbx
0x18f84:    pop     rbp
0x18f85:    pop     r12
0x18f87:    pop     r13
0x18f89:    ret
0x18f8a:    nop     word ptr [rax + rax]
0x18f90:    endbr64
0x18f94:    push    r13
0x18f96:    push    r12
0x18f98:    mov     r12, rdi
0x18f9b:    push    rbp
0x18f9c:    mov     rbp, rsi

```

```

0x18f9f:    push    rbx
0x18fa0:    sub     rsp, 8
0x18fa4:    call    0x4860
0x18fa9:    mov     r13d, dword ptr [rax]
0x18fac:    mov     rbx, rax
0x18faf:    jmp     0x18fbd
0x18fbd:
0x18fb1:    nop     dword ptr [rax]
0x18fb8:    cmp     dword ptr [rbx], 4
0x18fbb:    jne     0x18fcf
0x18fbd:    mov     dword ptr [rbx], r13d
0x18fc0:    mov     rsi, rbp
0x18fc3:    mov     rdi, r12
0x18fc6:    call    0x4ad0
0x18fcb:    test    eax, eax
0x18fcd:    js      0x18fb8
0x18fcf:    add     rsp, 8
0x18fd3:    pop     rbx
0x18fd4:    pop     rbp
0x18fd5:    pop     r12
0x18fd7:    pop     r13
0x18fd9:    ret
0x18fda:    nop     word ptr [rax + rax]
0x18fe0:    endbr64
0x18fe4:    push    r12
0x18fe6:    push    rbp
0x18fe7:    push    rbx
0x18fe8:    mov     ebx, 0x40
0x18fed:    call    0x4860
0x18ff2:    mov     edi, 0x40
0x18ff7:    mov     dword ptr [rax], 0
0x18ffd:    mov     r12, rax
0x19000:    call    0x1d670
0x19005:    mov     rbp, rax
0x19008:    jmp     0x19031
0x19031:
0x1900a:    nop     word ptr [rax + rax]
0x19010:    cmp     dword ptr [r12], 0x22
0x19015:    jne     0x19050
0x19017:    add     rbx, 0x40
0x1901b:    mov     rdi, rbp
0x1901e:    mov     rsi, rbx
0x19021:    call    0x1d690
0x19026:    mov     dword ptr [r12], 0
0x1902e:    mov     rbp, rax
0x19031:    mov     rsi, rbx
0x19034:    mov     rdi, rbp
0x19037:    call    0x4a80

```

```

0x1903c:      test    rax, rax
0x1903f:      je      0x19010
0x19041:      mov     rax, rbp
0x19044:      pop     rbx
0x19045:      pop     rbp
0x19046:      pop     r12
0x19048:      ret
0x19049:      nop     dword ptr [rax]
0x19050:      mov     rdi, rbp
0x19053:      xor     ebp, ebp
0x19055:      call    0x18d10
0x1905a:      mov     rax, rbp
0x1905d:      pop     rbx
0x1905e:      pop     rbp
0x1905f:      pop     r12
0x19061:      ret
0x19062:      nop     word ptr cs:[rax + rax]
0x1906d:      nop     dword ptr [rax]
0x19070:      endbr64
0x19074:      push    rbp
0x19075:      mov     esi, 0x2f
0x1907a:      mov     rbp, rdi
0x1907d:      push    rbx
0x1907e:      sub     rsp, 8
0x19082:      call    0x4a20
0x19087:      test    rax, rax
0x1908a:      je      0x190a8
0x1908c:      mov     rbx, rax
0x1908f:      mov     rdi, rbp
0x19092:      call    0x1d7a0
0x19097:      sub     rbx, rbp
0x1909a:      mov     byte ptr [rax + rbx], 0x30
0x1909e:      add     rsp, 8
0x190a2:      pop     rbx
0x190a3:      pop     rbp
0x190a4:      ret
0x190a5:      nop     dword ptr [rax]
0x190a8:      add     rsp, 8
0x190ac:      lea     rdi, [rip + 0x862b]
0x190b3:      pop     rbx
0x190b4:      pop     rbp
0x190b5:      jmp     0x1d7a0
0x1d7a0
0x190ba:      nop     word ptr [rax + rax]
0x190c0:      endbr64
0x190c4:      push    rbx
0x190c5:      mov     rbx, rdi
0x190c8:      call    0x49b0

```

```

0x190cd:    lea     rax, [rbx + rax - 1]
0x190d2:    cmp     byte ptr [rax], 0x2f
0x190d5:    jne     0x190da
0x190d7:    mov     byte ptr [rax], 0
0x190da:    mov     rdi, rbx
0x190dd:    mov     esi, 0x2f
0x190e2:    call    0x4a20
0x190e7:    lea     rdx, [rax + 1]
0x190eb:    test    rax, rax
0x190ee:    mov     rax, rbx
0x190f1:    pop     rbx
0x190f2:    cmovne  rax, rdx
0x190f6:    ret
0x190f7:    nop     word ptr [rax + rax]
0x19100:    endbr64
0x19104:    push    rbx
0x19105:    mov     esi, 0x10
0x1910a:    mov     ebx, edi
0x1910c:    mov     edi, 1
0x19111:    call    0x1d740
0x19116:    mov     dword ptr [rax + 8], ebx
0x19119:    pop     rbx
0x1911a:    ret
0x1911b:    nop     dword ptr [rax + rax]
0x19120:    endbr64
0x19124:    test    rdi, rdi
0x19127:    je      0x19148
0x19129:    push    rbp
0x1912a:    mov     rbp, rdi
0x1912d:    mov     rdi, qword ptr [rdi]
0x19130:    test    rdi, rdi
0x19133:    je      0x1913a
0x19135:    call    0x18d10
0x1913a:    mov     rdi, rbp
0x1913d:    pop     rbp
0x1913e:    jmp     0x18d10
0x18d10
0x19143:    nop     dword ptr [rax + rax]
0x19148:    ret
0x19149:    nop     dword ptr [rax]
0x19150:    endbr64
0x19154:    push    rbp
0x19155:    mov     rbp, rsi
0x19158:    push    rbx
0x19159:    mov     rbx, rdi
0x1915c:    sub     rsp, 8
0x19160:    mov     eax, dword ptr [rdi + 0xc]
0x19163:    lea     esi, [rax + 1]

```

```

0x19166:      mov     dword ptr [rdi + 0xc], esi
0x19169:      imul    esi, dword ptr [rdi + 8]
0x1916d:      mov     rdi, qword ptr [rdi]
0x19170:      movsxd  rsi, esi
0x19173:      call    0x1d690
0x19178:      mov     ecx, dword ptr [rbx + 0xc]
0x1917b:      movsxd  rdx, dword ptr [rbx + 8]
0x1917f:      mov     rsi, rbp
0x19182:      mov     qword ptr [rbx], rax
0x19185:      add     rsp, 8
0x19189:      sub     ecx, 1
0x1918c:      pop     rbx
0x1918d:      pop     rbp
0x1918e:      imul    ecx, edx
0x19191:      movsxd  rcx, ecx
0x19194:      lea     rdi, [rax + rcx]
0x19198:      jmp     0x4bb0
0x4bb0
0x1919d:      nop     dword ptr [rax]
0x191a0:      endbr64
0x191a4:      mov     eax, dword ptr [rdi + 0xc]
0x191a7:      test    eax, eax
0x191a9:      je      0x191f8
0x191ab:      push    rbp
0x191ac:      sub     eax, 1
0x191af:      mov     rbp, rsi
0x191b2:      push    rbx
0x191b3:      mov     rbx, rdi
0x191b6:      sub     rsp, 8
0x191ba:      movsxd  rdx, dword ptr [rdi + 8]
0x191be:      mov     dword ptr [rdi + 0xc], eax
0x191c1:      mov     rsi, rdx
0x191c4:      imul    esi, eax
0x191c7:      movsxd  rsi, esi
0x191ca:      add     rsi, qword ptr [rdi]
0x191cd:      mov     rdi, rbp
0x191d0:      call    0x4bb0
0x191d5:      mov     esi, dword ptr [rbx + 0xc]
0x191d8:      imul    esi, dword ptr [rbx + 8]
0x191dc:      mov     rdi, qword ptr [rbx]
0x191df:      movsxd  rsi, esi
0x191e2:      call    0x1d690
0x191e7:      mov     qword ptr [rbx], rax
0x191ea:      add     rsp, 8
0x191ee:      mov     rax, rbp
0x191f1:      pop     rbx
0x191f2:      pop     rbp
0x191f3:      ret

```

```

0x191f4:    nop     dword ptr [rax]
0x191f8:    xor     eax, eax
0x191fa:    ret
0x191fb:    nop     dword ptr [rax + rax]
0x19200:    endbr64
0x19204:    mov     ecx, dword ptr [rdi + 0xc]
0x19207:    cmp     ecx, edx
0x19209:    jl      0x19240
0x1920b:    test    ecx, ecx
0x1920d:    je      0x19240
0x1920f:    sub     rsp, 8
0x19213:    movsxd  r9, dword ptr [rdi + 8]
0x19217:    sub     ecx, edx
0x19219:    mov     r8, rsi
0x1921c:    imul    ecx, r9d
0x19220:    mov     rdx, r9
0x19223:    movsxd  rsi, ecx
0x19226:    add     rsi, qword ptr [rdi]
0x19229:    mov     rdi, r8
0x1922c:    call    0x4bb0
0x19231:    add     rsp, 8
0x19235:    ret
0x19236:    nop     word ptr cs:[rax + rax]
0x19240:    xor     eax, eax
0x19242:    ret
0x19243:    nop     word ptr cs:[rax + rax]
0x1924e:    nop
0x19250:    endbr64
0x19254:    push    rbx
0x19255:    mov     rbx, rdi
0x19258:    mov     dword ptr [rdi + 0xc], esi
0x1925b:    imul    esi, dword ptr [rdi + 8]
0x1925f:    mov     rdi, qword ptr [rdi]
0x19262:    movsxd  rsi, esi
0x19265:    call    0x1d690
0x1926a:    mov     qword ptr [rbx], rax
0x1926d:    pop     rbx
0x1926e:    ret
0x1926f:    nop
0x19270:    endbr64
0x19274:    mov     eax, dword ptr [rdi + 0xc]
0x19277:    ret
0x19278:    nop     dword ptr [rax + rax]
0x19280:    push    r12
0x19282:    mov     r12, rdi
0x19285:    mov     edi, 2
0x1928a:    push    rbp
0x1928b:    push    rbx

```



```

0x1928c:    mov     ebx, esi
0x1928e:    xor     esi, esi
0x19290:    call    0x4b50
0x19295:    xor     esi, esi
0x19297:    mov     edi, 3
0x1929c:    call    0x4b50
0x192a1:    test    ebx, ebx
0x192a3:    jne     0x192e0
0x192a5:    mov     r9, qword ptr [rip + 0x115b4]
0x192ac:    xor     r8d, r8d
0x192af:    mov     rcx, r12
0x192b2:    xor     eax, eax
0x192b4:    lea     rdi, [rip + 0x7363]
0x192bb:    lea     rdx, [rip + 0x911e]
0x192c2:    lea     rsi, [rip + 0x735a]
0x192c9:    call    0x4e30
0x192ce:    mov     edi, 0x7f
0x192d3:    call    0x48a0
0x192d8:    nop     dword ptr [rax + rax]
0x192e0:    mov     edi, 1
0x192e5:    lea     rbp, [rip + 0x90ed]
0x192ec:    call    0x4ab0
0x192f1:    mov     edi, 2
0x192f6:    call    0x4ab0
0x192fb:    mov     rdi, qword ptr [rip + 0x127ce]
0x19302:    mov     rsi, rbp
0x19305:    call    0x4da0
0x1930a:    mov     rdi, qword ptr [rip + 0x127b7]
0x19311:    mov     rsi, rbp
0x19314:    call    0x4da0
0x19319:    mov     edx, 4
0x1931e:    lea     rsi, [rip + 0x90b6]
0x19325:    mov     rdi, r12
0x19328:    call    0x4890
0x1932d:    test    eax, eax
0x1932f:    je      0x192a5
0x19335:    xor     edi, edi
0x19337:    call    0x4ab0
0x1933c:    jmp     0x192a5
0x192a5:
0x19341:    nop     word ptr cs:[rax + rax]
0x1934c:    nop     dword ptr [rax]
0x19350:    endbr64
0x19354:    push    r12
0x19356:    push    rbp
0x19357:    push    rbx
0x19358:    sub     rsp, 0x10
0x1935c:    mov     rax, qword ptr fs:[0x28]

```

```

0x19365:    mov     qword ptr [rsp + 8], rax
0x1936a:    xor     eax, eax
0x1936c:    test    rdi, rdi
0x1936f:    je      0x193d0
0x19371:    mov     rbp, rdi
0x19374:    mov     r12d, esi
0x19377:    call    0x4ed0
0x1937c:    mov     ebx, eax
0x1937e:    test    eax, eax
0x19380:    js      0x193c0
0x19382:    je      0x193dc
0x19384:    lea     rbp, [rsp + 4]
0x19389:    nop     dword ptr [rax]
0x19390:    mov     rdi, rbp
0x19393:    call    0x4ec0
0x19398:    cmp     eax, ebx
0x1939a:    jne     0x19390
0x1939c:    mov     eax, dword ptr [rsp + 4]
0x193a0:    mov     rdx, qword ptr [rsp + 8]
0x193a5:    sub     rdx, qword ptr fs:[0x28]
0x193ae:    jne     0x193d7
0x193b0:    add     rsp, 0x10
0x193b4:    pop     rbx
0x193b5:    pop     rbp
0x193b6:    pop     r12
0x193b8:    ret
0x193b9:    nop     dword ptr [rax]
0x193c0:    mov     eax, 0xffffffff
0x193c5:    jmp     0x193a0
0x193a0:
0x193c7:    nop     word ptr [rax + rax]
0x193d0:    mov     eax, 1
0x193d5:    jmp     0x193a0
0x193a0:
0x193d7:    call    0x49d0
0x193dc:    mov     esi, r12d
0x193df:    mov     rdi, rbp
0x193e2:    xor     eax, eax
0x193e4:    call    0x19280
0x193e9:    nop     dword ptr [rax]
0x193f0:    endbr64
0x193f4:    push    r13
0x193f6:    mov     r13, rdi
0x193f9:    push    r12
0x193fb:    push    rbp
0x193fc:    push    rbx
0x193fd:    sub     rsp, 0x18
0x19401:    mov     rax, qword ptr fs:[0x28]

```

```

0x1940a:    mov     qword ptr [rsp + 8], rax
0x1940f:    xor     eax, eax
0x19411:    test    esi, esi
0x19413:    je      0x19488
0x19415:    mov     edi, 1
0x1941a:    mov     ebp, esi
0x1941c:    call    0x18b70
0x19421:    test    r13, r13
0x19424:    je      0x19520
0x1942a:    call    0x4ed0
0x1942f:    mov     r12d, eax
0x19432:    test    eax, eax
0x19434:    js      0x19550
0x1943a:    je      0x19560
0x19440:    lea     rbx, [rsp + 4]
0x19445:    nop     dword ptr [rax]
0x19448:    mov     rdi, rbx
0x1944b:    call    0x4ec0
0x19450:    cmp     r12d, eax
0x19453:    jne     0x19448
0x19455:    mov     r12d, dword ptr [rsp + 4]
0x1945a:    xor     edi, edi
0x1945c:    call    0x18b70
0x19461:    mov     rax, qword ptr [rsp + 8]
0x19466:    sub     rax, qword ptr fs:[0x28]
0x1946f:    jne     0x1955b
0x19475:    add     rsp, 0x18
0x19479:    mov     eax, r12d
0x1947c:    pop     rbx
0x1947d:    pop     rbp
0x1947e:    pop     r12
0x19480:    pop     r13
0x19482:    ret
0x19483:    nop     dword ptr [rax + rax]
0x19488:    xor     edi, edi
0x1948a:    call    0x9b00
0x1948f:    xor     eax, eax
0x19491:    call    0xa6c0
0x19496:    mov     rdi, qword ptr [rip + 0x1142b]
0x1949d:    call    0xbb40
0x194a2:    xor     edi, edi
0x194a4:    call    0x18b90
0x194a9:    test    r13, r13
0x194ac:    je      0x19530
0x194b2:    call    0x4ed0
0x194b7:    mov     ebp, eax
0x194b9:    test    eax, eax
0x194bb:    js      0x19540

```

```

0x194c1:      je      0x1956c
0x194c7:      lea     rbx, [rsp + 4]
0x194cc:      nop     dword ptr [rax]
0x194d0:      mov     rdi, rbx
0x194d3:      call    0x4ec0
0x194d8:      cmp     ebp, eax
0x194da:      jne     0x194d0
0x194dc:      mov     r12d, dword ptr [rsp + 4]
0x194e1:      mov     edi, 1
0x194e6:      call    0x18b90
0x194eb:      mov     edx, 2
0x194f0:      mov     edi, 1
0x194f5:      lea     rsi, [rip + 0x8ee7]
0x194fc:      call    0x18da0
0x19501:      mov     edi, 1
0x19506:      call    0x9b00
0x1950b:      xor     eax, eax
0x1950d:      call    0xa6c0
0x19512:      xor     edi, edi
0x19514:      call    0x7890
0x19519:      jmp     0x19461
0x19461
0x1951e:      nop
0x19520:      mov     r12d, 1
0x19526:      jmp     0x1945a
0x1945a
0x1952b:      nop     dword ptr [rax + rax]
0x19530:      mov     r12d, 1
0x19536:      jmp     0x194e1
0x194e1
0x19538:      nop     dword ptr [rax + rax]
0x19540:      mov     r12d, 0xffffffff
0x19546:      jmp     0x194e1
0x194e1
0x19548:      nop     dword ptr [rax + rax]
0x19550:      mov     r12d, 0xffffffff
0x19556:      jmp     0x1945a
0x1945a
0x1955b:      call    0x49d0
0x19560:      mov     esi, ebp
0x19562:      mov     rdi, r13
0x19565:      xor     eax, eax
0x19567:      call    0x19280
0x1956c:      xor     esi, esi
0x1956e:      mov     rdi, r13
0x19571:      xor     eax, eax
0x19573:      call    0x19280
0x19578:      nop     dword ptr [rax + rax]

```

```

0x19580:      endbr64
0x19584:      mov     rdi, qword ptr [rip + 0x12545]
0x1958b:      test    rdi, rdi
0x1958e:      je       0x195b8
0x19590:      sub     rsp, 8
0x19594:      call    0x4870
0x19599:      mov     rdi, qword ptr [rip + 0x12528]
0x195a0:      test    rdi, rdi
0x195a3:      je       0x195b0
0x195a5:      add     rsp, 8
0x195a9:      jmp     0x4870
0x4870
0x195ae:      nop
0x195b0:      add     rsp, 8
0x195b4:      ret
0x195b5:      nop     dword ptr [rax]
0x195b8:      mov     rdi, qword ptr [rip + 0x12509]
0x195bf:      test    rdi, rdi
0x195c2:      je       0x195d0
0x195c4:      jmp     0x4870
0x4870
0x195c9:      nop     dword ptr [rax]
0x195d0:      ret
0x195d1:      nop     word ptr cs:[rax + rax]
0x195dc:      nop     dword ptr [rax]
0x195e0:      endbr64
0x195e4:      push    r12
0x195e6:      lea     rsi, [rip + 0x89dc]
0x195ed:      push    rbp
0x195ee:      mov     rbp, rdi
0x195f1:      sub     rsp, 8
0x195f5:      mov     rdi, qword ptr [rip + 0x124cc]
0x195fc:      call    0x4da0
0x19601:      test    rax, rax
0x19604:      je       0x19676
0x19606:      mov     edi, 0x801
0x1960b:      mov     r12, rax
0x1960e:      call    0x1d670
0x19613:      mov     rbp, rax
0x19616:      jmp     0x1964a
0x1964a
0x19618:      nop     dword ptr [rax + rax]
0x19620:      mov     rdi, rbp
0x19623:      call    0x49b0
0x19628:      cdqe
0x1962a:      lea     rax, [rbp + rax - 1]
0x1962f:      cmp     byte ptr [rax], 0xa
0x19632:      jne     0x19637

```

```

0x19634:      mov     byte ptr [rax], 0
0x19637:      xor     esi, esi
0x19639:      mov     edx, 0xa
0x1963e:      mov     rdi, rbp
0x19641:      call    0x8070
0x19646:      test    al, al
0x19648:      je      0x1965f
0x1964a:      mov     rdx, r12
0x1964d:      mov     esi, 0x801
0x19652:      mov     rdi, rbp
0x19655:      call    0x4b00
0x1965a:      test    rax, rax
0x1965d:      jne     0x19620
0x1965f:      mov     rdi, rbp
0x19662:      call    0x18d10
0x19667:      add     rsp, 8
0x1966b:      mov     rdi, r12
0x1966e:      pop     rbp
0x1966f:      pop     r12
0x19671:      jmp     0x4940
0x4940
0x19676:      mov     rdi, rbp
0x19679:      call    0x49b0
0x1967e:      mov     rdi, qword ptr [rip + 0x12443]
0x19685:      mov     r12, rax
0x19688:      call    0x49b0
0x1968d:      lea     rdi, [r12 + rax + 0x20]
0x19692:      call    0x1d670
0x19697:      mov     r9, qword ptr [rip + 0x1242a]
0x1969e:      mov     r8, rbp
0x196a1:      or      rdx, 0xffffffffffffffff
0x196a5:      mov     r12, rax
0x196a8:      mov     rdi, rax
0x196ab:      mov     esi, 1
0x196b0:      xor     eax, eax
0x196b2:      lea     rcx, [rip + 0x8d2d]
0x196b9:      call    0x4f10
0x196be:      mov     rdi, r12
0x196c1:      mov     edx, 0x1e
0x196c6:      xor     esi, esi
0x196c8:      call    0x8070
0x196cd:      pop     rax
0x196ce:      mov     rdi, r12
0x196d1:      pop     rbp
0x196d2:      pop     r12
0x196d4:      jmp     0x18d10
0x18d10
0x196d9:      nop     dword ptr [rax]

```

```

0x196e0:      endbr64
0x196e4:      push     r15
0x196e6:      push     r14
0x196e8:      push     r13
0x196ea:      push     r12
0x196ec:      push     rbp
0x196ed:      push     rbx
0x196ee:      sub      rsp, 8
0x196f2:      test     rdi, rdi
0x196f5:      je       0x19878
0x196fb:      mov      rbp, rdi
0x196fe:      call     0x49b0
0x19703:      lea      rdi, [rax + 1]
0x19707:      call     0x1d670
0x1970c:      mov      rsi, rbp
0x1970f:      mov      rdi, rax
0x19712:      call     0x48b0
0x19717:      cmp      byte ptr [rax], 0x7e
0x1971a:      mov      rbp, rax
0x1971d:      mov      r12, rax
0x19720:      jne      0x19800
0x19726:      movzx    eax, byte ptr [rax + 1]
0x1972a:      lea      r13, [rbp + 1]
0x1972e:      test     al, al
0x19730:      je       0x19818
0x19736:      cmp      al, 0x2f
0x19738:      je       0x19818
0x1973e:      mov      rdi, rbp
0x19741:      call     0x49b0
0x19746:      mov      rdi, rax
0x19749:      call     0x1d670
0x1974e:      movzx    ecx, byte ptr [rbp + 1]
0x19752:      mov      r14, rax
0x19755:      cmp      cl, 0x2f
0x19758:      je       0x19938
0x1975e:      test     cl, cl
0x19760:      je       0x19938
0x19766:      mov      edx, 2
0x1976b:      nop      dword ptr [rax + rax]
0x19770:      mov      byte ptr [r14 + rdx - 2], cl
0x19775:      movzx    ecx, byte ptr [rbp + rdx]
0x1977a:      lea      r13, [rbp + rdx]
0x1977f:      mov      rax, rdx
0x19782:      add      rdx, 1
0x19786:      test     cl, cl
0x19788:      je       0x1978f
0x1978a:      cmp      cl, 0x2f
0x1978d:      jne      0x19770

```

```

0x1978f:    lea     rax, [r14 + rax - 1]
0x19794:    mov     byte ptr [rax], 0
0x19797:    mov     rdi, r14
0x1979a:    call    0x4b60
0x1979f:    mov     rbx, rax
0x197a2:    test    rax, rax
0x197a5:    je      0x19880
0x197ab:    mov     rdi, qword ptr [rax + 0x20]
0x197af:    call    0x49b0
0x197b4:    mov     rdi, r13
0x197b7:    mov     r15, rax
0x197ba:    call    0x49b0
0x197bf:    lea     edi, [r15 + 1]
0x197c3:    movsxd  rdi, edi
0x197c6:    add     rdi, rax
0x197c9:    call    0x1d670
0x197ce:    mov     rsi, qword ptr [rbx + 0x20]
0x197d2:    mov     rdi, rax
0x197d5:    mov     r12, rax
0x197d8:    call    0x48b0
0x197dd:    movsxd  rdi, r15d
0x197e0:    mov     rsi, r13
0x197e3:    add     rdi, r12
0x197e6:    call    0x48b0
0x197eb:    mov     rdi, rbp
0x197ee:    call    0x18d10
0x197f3:    call    0x4c70
0x197f8:    mov     rdi, r14
0x197fb:    call    0x18d10
0x19800:    add     rsp, 8
0x19804:    mov     rax, r12
0x19807:    pop     rbx
0x19808:    pop     rbp
0x19809:    pop     r12
0x1980b:    pop     r13
0x1980d:    pop     r14
0x1980f:    pop     r15
0x19811:    ret
0x19812:    nop     word ptr [rax + rax]
0x19818:    lea     rdi, [rip + 0x6d41]
0x1981f:    call    0x47e0
0x19824:    mov     r14, rax
0x19827:    test    rax, rax
0x1982a:    je      0x198f8
0x19830:    mov     rdi, r14
0x19833:    call    0x49b0
0x19838:    mov     rdi, r13
0x1983b:    mov     rbx, rax

```



```

0x1983e:      call    0x49b0
0x19843:      movsxd  rbx, ebx
0x19846:      lea     rdi, [rbx + rax + 1]
0x1984b:      call    0x1d670
0x19850:      mov     rsi, r14
0x19853:      mov     r12, rax
0x19856:      mov     rdi, rax
0x19859:      call    0x48b0
0x1985e:      lea     rdi, [r12 + rbx]
0x19862:      mov     rsi, r13
0x19865:      call    0x48b0
0x1986a:      mov     rdi, rbp
0x1986d:      call    0x18d10
0x19872:      jmp     0x19800
0x19800
0x19874:      nop     dword ptr [rax]
0x19878:      xor     r12d, r12d
0x1987b:      jmp     0x19800
0x19800
0x1987d:      nop     dword ptr [rax]
0x19880:      mov     rdx, qword ptr [rip + 0x12251]
0x19887:      test    rdx, rdx
0x1988a:      je      0x197f3
0x19890:      mov     rdi, r14
0x19893:      xor     eax, eax
0x19895:      call    rdx
0x19897:      mov     r15, rax
0x1989a:      test    rax, rax
0x1989d:      je      0x197f3
0x198a3:      mov     rdi, rax
0x198a6:      call    0x49b0
0x198ab:      mov     rdi, r13
0x198ae:      mov     rbx, rax
0x198b1:      call    0x49b0
0x198b6:      lea     edi, [rbx + 1]
0x198b9:      movsxd  rdi, edi
0x198bc:      add     rdi, rax
0x198bf:      call    0x1d670
0x198c4:      mov     rsi, r15
0x198c7:      mov     rdi, rax
0x198ca:      mov     r12, rax
0x198cd:      call    0x48b0
0x198d2:      movsxd  rdi, ebx
0x198d5:      mov     rsi, r13
0x198d8:      add     rdi, r12
0x198db:      call    0x48b0
0x198e0:      mov     rdi, r15
0x198e3:      call    0x18d10

```

```

0x198e8:      mov     rdi, rbp
0x198eb:      call   0x18d10
0x198f0:      jmp     0x197f3
0x197f3:
0x198f5:      nop     dword ptr [rax]
0x198f8:      call   0x49e0
0x198fd:      mov     edi, eax
0x198ff:      call   0x4970
0x19904:      test    rax, rax
0x19907:      je      0x19916
0x19909:      mov     r14, qword ptr [rax + 0x20]
0x1990d:      test    r14, r14
0x19910:      jne     0x19830
0x19916:      mov     rdi, r13
0x19919:      call   0x49b0
0x1991e:      lea     rdi, [rax + 1]
0x19922:      call   0x1d670
0x19927:      mov     r12, rax
0x1992a:      mov     rdi, rax
0x1992d:      jmp     0x19862
0x19862:
0x19932:      nop     word ptr [rax + rax]
0x19938:      mov     rax, r14
0x1993b:      jmp     0x19794
0x19794:
0x19940:      endbr64
0x19944:      push    r15
0x19946:      push    r14
0x19948:      mov     r14, rdi
0x1994b:      push    r13
0x1994d:      push    r12
0x1994f:      push    rbp
0x19950:      push    rbx
0x19951:      xor     ebx, ebx
0x19953:      sub     rsp, 0x48
0x19957:      mov     dword ptr [rsp + 0x1c], 0
0x1995f:      mov     qword ptr [rsp + 0x10], 0
0x19968:      nop     dword ptr [rax + rax]
0x19970:      mov     r15, qword ptr [rip + 0x10e31]
0x19977:      mov     rdi, r14
0x1997a:      mov     qword ptr [rsp + 8], r15
0x1997f:      call   0x49b0
0x19984:      mov     qword ptr [rsp + 0x20], rax
0x19989:      mov     rcx, rax
0x1998c:      movzx   eax, byte ptr [r14]
0x19990:      test    al, al
0x19992:      je      0x19bd0
0x19998:      cmp     al, 0x7e

```

```

0x1999a:      je      0x19bd0
0x199a0:      mov     dword ptr [rsp + 0x18], ecx
0x199a4:      mov     rax, rcx
0x199a7:      test    r15, r15
0x199aa:      je      0x19c50
0x199b0:      mov     edx, dword ptr [rsp + 0x20]
0x199b4:      test    edx, edx
0x199b6:      jle     0x19c4a
0x199bc:      mov     rax, qword ptr [rsp + 8]
0x199c1:      mov     dword ptr [rsp + 0x3c], ebx
0x199c5:      xor     r12d, r12d
0x199c8:      mov     rax, qword ptr [rax]
0x199cb:      mov     qword ptr [rsp + 0x28], rax
0x199d0:      mov     rax, qword ptr [rsp + 0x20]
0x199d5:      sub     eax, 1
0x199d8:      mov     qword ptr [rsp + 0x30], rax
0x199dd:      mov     rax, qword ptr [rsp + 8]
0x199e2:      mov     dword ptr [rsp + 0x38], r12d
0x199e7:      lea     rbp, [r14 + r12]
0x199eb:      lea     rbx, [rax + 8]
0x199ef:      mov     rax, qword ptr [rsp + 0x28]
0x199f4:      mov     r13, rax
0x199f7:      test    rax, rax
0x199fa:      jne     0x19a18
0x199fc:      jmp     0x19bb0
0x19bb0:
0x19a01:      nop     dword ptr [rax]
0x19a08:      mov     r13, qword ptr [rbx]
0x19a0b:      add     rbx, 8
0x19a0f:      test    r13, r13
0x19a12:      je      0x19bb0
0x19a18:      mov     rdi, r13
0x19a1b:      call    0x49b0
0x19a20:      mov     rsi, r13
0x19a23:      mov     rdi, rbp
0x19a26:      mov     rdx, rax
0x19a29:      mov     r15, rax
0x19a2c:      call    0x4890
0x19a31:      test    eax, eax
0x19a33:      jne     0x19a08
0x19a35:      mov     eax, dword ptr [rsp + 0x38]
0x19a39:      mov     ebx, dword ptr [rsp + 0x3c]
0x19a3d:      lea     eax, [rax + r15 - 1]
0x19a42:      lea     ecx, [rbx + rax]
0x19a45:      mov     dword ptr [rsp + 0x18], eax
0x19a49:      movsxd  r13, eax
0x19a4c:      mov     dword ptr [rsp + 8], ecx
0x19a50:      lea     r15, [r14 + r13]

```

```

0x19a54:    mov     ecx, dword ptr [rsp + 8]
0x19a58:    cmp     dword ptr [rsp + 0x1c], ecx
0x19a5c:    jle     0x19bf0
0x19a62:    movsxd  rbx, ebx
0x19a65:    add     rbx, qword ptr [rsp + 0x10]
0x19a6a:    mov     rdx, r13
0x19a6d:    mov     rsi, r14
0x19a70:    mov     rdi, rbx
0x19a73:    mov     qword ptr [rsp + 0x28], rbx
0x19a78:    call    0x4880
0x19a7d:    mov     rdi, r15
0x19a80:    mov     r13, qword ptr [rip + 0x10d19]
0x19a87:    call    0x49b0
0x19a8c:    mov     dword ptr [rsp + 0x20], eax
0x19a90:    test    eax, eax
0x19a92:    jle     0x19c64
0x19a98:    mov     rbx, r15
0x19a9b:    xor     ebp, ebp
0x19a9d:    movzx   eax, byte ptr [rbx]
0x19aa0:    cmp     al, 0x2f
0x19aa2:    je      0x19b80
0x19aa8:    test    al, al
0x19aaa:    je      0x19b80
0x19ab0:    test    r13, r13
0x19ab3:    je      0x19aea
0x19ab5:    mov     r14, r13
0x19ab8:    jmp     0x19ae2
0x19ae2:
0x19aba:    nop     word ptr [rax + rax]
0x19ac0:    mov     rdi, r12
0x19ac3:    add     r14, 8
0x19ac7:    call    0x49b0
0x19acc:    mov     rsi, r12
0x19acf:    mov     rdi, rbx
0x19ad2:    mov     rdx, rax
0x19ad5:    call    0x4890
0x19ada:    test    eax, eax
0x19adc:    je      0x19b80
0x19ae2:    mov     r12, qword ptr [r14]
0x19ae5:    test    r12, r12
0x19ae8:    jne     0x19ac0
0x19aea:    add     ebp, 1
0x19aed:    add     rbx, 1
0x19af1:    cmp     dword ptr [rsp + 0x20], ebp
0x19af5:    jne     0x19a9d
0x19af7:    nop     word ptr [rax + rax]
0x19b00:    lea     edi, [rbp + 1]
0x19b03:    movsxd  rbp, ebp

```

```

0x19b06:      movsxd  rdi, edi
0x19b09:      lea     r14, [r15 + rbp]
0x19b0d:      call    0x1d670
0x19b12:      mov     rdx, rbp
0x19b15:      mov     rsi, r15
0x19b18:      mov     r13, rax
0x19b1b:      mov     rdi, rax
0x19b1e:      call    0x4880
0x19b23:      mov     byte ptr [r13 + rbp], 0
0x19b29:      mov     rdi, r13
0x19b2c:      xor     eax, eax
0x19b2e:      call    0x196e0
0x19b33:      mov     rdi, r13
0x19b36:      mov     rbp, rax
0x19b39:      call    0x18d10
0x19b3e:      mov     rdi, rbp
0x19b41:      call    0x49b0
0x19b46:      mov     ecx, dword ptr [rsp + 8]
0x19b4a:      lea     ebx, [rcx + rax]
0x19b4d:      cmp     ebx, dword ptr [rsp + 0x1c]
0x19b51:      jge     0x19c20
0x19b57:      movsxd  rdi, dword ptr [rsp + 8]
0x19b5c:      mov     rsi, rbp
0x19b5f:      add     rdi, qword ptr [rsp + 0x10]
0x19b64:      call    0x48b0
0x19b69:      mov     rdi, rbp
0x19b6c:      call    0x18d10
0x19b71:      jmp     0x19970
0x19970:
0x19b76:      nop     word ptr cs:[rax + rax]
0x19b80:      or      dword ptr [rsp + 0x18], ebp
0x19b84:      mov     eax, dword ptr [rsp + 0x18]
0x19b88:      test    eax, eax
0x19b8a:      jne     0x19b00
0x19b90:      mov     rax, qword ptr [rsp + 0x28]
0x19b95:      mov     byte ptr [rax], 0
0x19b98:      mov     rax, qword ptr [rsp + 0x10]
0x19b9d:      add     rsp, 0x48
0x19ba1:      pop     rbx
0x19ba2:      pop     rbp
0x19ba3:      pop     r12
0x19ba5:      pop     r13
0x19ba7:      pop     r14
0x19ba9:      pop     r15
0x19bab:      ret
0x19bac:      nop     dword ptr [rax]
0x19bb0:      lea     rax, [r12 + 1]
0x19bb5:      cmp     qword ptr [rsp + 0x30], r12

```

```

0x19bba:      je      0x19c46
0x19bc0:      mov     r12, rax
0x19bc3:      jmp     0x199dd
0x199dd
0x19bc8:      nop     dword ptr [rax + rax]
0x19bd0:      mov     dword ptr [rsp + 8], ebx
0x19bd4:      mov     r15, r14
0x19bd7:      mov     ecx, dword ptr [rsp + 8]
0x19bdb:      xor     r13d, r13d
0x19bde:      mov     dword ptr [rsp + 0x18], 0
0x19be6:      cmp     dword ptr [rsp + 0x1c], ecx
0x19bea:      jg       0x19a62
0x19bf0:      mov     eax, dword ptr [rsp + 0x1c]
0x19bf4:      mov     ecx, dword ptr [rsp + 0x18]
0x19bf8:      mov     rdi, qword ptr [rsp + 0x10]
0x19bfd:      lea     eax, [rax + rcx + 0x14]
0x19c01:      lea     esi, [rax + 1]
0x19c04:      mov     dword ptr [rsp + 0x1c], eax
0x19c08:      movsxd  rsi, esi
0x19c0b:      call    0x1d690
0x19c10:      mov     qword ptr [rsp + 0x10], rax
0x19c15:      jmp     0x19a62
0x19a62
0x19c1a:      nop     word ptr [rax + rax]
0x19c20:      mov     ecx, dword ptr [rsp + 0x1c]
0x19c24:      mov     rdi, qword ptr [rsp + 0x10]
0x19c29:      lea     eax, [rcx + rax + 0x14]
0x19c2d:      lea     esi, [rax + 1]
0x19c30:      mov     dword ptr [rsp + 0x1c], eax
0x19c34:      movsxd  rsi, esi
0x19c37:      call    0x1d690
0x19c3c:      mov     qword ptr [rsp + 0x10], rax
0x19c41:      jmp     0x19b57
0x19b57
0x19c46:      mov     ebx, dword ptr [rsp + 0x3c]
0x19c4a:      mov     rax, qword ptr [rsp + 0x20]
0x19c4f:      nop
0x19c50:      mov     ecx, eax
0x19c52:      movsxd  r13, eax
0x19c55:      add     ecx, ebx
0x19c57:      lea     r15, [r14 + r13]
0x19c5b:      mov     dword ptr [rsp + 8], ecx
0x19c5f:      jmp     0x19a54
0x19a54
0x19c64:      xor     ebp, ebp
0x19c66:      jmp     0x19b84
0x19b84
0x19c6b:      nop     dword ptr [rax + rax]

```

```

0x19c70:    push    r15
0x19c72:    mov     r15, rsi
0x19c75:    push    r14
0x19c77:    mov     r14d, edx
0x19c7a:    push    r13
0x19c7c:    push    r12
0x19c7e:    mov     r12, rdi
0x19c81:    push    rbp
0x19c82:    push    rbx
0x19c83:    sub     rsp, 0x18
0x19c87:    movsxd  rsi, dword ptr [rsi]
0x19c8a:    lea     r13, [rdi + rsi]
0x19c8e:    mov     edi, ecx
0x19c90:    mov     ebx, esi
0x19c92:    mov     rbp, rsi
0x19c95:    movzx   eax, byte ptr [r13]
0x19c9a:    and     edi, 1
0x19c9d:    mov     rdx, rsi
0x19ca0:    test    al, al
0x19ca2:    jne     0x19cdc
0x19ca4:    jmp     0x19db9
0x19db9:
0x19ca9:    nop     dword ptr [rax]
0x19cb0:    cmp     al, 0x5c
0x19cb2:    jne     0x19cc5
0x19cb4:    movsx   eax, byte ptr [r12 + rdx + 1]
0x19cba:    cmp     eax, r14d
0x19cbd:    sete    al
0x19cc0:    movzx   eax, al
0x19cc3:    add     ebx, eax
0x19cc5:    add     ebx, 1
0x19cc8:    movsxd  rdx, ebx
0x19ccb:    lea     r13, [r12 + rdx]
0x19ccf:    movzx   eax, byte ptr [r13]
0x19cd4:    test    al, al
0x19cd6:    je      0x19da0
0x19cdc:    movsx   ecx, al
0x19cdf:    cmp     ecx, r14d
0x19ce2:    jne     0x19cb0
0x19ce4:    cmp     ebp, ebx
0x19ce6:    jl      0x19d10
0x19ce8:    test    dil, dil
0x19ceb:    jne     0x19d10
0x19ced:    movzx   eax, byte ptr [r13]
0x19cf2:    xor     r9d, r9d
0x19cf5:    cmp     al, 1
0x19cf7:    sbb     ebx, -1
0x19cfa:    mov     dword ptr [r15], ebx

```

```

0x19cfd:      add     rsp, 0x18
0x19d01:      mov     rax, r9
0x19d04:      pop     rbx
0x19d05:      pop     rbp
0x19d06:      pop     r12
0x19d08:      pop     r13
0x19d0a:      pop     r14
0x19d0c:      pop     r15
0x19d0e:      ret
0x19d0f:      nop
0x19d10:      mov     edi, ebx
0x19d12:      mov     qword ptr [rsp], r8
0x19d16:      sub     edi, ebp
0x19d18:      mov     qword ptr [rsp + 8], rsi
0x19d1d:      add     edi, 1
0x19d20:      movsxd  rdi, edi
0x19d23:      call    0x1d670
0x19d28:      cmp     ebp, ebx
0x19d2a:      mov     r8, qword ptr [rsp]
0x19d2e:      mov     r9, rax
0x19d31:      jge     0x19dd6
0x19d37:      mov     rsi, qword ptr [rsp + 8]
0x19d3c:      mov     eax, 1
0x19d41:      jmp     0x19d5d
0x19d5d
0x19d43:      nop     dword ptr [rax + rax]
0x19d48:      add     ebp, 1
0x19d4b:      mov     byte ptr [r9 + rax - 1], dl
0x19d50:      mov     edx, eax
0x19d52:      add     rax, 1
0x19d56:      cmp     ebp, ebx
0x19d58:      jge     0x19d80
0x19d5a:      movsxd  rsi, ebp
0x19d5d:      movzx   edx, byte ptr [r12 + rsi]
0x19d62:      cmp     dl, 0x5c
0x19d65:      jne     0x19d48
0x19d67:      movsx   esi, byte ptr [r12 + rsi + 1]
0x19d6d:      cmp     esi, r14d
0x19d70:      jne     0x19d48
0x19d72:      add     ebp, 1
0x19d75:      mov     edx, esi
0x19d77:      jmp     0x19d48
0x19d48
0x19d79:      nop     dword ptr [rax]
0x19d80:      movsxd  rax, edx
0x19d83:      add     rax, r9
0x19d86:      mov     byte ptr [rax], 0
0x19d89:      mov     dword ptr [r8], edx

```



```

0x19d8c:    movzx    eax, byte ptr [r13]
0x19d91:    jmp     0x19cf5
0x19cf5:
0x19d96:    nop     word ptr cs:[rax + rax]
0x19da0:    cmp     ebp, ebx
0x19da2:    jl      0x19d10
0x19da8:    test    dil, dil
0x19dab:    jne     0x19d10
0x19db1:    xor     r9d, r9d
0x19db4:    jmp     0x19cfa
0x19cfa:
0x19db9:    test    ecx, ecx
0x19dbb:    je      0x19db1
0x19dbd:    mov     edi, 1
0x19dc2:    mov     qword ptr [rsp], r8
0x19dc6:    call    0x1d670
0x19dcb:    mov     r8, qword ptr [rsp]
0x19dcf:    xor     edx, edx
0x19dd1:    mov     r9, rax
0x19dd4:    jmp     0x19d86
0x19d86:
0x19dd6:    xor     edx, edx
0x19dd8:    jmp     0x19d86
0x19d86:
0x19dda:    nop     word ptr [rax + rax]
0x19de0:    push    r15
0x19de2:    push    r14
0x19de4:    push    r13
0x19de6:    push    r12
0x19de8:    push    rbp
0x19de9:    push    rbx
0x19dea:    sub     rsp, 0x28
0x19dee:    mov     eax, dword ptr [rip + 0x11cf0]
0x19df4:    mov     ebx, dword ptr [rip + 0x11d02]
0x19dfa:    mov     dword ptr [rsp + 8], esi
0x19dfe:    mov     dword ptr [rsp + 0x18], edx
0x19e02:    mov     dword ptr [rsp + 0x1c], eax
0x19e06:    test    eax, eax
0x19e08:    je      0x19f9d
0x19e0e:    mov     rbp, rdi
0x19e11:    cmp     eax, ebx
0x19e13:    je      0x19fb0
0x19e19:    call    0x49b0
0x19e1e:    mov     ecx, dword ptr [rsp + 8]
0x19e22:    mov     dword ptr [rsp + 0xc], eax
0x19e26:    test    ecx, ecx
0x19e28:    js      0x19fbb
0x19e2e:    mov     rax, qword ptr [rip + 0x11ccb]

```

```

0x19e35:      movsxd  r14, dword ptr [rsp + 0xc]
0x19e3a:      mov     qword ptr [rsp + 0x10], rax
0x19e3f:      mov     edx, dword ptr [rsp + 8]
0x19e43:      test    edx, edx
0x19e45:      js      0x19f16
0x19e4b:      cmp     dword ptr [rsp + 0x1c], ebx
0x19e4f:      je      0x19f9d
0x19e55:      mov     rcx, qword ptr [rsp + 0x10]
0x19e5a:      movsxd  rax, ebx
0x19e5d:      mov     rax, qword ptr [rcx + rax*8]
0x19e61:      mov     r13, qword ptr [rax]
0x19e64:      mov     rdi, r13
0x19e67:      call    0x49b0
0x19e6c:      mov     r10, rax
0x19e6f:      cmp     dword ptr [rsp + 0xc], eax
0x19e73:      jg      0x19f07
0x19e79:      cmp     dword ptr [rsp + 0x18], 1
0x19e7e:      movzx   r12d, byte ptr [rbp]
0x19e83:      je      0x19ee0
0x19e85:      mov     r15d, dword ptr [rsp + 0x18]
0x19e8a:      sub     r10d, dword ptr [rsp + 0xc]
0x19e8f:      jmp     0x19ea5
0x19ea5:
0x19e91:      nop     dword ptr [rax]
0x19e98:      add     r15d, 1
0x19e9c:      add     r13, 1
0x19ea0:      cmp     r15d, r10d
0x19ea3:      jg      0x19f07
0x19ea5:      cmp     byte ptr [r13], r12b
0x19ea9:      jne     0x19e98
0x19eab:      mov     rdx, r14
0x19eae:      mov     rsi, r13
0x19eb1:      mov     rdi, rbp
0x19eb4:      mov     dword ptr [rsp + 4], r10d
0x19eb9:      call    0x4890
0x19ebe:      mov     r10d, dword ptr [rsp + 4]
0x19ec3:      test    eax, eax
0x19ec5:      jne     0x19e98
0x19ec7:      mov     dword ptr [rip + 0x11c2f], ebx
0x19ecd:      add     rsp, 0x28
0x19ed1:      mov     eax, r15d
0x19ed4:      pop     rbx
0x19ed5:      pop     rbp
0x19ed6:      pop     r12
0x19ed8:      pop     r13
0x19eda:      pop     r14
0x19edc:      pop     r15
0x19ede:      ret

```

```

0x19edf:      nop
0x19ee0:      cmp     byte ptr [r13], r12b
0x19ee4:      jne     0x19efb
0x19ee6:      mov     rdx, r14
0x19ee9:      mov     rsi, r13
0x19eec:      mov     rdi, rbp
0x19eef:      call    0x4890
0x19ef4:      mov     r15d, eax
0x19ef7:      test    eax, eax
0x19ef9:      je      0x19ec7
0x19efb:      mov     eax, dword ptr [rsp + 8]
0x19eff:      test    eax, eax
0x19f01:      js      0x19f90
0x19f07:      mov     edx, dword ptr [rsp + 8]
0x19f0b:      add     ebx, 1
0x19f0e:      test    edx, edx
0x19f10:      jns     0x19e4b
0x19f16:      mov     rcx, qword ptr [rsp + 0x10]
0x19f1b:      movsxd  rax, ebx
0x19f1e:      mov     rax, qword ptr [rcx + rax*8]
0x19f22:      mov     r13, qword ptr [rax]
0x19f25:      mov     rdi, r13
0x19f28:      call    0x49b0
0x19f2d:      mov     r8, rax
0x19f30:      cmp     dword ptr [rsp + 0xc], eax
0x19f34:      jg      0x19f90
0x19f36:      cmp     dword ptr [rsp + 0x18], 1
0x19f3b:      movzx   r12d, byte ptr [rbp]
0x19f40:      je      0x19ee0
0x19f42:      mov     r15d, r8d
0x19f45:      sub     r15d, dword ptr [rsp + 0xc]
0x19f4a:      movsxd  rcx, r15d
0x19f4d:      add     r13, rcx
0x19f50:      jmp     0x19f66
0x19f66:
0x19f52:      nop     word ptr [rax + rax]
0x19f58:      sub     r15d, 1
0x19f5c:      sub     r13, 1
0x19f60:      cmp     r15d, -1
0x19f64:      je      0x19f90
0x19f66:      cmp     byte ptr [r13], r12b
0x19f6a:      jne     0x19f58
0x19f6c:      mov     rdx, r14
0x19f6f:      mov     rsi, r13
0x19f72:      mov     rdi, rbp
0x19f75:      call    0x4890
0x19f7a:      test    eax, eax
0x19f7c:      je      0x19ec7

```

```

0x19f82:      sub     r15d, 1
0x19f86:      sub     r13, 1
0x19f8a:      cmp     r15d, -1
0x19f8e:      jne     0x19f66
0x19f90:      sub     ebx, 1
0x19f93:      test    dword ptr [rsp + 8], ebx
0x19f97:      jns     0x19e3f
0x19f9d:      mov     r15d, 0xffffffff
0x19fa3:      jmp     0x19ecd
0x19ecd
0x19fa8:      nop     dword ptr [rax + rax]
0x19fb0:      test    esi, esi
0x19fb2:      jns     0x19f9d
0x19fb4:      mov     eax, dword ptr [rsp + 0x1c]
0x19fb8:      lea     ebx, [rax - 1]
0x19fbb:      mov     rdi, rbp
0x19fbe:      call    0x49b0
0x19fc3:      mov     dword ptr [rsp + 0xc], eax
0x19fc7:      test    dword ptr [rsp + 8], ebx
0x19fcb:      jns     0x19e2e
0x19fd1:      jmp     0x19f9d
0x19f9d
0x19fd3:      nop     word ptr cs:[rax + rax]
0x19fde:      nop
0x19fe0:      push    r14
0x19fe2:      push    r13
0x19fe4:      movsxd  r13, esi
0x19fe7:      sub     edx, r13d
0x19fea:      push    r12
0x19fec:      push    rbp
0x19fed:      mov     rbp, rdi
0x19ff0:      push    rbx
0x19ff1:      movsxd  rbx, edx
0x19ff4:      cmp     ecx, 2
0x19ff7:      je      0x1a070
0x19ff9:      mov     eax, 0x1d
0x19ffe:      lea     r14, [rip + 0x843a]
0x1a005:      cmp     ecx, 3
0x1a008:      je      0x1a028
0x1a00a:      xor     eax, eax
0x1a00c:      lea     r14, [rip + 0x83f5]
0x1a013:      lea     rdx, [rip + 0x83fe]
0x1a01a:      cmp     ecx, 1
0x1a01d:      sete    al
0x1a020:      cmov     r14, rdx
0x1a024:      lea     eax, [rax + rax*2 + 0xf]
0x1a028:      lea     edi, [rax + rbx + 3]
0x1a02c:      movsxd  rdi, edi

```

```

0x1a02f:      call    0x1d670
0x1a034:      lea     rsi, [rbp + r13]
0x1a039:      mov     rdx, rbx
0x1a03c:      mov     r12, rax
0x1a03f:      mov     rdi, rax
0x1a042:      call    0x4880
0x1a047:      mov     byte ptr [r12 + rbx], 0x3a
0x1a04c:      lea     rdi, [r12 + rbx + 2]
0x1a051:      mov     rsi, r14
0x1a054:      mov     byte ptr [r12 + rbx + 1], 0x20
0x1a05a:      call    0x48b0
0x1a05f:      pop     rbx
0x1a060:      mov     rax, r12
0x1a063:      pop     rbp
0x1a064:      pop     r12
0x1a066:      pop     r13
0x1a068:      pop     r14
0x1a06a:      ret
0x1a06b:      nop     dword ptr [rax + rax]
0x1a070:      mov     eax, 0x13
0x1a075:      lea     r14, [rip + 0x83af]
0x1a07c:      jmp     0x1a028
0x1a028
0x1a07e:      nop
0x1a080:      push    r12
0x1a082:      push    rbp
0x1a083:      push    rbx
0x1a084:      test    rdi, rdi
0x1a087:      je      0x1a0b0
0x1a089:      mov     rbp, rdi
0x1a08c:      call    0x49b0
0x1a091:      lea     rdi, [rax + 1]
0x1a095:      call    0x1d670
0x1a09a:      pop     rbx
0x1a09b:      mov     rsi, rbp
0x1a09e:      pop     rbp
0x1a09f:      mov     rdi, rax
0x1a0a2:      pop     r12
0x1a0a4:      jmp     0x48b0
0x48b0
0x1a0a9:      nop     dword ptr [rax]
0x1a0b0:      lea     rdi, [rip + 0x64a9]
0x1a0b7:      call    0x47e0
0x1a0bc:      mov     r12, rax
0x1a0bf:      test    rax, rax
0x1a0c2:      je      0x1a110
0x1a0c4:      mov     rdi, rax
0x1a0c7:      call    0x49b0

```

```

0x1a0cc:    lea     edi, [rax + 0xa]
0x1a0cf:    movsxd  rbp, eax
0x1a0d2:    movsxd  rdi, edi
0x1a0d5:    lea     rbx, [rbp + 1]
0x1a0d9:    call    0x1d670
0x1a0de:    mov     rsi, r12
0x1a0e1:    mov     rdi, rax
0x1a0e4:    call    0x48b0
0x1a0e9:    movabs  rdx, 0x79726f747369682e
0x1a0f3:    mov     byte ptr [rax + rbp], 0x2f
0x1a0f7:    mov     r8, rax
0x1a0fa:    lea     rax, [rax + rbx]
0x1a0fe:    mov     qword ptr [rax], rdx
0x1a101:    mov     byte ptr [rax + 8], 0
0x1a105:    mov     rax, r8
0x1a108:    pop     rbx
0x1a109:    pop     rbp
0x1a10a:    pop     r12
0x1a10c:    ret
0x1a10d:    nop     dword ptr [rax]
0x1a110:    mov     ebx, 2
0x1a115:    mov     ebp, 1
0x1a11a:    mov     edi, 0xb
0x1a11f:    lea     r12, [rip + 0x75b8]
0x1a126:    jmp     0x1a0d9
0x1a0d9:
0x1a128:    nop     dword ptr [rax + rax]
0x1a130:    push    r15
0x1a132:    xor     eax, eax
0x1a134:    push    r14
0x1a136:    mov     r14d, esi
0x1a139:    push    r13
0x1a13b:    push    r12
0x1a13d:    push    rbp
0x1a13e:    push    rbx
0x1a13f:    mov     ebx, edx
0x1a141:    sub     rsp, 0x18
0x1a145:    call    0x1a080
0x1a14a:    cmp     ebx, 1
0x1a14d:    mov     edx, 0x1b6
0x1a152:    sbb     esi, esi
0x1a154:    mov     r13, rax
0x1a157:    mov     rdi, rax
0x1a15a:    xor     eax, eax
0x1a15c:    and     esi, 0x1c0
0x1a162:    add     esi, 0x241
0x1a168:    call    0x4d80
0x1a16d:    cmp     eax, -1

```

```

0x1a170:    je      0x1a270
0x1a176:    mov     ebp, eax
0x1a178:    mov     eax, dword ptr [rip + 0x11966]
0x1a17e:    cmp     eax, r14d
0x1a181:    mov     edx, eax
0x1a183:    cmovle  r14d, eax
0x1a187:    sub     edx, r14d
0x1a18a:    cmp     eax, edx
0x1a18c:    jle     0x1a290
0x1a192:    sub     eax, 1
0x1a195:    mov     rcx, qword ptr [rip + 0x11964]
0x1a19c:    movsxd  rsi, edx
0x1a19f:    xor     r12d, r12d
0x1a1a2:    sub     eax, edx
0x1a1a4:    add     rax, rsi
0x1a1a7:    lea     rbx, [rcx + rsi*8]
0x1a1ab:    lea     r15, [rcx + rax*8 + 8]
0x1a1b0:    mov     rax, qword ptr [rbx]
0x1a1b3:    add     rbx, 8
0x1a1b7:    mov     rdi, qword ptr [rax]
0x1a1ba:    call    0x49b0
0x1a1bf:    lea     r12d, [r12 + rax + 1]
0x1a1c4:    cmp     r15, rbx
0x1a1c7:    jne     0x1a1b0
0x1a1c9:    movsxd  rax, r12d
0x1a1cc:    mov     qword ptr [rsp + 8], rax
0x1a1d1:    mov     rdi, qword ptr [rsp + 8]
0x1a1d6:    call    0x1d670
0x1a1db:    mov     r15, rax
0x1a1de:    mov     eax, dword ptr [rip + 0x11900]
0x1a1e4:    mov     ebx, eax
0x1a1e6:    sub     ebx, r14d
0x1a1e9:    cmp     eax, ebx
0x1a1eb:    jle     0x1a238
0x1a1ed:    mov     r12, qword ptr [rip + 0x1190c]
0x1a1f4:    movsxd  rbx, ebx
0x1a1f7:    xor     r14d, r14d
0x1a1fa:    nop     word ptr [rax + rax]
0x1a200:    mov     rax, qword ptr [r12 + rbx*8]
0x1a204:    movsxd  rdi, r14d
0x1a207:    add     rdi, r15
0x1a20a:    mov     rsi, qword ptr [rax]
0x1a20d:    call    0x48b0
0x1a212:    mov     rax, qword ptr [r12 + rbx*8]
0x1a216:    add     rbx, 1
0x1a21a:    mov     rdi, qword ptr [rax]
0x1a21d:    call    0x49b0
0x1a222:    add     eax, r14d

```

```

0x1a225:    lea     r14d, [rax + 1]
0x1a229:    cdqe
0x1a22b:    mov     byte ptr [r15 + rax], 0xa
0x1a230:    cmp     dword ptr [rip + 0x118ae], ebx
0x1a236:    jg      0x1a200
0x1a238:    mov     rdx, qword ptr [rsp + 8]
0x1a23d:    mov     rsi, r15
0x1a240:    mov     edi, ebp
0x1a242:    call    0x4790
0x1a247:    mov     rdi, r15
0x1a24a:    call    0x18d10
0x1a24f:    mov     edi, ebp
0x1a251:    call    0x4ab0
0x1a256:    mov     rdi, r13
0x1a259:    call    0x18d10
0x1a25e:    add     rsp, 0x18
0x1a262:    xor     eax, eax
0x1a264:    pop     rbx
0x1a265:    pop     rbp
0x1a266:    pop     r12
0x1a268:    pop     r13
0x1a26a:    pop     r14
0x1a26c:    pop     r15
0x1a26e:    ret
0x1a26f:    nop
0x1a270:    mov     rdi, r13
0x1a273:    call    0x18d10
0x1a278:    call    0x4860
0x1a27d:    mov     eax, dword ptr [rax]
0x1a27f:    add     rsp, 0x18
0x1a283:    pop     rbx
0x1a284:    pop     rbp
0x1a285:    pop     r12
0x1a287:    pop     r13
0x1a289:    pop     r14
0x1a28b:    pop     r15
0x1a28d:    ret
0x1a28e:    nop
0x1a290:    mov     qword ptr [rsp + 8], 0
0x1a299:    jmp     0x1a1d1
0x1a1d1:
0x1a29e:    nop
0x1a2a0:    endbr64
0x1a2a4:    sub     rsp, 8
0x1a2a8:    mov     edi, 0x18
0x1a2ad:    call    0x1d670
0x1a2b2:    mov     rdx, qword ptr [rip + 0x11847]
0x1a2b9:    mov     qword ptr [rax], rdx

```



```

0x1a2bc:    mov     edx, dword ptr [rip + 0x1183a]
0x1a2c2:    mov     dword ptr [rax + 8], edx
0x1a2c5:    mov     edx, dword ptr [rip + 0x11819]
0x1a2cb:    mov     dword ptr [rax + 0xc], edx
0x1a2ce:    mov     edx, dword ptr [rip + 0x11824]
0x1a2d4:    mov     dword ptr [rax + 0x10], edx
0x1a2d7:    add     rsp, 8
0x1a2db:    ret
0x1a2dc:    nop     dword ptr [rax]
0x1a2e0:    endbr64
0x1a2e4:    mov     rax, qword ptr [rdi]
0x1a2e7:    mov     qword ptr [rip + 0x11812], rax
0x1a2ee:    mov     eax, dword ptr [rdi + 8]
0x1a2f1:    mov     dword ptr [rip + 0x11805], eax
0x1a2f7:    mov     eax, dword ptr [rdi + 0xc]
0x1a2fa:    mov     dword ptr [rip + 0x117e4], eax
0x1a300:    mov     eax, dword ptr [rdi + 0x10]
0x1a303:    mov     dword ptr [rip + 0x117ef], eax
0x1a309:    ret
0x1a30a:    nop     word ptr [rax + rax]
0x1a310:    endbr64
0x1a314:    mov     eax, dword ptr [rip + 0x117ca]
0x1a31a:    mov     dword ptr [rip + 0x117dc], eax
0x1a320:    ret
0x1a321:    nop     word ptr cs:[rax + rax]
0x1a32c:    nop     dword ptr [rax]
0x1a330:    endbr64
0x1a334:    push    r12
0x1a336:    xor     r12d, r12d
0x1a339:    push    rbx
0x1a33a:    sub     rsp, 8
0x1a33e:    mov     rbx, qword ptr [rip + 0x117bb]
0x1a345:    test    rbx, rbx
0x1a348:    jne     0x1a35b
0x1a34a:    jmp     0x1a367
0x1a367:
0x1a34c:    nop     dword ptr [rax]
0x1a350:    mov     rdi, qword ptr [rax]
0x1a353:    call    0x49b0
0x1a358:    add     r12d, eax
0x1a35b:    mov     rax, qword ptr [rbx]
0x1a35e:    add     rbx, 8
0x1a362:    test    rax, rax
0x1a365:    jne     0x1a350
0x1a367:    add     rsp, 8
0x1a36b:    mov     eax, r12d
0x1a36e:    pop     rbx
0x1a36f:    pop     r12

```

```

0x1a371:    ret
0x1a372:    nop      word ptr cs:[rax + rax]
0x1a37d:    nop      dword ptr [rax]
0x1a380:    endbr64
0x1a384:    push     rbp
0x1a385:    mov      rbp, rdi
0x1a388:    push     rbx
0x1a389:    sub      rsp, 8
0x1a38d:    mov      eax, dword ptr [rip + 0x11759]
0x1a393:    test     eax, eax
0x1a395:    je       0x1a3aa
0x1a397:    movsxd   rdx, dword ptr [rip + 0x11746]
0x1a39e:    cmp      edx, dword ptr [rip + 0x11744]
0x1a3a4:    je       0x1a458
0x1a3aa:    mov      eax, dword ptr [rip + 0x11748]
0x1a3b0:    test     eax, eax
0x1a3b2:    je       0x1a428
0x1a3b4:    mov      edx, dword ptr [rip + 0x1172a]
0x1a3ba:    lea      ecx, [rax - 1]
0x1a3bd:    cmp      ecx, edx
0x1a3bf:    je       0x1a4b0
0x1a3c5:    add      edx, 1
0x1a3c8:    mov      dword ptr [rip + 0x11716], edx
0x1a3ce:    mov      edi, 0x10
0x1a3d3:    call     0x1d670
0x1a3d8:    mov      rdi, rbp
0x1a3db:    mov      rbx, rax
0x1a3de:    call     0x49b0
0x1a3e3:    lea      rdi, [rax + 1]
0x1a3e7:    call     0x1d670
0x1a3ec:    mov      rsi, rbp
0x1a3ef:    mov      rdi, rax
0x1a3f2:    call     0x48b0
0x1a3f7:    movsxd   rdx, dword ptr [rip + 0x116e6]
0x1a3fe:    mov      qword ptr [rbx + 8], 0
0x1a406:    mov      qword ptr [rbx], rax
0x1a409:    mov      rax, qword ptr [rip + 0x116f0]
0x1a410:    mov      qword ptr [rax + rdx*8], 0
0x1a418:    mov      qword ptr [rax + rdx*8 - 8], rbx
0x1a41d:    add      rsp, 8
0x1a421:    pop      rbx
0x1a422:    pop      rbp
0x1a423:    ret
0x1a424:    nop      dword ptr [rax]
0x1a428:    mov      dword ptr [rip + 0x116c6], 0x32
0x1a432:    mov      edi, 0x190
0x1a437:    call     0x1d670
0x1a43c:    mov      dword ptr [rip + 0x1169e], 1

```

```

0x1a446:    mov     qword ptr [rip + 0x116b3], rax
0x1a44d:    jmp     0x1a3ce
0x1a3ce:
0x1a452:    nop     word ptr [rax + rax]
0x1a458:    test    edx, edx
0x1a45a:    je      0x1a41d
0x1a45c:    mov     rax, qword ptr [rip + 0x1169d]
0x1a463:    mov     rax, qword ptr [rax]
0x1a466:    test    rax, rax
0x1a469:    je      0x1a489
0x1a46b:    mov     rdi, qword ptr [rax]
0x1a46e:    call    0x18d10
0x1a473:    mov     rax, qword ptr [rip + 0x11686]
0x1a47a:    mov     rdi, qword ptr [rax]
0x1a47d:    call    0x18d10
0x1a482:    movsxd  rdx, dword ptr [rip + 0x1165b]
0x1a489:    test    edx, edx
0x1a48b:    jle     0x1a4a1
0x1a48d:    mov     rdi, qword ptr [rip + 0x1166c]
0x1a494:    shl     rdx, 3
0x1a498:    lea     rsi, [rdi + 8]
0x1a49c:    call    0x4d40
0x1a4a1:    add     dword ptr [rip + 0x10340], 1
0x1a4a8:    jmp     0x1a3ce
0x1a3ce:
0x1a4ad:    nop     dword ptr [rax]
0x1a4b0:    add     eax, 0x32
0x1a4b3:    mov     rdi, qword ptr [rip + 0x11646]
0x1a4ba:    mov     dword ptr [rip + 0x11638], eax
0x1a4c0:    cdqe
0x1a4c2:    lea     rsi, [rax*8]
0x1a4ca:    call    0x1d690
0x1a4cf:    mov     edx, dword ptr [rip + 0x1160f]
0x1a4d5:    mov     qword ptr [rip + 0x11624], rax
0x1a4dc:    jmp     0x1a3c5
0x1a3c5:
0x1a4e1:    nop     word ptr cs:[rax + rax]
0x1a4ec:    nop     dword ptr [rax]
0x1a4f0:    endbr64
0x1a4f4:    push    r14
0x1a4f6:    xor     r14d, r14d
0x1a4f9:    push    r13
0x1a4fb:    mov     r13, rdx
0x1a4fe:    push    r12
0x1a500:    mov     r12, rsi
0x1a503:    push    rbp
0x1a504:    push    rbx
0x1a505:    movsxd  rbx, edi

```

```

0x1a508:    mov     edi, 0x10
0x1a50d:    call    0x1d670
0x1a512:    cmp     dword ptr [rip + 0x115cc], ebx
0x1a518:    jle     0x1a557
0x1a51a:    mov     rbp, rax
0x1a51d:    mov     rax, qword ptr [rip + 0x115dc]
0x1a524:    mov     rdi, r12
0x1a527:    mov     r14, qword ptr [rax + rbx*8]
0x1a52b:    call    0x49b0
0x1a530:    lea     rdi, [rax + 1]
0x1a534:    call    0x1d670
0x1a539:    mov     rsi, r12
0x1a53c:    mov     rdi, rax
0x1a53f:    call    0x48b0
0x1a544:    mov     qword ptr [rbp + 8], r13
0x1a548:    mov     qword ptr [rbp], rax
0x1a54c:    mov     rax, qword ptr [rip + 0x115ad]
0x1a553:    mov     qword ptr [rax + rbx*8], rbp
0x1a557:    pop     rbx
0x1a558:    mov     rax, r14
0x1a55b:    pop     rbp
0x1a55c:    pop     r12
0x1a55e:    pop     r13
0x1a560:    pop     r14
0x1a562:    ret
0x1a563:    nop     word ptr cs:[rax + rax]
0x1a56e:    nop
0x1a570:    endbr64
0x1a574:    mov     eax, dword ptr [rip + 0x11582]
0x1a57a:    ret
0x1a57b:    nop     dword ptr [rax + rax]
0x1a580:    endbr64
0x1a584:    xor     edx, edx
0x1a586:    xor     eax, eax
0x1a588:    jmp     0x19de0
0x19de0:
0x1a58d:    nop     dword ptr [rax]
0x1a590:    endbr64
0x1a594:    mov     edx, 1
0x1a599:    xor     eax, eax
0x1a59b:    jmp     0x19de0
0x19de0:
0x1a5a0:    endbr64
0x1a5a4:    push    r12
0x1a5a6:    push    rbx
0x1a5a7:    sub     rsp, 8
0x1a5ab:    mov     eax, dword ptr [rip + 0x11533]
0x1a5b1:    cmp     eax, edi

```

```

0x1a5b3:    jle     0x1a610
0x1a5b5:    test    eax, eax
0x1a5b7:    je      0x1a610
0x1a5b9:    lea     ebx, [rax - 1]
0x1a5bc:    movsxd  rdx, edi
0x1a5bf:    mov     rcx, qword ptr [rip + 0x1153a]
0x1a5c6:    mov     esi, ebx
0x1a5c8:    shl     rdx, 3
0x1a5cc:    sub     esi, edi
0x1a5ce:    cmp     eax, edi
0x1a5d0:    lea     r9, [rcx + rdx]
0x1a5d4:    mov     eax, 8
0x1a5d9:    lea     r8, [rsi*8 + 8]
0x1a5e1:    mov     r12, qword ptr [r9]
0x1a5e4:    mov     rdi, r9
0x1a5e7:    lea     rsi, [rcx + rdx + 8]
0x1a5ec:    cmovle  r8, rax
0x1a5f0:    mov     rdx, r8
0x1a5f3:    call    0x4d40
0x1a5f8:    mov     dword ptr [rip + 0x114e6], ebx
0x1a5fe:    add     rsp, 8
0x1a602:    mov     rax, r12
0x1a605:    pop     rbx
0x1a606:    pop     r12
0x1a608:    ret
0x1a609:    nop     dword ptr [rax]
0x1a610:    xor     r12d, r12d
0x1a613:    add     rsp, 8
0x1a617:    mov     rax, r12
0x1a61a:    pop     rbx
0x1a61b:    pop     r12
0x1a61d:    ret
0x1a61e:    nop
0x1a620:    endbr64
0x1a624:    push    r12
0x1a626:    mov     eax, dword ptr [rip + 0x114b8]
0x1a62c:    push    rbp
0x1a62d:    xor     ebp, ebp
0x1a62f:    test    edi, edi
0x1a631:    cmovns  ebp, edi
0x1a634:    push    rbx
0x1a635:    cmp     eax, ebp
0x1a637:    jle     0x1a6ce
0x1a63d:    sub     eax, ebp
0x1a63f:    mov     r12d, edi
0x1a642:    test    eax, eax
0x1a644:    jle     0x1a6e8
0x1a64a:    xor     ebx, ebx

```

```

0x1a64c:      nop      dword ptr [rax]
0x1a650:      mov      rax, qword ptr [rip + 0x114a9]
0x1a657:      mov      rax, qword ptr [rax + rbx*8]
0x1a65b:      mov      rdi, qword ptr [rax]
0x1a65e:      call     0x18d10
0x1a663:      mov      rax, qword ptr [rip + 0x11496]
0x1a66a:      mov      rdi, qword ptr [rax + rbx*8]
0x1a66e:      call     0x18d10
0x1a673:      mov      eax, dword ptr [rip + 0x1146b]
0x1a679:      lea      edx, [rbx + 1]
0x1a67c:      add      rbx, 1
0x1a680:      sub      eax, ebp
0x1a682:      cmp      eax, ebx
0x1a684:      jg       0x1a650
0x1a686:      mov      dword ptr [rip + 0x1015c], edx
0x1a68c:      mov      rcx, qword ptr [rip + 0x1146d]
0x1a693:      test     r12d, r12d
0x1a696:      jle      0x1a6f0
0x1a698:      cdqe
0x1a69a:      lea      rsi, [rcx + rax*8]
0x1a69e:      xor      eax, eax
0x1a6a0:      mov      rdx, qword ptr [rsi + rax*8]
0x1a6a4:      mov      qword ptr [rcx + rax*8], rdx
0x1a6a8:      add      rax, 1
0x1a6ac:      cmp      ebp, eax
0x1a6ae:      jg       0x1a6a0
0x1a6b0:      test     ebp, ebp
0x1a6b2:      mov      eax, 1
0x1a6b7:      cmovg    eax, ebp
0x1a6ba:      movsxd   rdx, eax
0x1a6bd:      lea      rcx, [rcx + rdx*8]
0x1a6c1:      mov      qword ptr [rcx], 0
0x1a6c8:      mov      dword ptr [rip + 0x11416], eax
0x1a6ce:      pop      rbx
0x1a6cf:      mov      dword ptr [rip + 0x11413], ebp
0x1a6d5:      mov      dword ptr [rip + 0x1140d], 1
0x1a6df:      pop      rbp
0x1a6e0:      pop      r12
0x1a6e2:      ret
0x1a6e3:      nop      dword ptr [rax + rax]
0x1a6e8:      xor      edx, edx
0x1a6ea:      jmp      0x1a686
0x1a686:
0x1a6ec:      nop      dword ptr [rax]
0x1a6f0:      xor      eax, eax
0x1a6f2:      jmp      0x1a6c1
0x1a6c1:
0x1a6f4:      nop      word ptr cs:[rax + rax]

```

```

0x1a6ff:      nop
0x1a700:      endbr64
0x1a704:      mov     edx, dword ptr [rip + 0x113e2]
0x1a70a:      mov     eax, dword ptr [rip + 0x113d8]
0x1a710:      test    edx, edx
0x1a712:      je      0x1a720
0x1a714:      mov     dword ptr [rip + 0x113ce], 0
0x1a71e:      neg     eax
0x1a720:      ret
0x1a721:      nop     word ptr cs:[rax + rax]
0x1a72c:      nop     dword ptr [rax]
0x1a730:      endbr64
0x1a734:      push    r15
0x1a736:      push    r14
0x1a738:      mov     r14d, esi
0x1a73b:      push    r13
0x1a73d:      push    r12
0x1a73f:      mov     r12d, edx
0x1a742:      push    rbp
0x1a743:      push    rbx
0x1a744:      sub     rsp, 0xa8
0x1a74b:      mov     rax, qword ptr fs:[0x28]
0x1a754:      mov     qword ptr [rsp + 0x98], rax
0x1a75c:      xor     eax, eax
0x1a75e:      call    0x1a080
0x1a763:      xor     esi, esi
0x1a765:      mov     edx, 0x1b6
0x1a76a:      mov     rbp, rax
0x1a76d:      mov     rdi, rax
0x1a770:      xor     eax, eax
0x1a772:      call    0x4d80
0x1a777:      test    eax, eax
0x1a779:      js      0x1a80f
0x1a77f:      mov     rsi, rsp
0x1a782:      mov     edi, eax
0x1a784:      mov     r15d, eax
0x1a787:      call    0x18ef0
0x1a78c:      cmp     eax, -1
0x1a78f:      je      0x1a807
0x1a791:      mov     eax, dword ptr [rsp + 0x30]
0x1a795:      lea     edi, [rax + 1]
0x1a798:      movsxd  rdi, edi
0x1a79b:      call    0x1d670
0x1a7a0:      mov     rdx, qword ptr [rsp + 0x30]
0x1a7a5:      mov     edi, r15d
0x1a7a8:      mov     rsi, rax
0x1a7ab:      mov     r13, rax
0x1a7ae:      call    0x47a0

```

```

0x1a7b3:      cmp     qword ptr [rsp + 0x30], rax
0x1a7b8:      je      0x1a819
0x1a7ba:      mov     edi, r15d
0x1a7bd:      call    0x4ab0
0x1a7c2:      mov     rdi, rbp
0x1a7c5:      call    0x18d10
0x1a7ca:      test    r13, r13
0x1a7cd:      je      0x1a7d7
0x1a7cf:      mov     rdi, r13
0x1a7d2:      call    0x18d10
0x1a7d7:      call    0x4860
0x1a7dc:      mov     eax, dword ptr [rax]
0x1a7de:      mov     rdx, qword ptr [rsp + 0x98]
0x1a7e6:      sub     rdx, qword ptr fs:[0x28]
0x1a7ef:      jne     0x1a8f0
0x1a7f5:      add     rsp, 0xa8
0x1a7fc:      pop     rbx
0x1a7fd:      pop     rbp
0x1a7fe:      pop     r12
0x1a800:      pop     r13
0x1a802:      pop     r14
0x1a804:      pop     r15
0x1a806:      ret
0x1a807:      mov     edi, r15d
0x1a80a:      call    0x4ab0
0x1a80f:      mov     rdi, rbp
0x1a812:      call    0x18d10
0x1a817:      jmp     0x1a7d7
0x1a7d7:
0x1a819:      mov     edi, r15d
0x1a81c:      call    0x4ab0
0x1a821:      mov     rcx, qword ptr [rsp + 0x30]
0x1a826:      test    r12d, r12d
0x1a829:      jns     0x1a82e
0x1a82b:      mov     r12d, ecx
0x1a82e:      xor     ebx, ebx
0x1a830:      xor     esi, esi
0x1a832:      movsxd  rdx, esi
0x1a835:      cmp     ebx, r14d
0x1a838:      jge     0x1a885
0x1a83a:      cmp     rdx, rcx
0x1a83d:      jge     0x1a885
0x1a83f:      lea     eax, [rsi + 1]
0x1a842:      cdq     q
0x1a844:      cmp     rdx, rcx
0x1a847:      jl      0x1a85c
0x1a849:      jmp     0x1a87b
0x1a87b:

```



```

0x1a84b:      nop      dword ptr [rax + rax]
0x1a850:      lea      rdx, [rax + 1]
0x1a854:      cmp      rax, rcx
0x1a857:      je       0x1a878
0x1a859:      mov      rax, rdx
0x1a85c:      cmp      byte ptr [r13 + rax - 1], 0xa
0x1a862:      jne      0x1a850
0x1a864:      add      ebx, 1
0x1a867:      movsxd   rsi, eax
0x1a86a:      cmp      r14d, ebx
0x1a86d:      je       0x1a832
0x1a86f:      lea      rdx, [rax + 1]
0x1a873:      cmp      rax, rcx
0x1a876:      jne      0x1a859
0x1a878:      movsxd   rdx, esi
0x1a87b:      cmp      rcx, rdx
0x1a87e:      jle      0x1a885
0x1a880:      cmp      r14d, ebx
0x1a883:      jg       0x1a83f
0x1a885:      lea      r14, [r13 + rdx]
0x1a88a:      lea      r15d, [rsi + 1]
0x1a88e:      cmp      rcx, rdx
0x1a891:      jg       0x1a8aa
0x1a893:      jmp      0x1a8d4
0x1a8d4
0x1a895:      nop      dword ptr [rax]
0x1a898:      movsxd   rax, r15d
0x1a89b:      add      r14, 1
0x1a89f:      add      r15d, 1
0x1a8a3:      cmp      rax, qword ptr [rsp + 0x30]
0x1a8a8:      jge      0x1a8d4
0x1a8aa:      cmp      byte ptr [r14], 0xa
0x1a8ae:      jne      0x1a898
0x1a8b0:      lea      rdi, [r13 + rsi]
0x1a8b5:      mov      byte ptr [r14], 0
0x1a8b9:      cmp      byte ptr [rdi], 0
0x1a8bc:      jne      0x1a8cb
0x1a8be:      add      ebx, 1
0x1a8c1:      cmp      r12d, ebx
0x1a8c4:      jle      0x1a8d4
0x1a8c6:      movsxd   rsi, r15d
0x1a8c9:      jmp      0x1a898
0x1a898
0x1a8cb:      xor      eax, eax
0x1a8cd:      call     0x1a380
0x1a8d2:      jmp      0x1a8be
0x1a8be
0x1a8d4:      mov      rdi, rbp

```

```

0x1a8d7:      call    0x18d10
0x1a8dc:      test    r13, r13
0x1a8df:      je      0x1a8e9
0x1a8e1:      mov     rdi, r13
0x1a8e4:      call    0x18d10
0x1a8e9:      xor     eax, eax
0x1a8eb:      jmp     0x1a7de
0x1a7de
0x1a8f0:      call    0x49d0
0x1a8f5:      nop     word ptr cs:[rax + rax]
0x1a900:      endbr64
0x1a904:      mov     edx, 0xffffffff
0x1a909:      xor     esi, esi
0x1a90b:      xor     eax, eax
0x1a90d:      jmp     0x1a730
0x1a730
0x1a912:      nop     word ptr cs:[rax + rax]
0x1a91d:      nop     dword ptr [rax]
0x1a920:      endbr64
0x1a924:      push    r14
0x1a926:      push    r13
0x1a928:      push    r12
0x1a92a:      mov     r12d, esi
0x1a92d:      push    rbp
0x1a92e:      push    rbx
0x1a92f:      sub     rsp, 0xa0
0x1a936:      mov     rax, qword ptr fs:[0x28]
0x1a93f:      mov     qword ptr [rsp + 0x98], rax
0x1a947:      xor     eax, eax
0x1a949:      call    0x1a080
0x1a94e:      xor     esi, esi
0x1a950:      mov     edx, 0x1b6
0x1a955:      mov     rbp, rax
0x1a958:      mov     rdi, rax
0x1a95b:      xor     eax, eax
0x1a95d:      call    0x4d80
0x1a962:      cmp     eax, -1
0x1a965:      je      0x1aa58
0x1a96b:      mov     rsi, rsp
0x1a96e:      mov     edi, eax
0x1a970:      mov     r14d, eax
0x1a973:      call    0x18ef0
0x1a978:      cmp     eax, -1
0x1a97b:      je      0x1aa58
0x1a981:      mov     eax, dword ptr [rsp + 0x30]
0x1a985:      lea     edi, [rax + 1]
0x1a988:      movsxd  rdi, edi
0x1a98b:      call    0x1d670

```

```

0x1a990:    mov     rdx, qword ptr [rsp + 0x30]
0x1a995:    mov     edi, r14d
0x1a998:    mov     rsi, rax
0x1a99b:    mov     r13, rax
0x1a99e:    call    0x47a0
0x1a9a3:    mov     edi, r14d
0x1a9a6:    mov     rbx, rax
0x1a9a9:    call    0x4ab0
0x1a9ae:    test    ebx, ebx
0x1a9b0:    jle     0x1aa88
0x1a9b6:    lea     eax, [rbx - 1]
0x1a9b9:    cmp     ebx, 1
0x1a9bc:    je      0x1a9f3
0x1a9be:    test    r12d, r12d
0x1a9c1:    je      0x1a9f3
0x1a9c3:    lea     edx, [rbx - 2]
0x1a9c6:    movsxd  rdx, edx
0x1a9c9:    nop     dword ptr [rax]
0x1a9d0:    cmp     byte ptr [r13 + rdx + 1], 0xa
0x1a9d6:    jne     0x1a9dc
0x1a9d8:    sub     r12d, 1
0x1a9dc:    test    r12d, r12d
0x1a9df:    mov     eax, edx
0x1a9e1:    setne   sil
0x1a9e5:    test    edx, edx
0x1a9e7:    setne   cl
0x1a9ea:    sub     rdx, 1
0x1a9ee:    test    sil, cl
0x1a9f1:    jne     0x1a9d0
0x1a9f3:    test    eax, eax
0x1a9f5:    je      0x1aa88
0x1a9fb:    cdqe
0x1a9fd:    jmp     0x1aa08
0x1aa08:
0x1a9ff:    nop
0x1aa00:    sub     rax, 1
0x1aa04:    test    eax, eax
0x1aa06:    je      0x1aa50
0x1aa08:    cmp     byte ptr [r13 + rax], 0xa
0x1aa0e:    mov     ebx, eax
0x1aa10:    jne     0x1aa00
0x1aa12:    mov     edx, 0x1b6
0x1aa17:    mov     esi, 0x201
0x1aa1c:    mov     rdi, rbp
0x1aa1f:    xor     eax, eax
0x1aa21:    call    0x4d80
0x1aa26:    mov     r12d, eax
0x1aa29:    cmp     eax, -1

```

```

0x1aa2c:      je      0x1aa50
0x1aa2e:      mov     rdx, qword ptr [rsp + 0x30]
0x1aa33:      lea     esi, [rbx + 1]
0x1aa36:      mov     edi, eax
0x1aa38:      movsxd   rsi, esi
0x1aa3b:      sub     rdx, rsi
0x1aa3e:      add     rsi, r13
0x1aa41:      call    0x4790
0x1aa46:      mov     edi, r12d
0x1aa49:      call    0x4ab0
0x1aa4e:      nop
0x1aa50:      mov     rdi, r13
0x1aa53:      call    0x18d10
0x1aa58:      mov     rdi, rbp
0x1aa5b:      call    0x18d10
0x1aa60:      mov     rax, qword ptr [rsp + 0x98]
0x1aa68:      sub     rax, qword ptr fs:[0x28]
0x1aa71:      jne     0x1aa8f
0x1aa73:      add     rsp, 0xa0
0x1aa7a:      xor     eax, eax
0x1aa7c:      pop     rbx
0x1aa7d:      pop     rbp
0x1aa7e:      pop     r12
0x1aa80:      pop     r13
0x1aa82:      pop     r14
0x1aa84:      ret
0x1aa85:      nop     dword ptr [rax]
0x1aa88:      test    r13, r13
0x1aa8b:      je      0x1aa58
0x1aa8d:      jmp     0x1aa50
0x1aa50
0x1aa8f:      call    0x49d0
0x1aa94:      nop     word ptr cs:[rax + rax]
0x1aa9f:      nop
0x1aaa0:      endbr64
0x1aaa4:      mov     r8d, edi
0x1aaa7:      xor     edx, edx
0x1aaa9:      mov     rdi, rsi
0x1aaac:      xor     eax, eax
0x1aaae:      mov     esi, r8d
0x1aab1:      jmp     0x1a130
0x1a130
0x1aab6:      nop     word ptr cs:[rax + rax]
0x1aac0:      endbr64
0x1aac4:      mov     esi, dword ptr [rip + 0x1101a]
0x1aaca:      mov     edx, 1
0x1aacf:      xor     eax, eax
0x1aad1:      jmp     0x1a130

```

```

0x1a130
0x1aad6:      nop      word ptr cs:[rax + rax]
0x1aae0:      endbr64
0x1aae4:      movsxd   rdx, dword ptr [rip + 0x11011]
0x1aaeb:      cmp      edx, dword ptr [rip + 0x10ff3]
0x1aaf1:      je       0x1ab08
0x1aaf3:      mov      rax, qword ptr [rip + 0x11006]
0x1aafa:      test     rax, rax
0x1aafd:      je       0x1ab0a
0x1aaff:      mov      rax, qword ptr [rax + rdx*8]
0x1ab03:      ret
0x1ab04:      nop      dword ptr [rax]
0x1ab08:      xor      eax, eax
0x1ab0a:      ret
0x1ab0b:      nop      dword ptr [rax + rax]
0x1ab10:      endbr64
0x1ab14:      mov      eax, dword ptr [rip + 0x10fe2]
0x1ab1a:      xor      r8d, r8d
0x1ab1d:      test     eax, eax
0x1ab1f:      je       0x1ab37
0x1ab21:      sub      eax, 1
0x1ab24:      mov      rdx, qword ptr [rip + 0x10fd5]
0x1ab2b:      mov      dword ptr [rip + 0x10fcb], eax
0x1ab31:      cdqe
0x1ab33:      mov      r8, qword ptr [rdx + rax*8]
0x1ab37:      mov      rax, r8
0x1ab3a:      ret
0x1ab3b:      nop      dword ptr [rax + rax]
0x1ab40:      endbr64
0x1ab44:      mov      eax, dword ptr [rip + 0x10fb2]
0x1ab4a:      cmp      eax, dword ptr [rip + 0x10f94]
0x1ab50:      je       0x1ab70
0x1ab52:      add      eax, 1
0x1ab55:      mov      rdx, qword ptr [rip + 0x10fa4]
0x1ab5c:      mov      dword ptr [rip + 0x10f9a], eax
0x1ab62:      cdqe
0x1ab64:      mov      rax, qword ptr [rdx + rax*8]
0x1ab68:      ret
0x1ab69:      nop      dword ptr [rax]
0x1ab70:      xor      eax, eax
0x1ab72:      ret
0x1ab73:      nop      word ptr cs:[rax + rax]
0x1ab7e:      nop
0x1ab80:      endbr64
0x1ab84:      mov      rax, qword ptr [rip + 0x10f75]
0x1ab8b:      ret
0x1ab8c:      nop      dword ptr [rax]
0x1ab90:      endbr64

```

```

0x1ab94:    sub     edi, dword ptr [rip + 0xfc4e]
0x1ab9a:    cmp     dword ptr [rip + 0x10f44], edi
0x1aba0:    jle     0x1abc0
0x1aba2:    test    edi, edi
0x1aba4:    js      0x1abc0
0x1aba6:    mov     rax, qword ptr [rip + 0x10f53]
0x1abad:    test    rax, rax
0x1abb0:    je      0x1abc2
0x1abb2:    movsxd  rdi, edi
0x1abb5:    mov     rax, qword ptr [rax + rdi*8]
0x1abb9:    ret
0x1abba:    nop     word ptr [rax + rax]
0x1abc0:    xor     eax, eax
0x1abc2:    ret
0x1abc3:    nop     word ptr cs:[rax + rax]
0x1abce:    nop
0x1abd0:    endbr64
0x1abd4:    cmp     edx, dword ptr [rip + 0x10f0a]
0x1abda:    push    rbx
0x1abdb:    mov     ebx, dword ptr [rip + 0x10f1b]
0x1abe1:    jg      0x1abf7
0x1abe3:    test    edx, edx
0x1abe5:    js      0x1abf7
0x1abe7:    cmp     qword ptr [rip + 0x10f11], 0
0x1abef:    je      0x1abf7
0x1abf1:    mov     dword ptr [rip + 0x10f05], edx
0x1abf7:    xor     edx, edx
0x1abf9:    xor     eax, eax
0x1abfb:    call    0x19de0
0x1ac00:    mov     edx, ebx
0x1ac02:    shr     edx, 0x1f
0x1ac05:    cmp     eax, -1
0x1ac08:    je      0x1ac10
0x1ac0a:    mov     eax, dword ptr [rip + 0x10eec]
0x1ac10:    cmp     ebx, dword ptr [rip + 0x10ece]
0x1ac16:    jg      0x1ac2c
0x1ac18:    test    dl, dl
0x1ac1a:    jne     0x1ac2c
0x1ac1c:    cmp     qword ptr [rip + 0x10edc], 0
0x1ac24:    je      0x1ac2c
0x1ac26:    mov     dword ptr [rip + 0x10ed0], ebx
0x1ac2c:    pop     rbx
0x1ac2d:    ret
0x1ac2e:    nop
0x1ac30:    endbr64
0x1ac34:    cmp     dword ptr [rip + 0x10eaa], edi
0x1ac3a:    jl      0x1ac60
0x1ac3c:    test    edi, edi

```

```

0x1ac3e:    js      0x1ac60
0x1ac40:    cmp     qword ptr [rip + 0x10eb8], 0
0x1ac48:    je      0x1ac60
0x1ac4a:    mov     dword ptr [rip + 0x10eac], edi
0x1ac50:    mov     eax, 1
0x1ac55:    ret
0x1ac56:    nop     word ptr cs:[rax + rax]
0x1ac60:    xor     eax, eax
0x1ac62:    ret
0x1ac63:    nop     word ptr cs:[rax + rax]
0x1ac6e:    nop
0x1ac70:    endbr64
0x1ac74:    movsxd  rax, dword ptr [rsi]
0x1ac77:    movzx   r8d, byte ptr [rip + 0xfb6e]
0x1ac7f:    cmp     byte ptr [rdi + rax], r8b
0x1ac83:    jne     0x1afd0
0x1ac89:    push    r15
0x1ac8b:    lea     ecx, [rax + 1]
0x1ac8e:    mov     r9d, edx
0x1ac91:    mov     rdx, rax
0x1ac94:    push    r14
0x1ac96:    movsxd  r10, ecx
0x1ac99:    push    r13
0x1ac9b:    mov     r13, rsi
0x1ac9e:    lea     rsi, [rdi + r10]
0x1aca2:    push    r12
0x1aca4:    push    rbp
0x1aca5:    push    rbx
0x1aca6:    sub     rsp, 0x18
0x1acaa:    movzx   eax, byte ptr [rsi]
0x1acad:    cmp     r8b, al
0x1acb0:    je      0x1af20
0x1acb6:    mov     r8d, 1
0x1acbc:    cmp     al, 0x2d
0x1acbe:    je      0x1aef8
0x1acc4:    lea     edx, [rax - 0x30]
0x1acc7:    cmp     dl, 9
0x1acca:    jbe     0x1ae68
0x1acd0:    mov     ebp, 1
0x1acd5:    xor     ebx, ebx
0x1acd7:    cmp     al, 0x3f
0x1acd9:    jne     0x1acef
0x1acdb:    add     ecx, 1
0x1acde:    xor     ebp, ebp
0x1ace0:    mov     ebx, 1
0x1ace5:    movsxd  r10, ecx
0x1ace8:    lea     rsi, [rdi + r10]
0x1acec:    movzx   eax, byte ptr [rsi]

```

```

0x1acef:    test    al, al
0x1acf1:    je      0x1b042
0x1acf7:    lea     rdx, [rdi + r10 + 1]
0x1acfc:    mov     r12d, ecx
0x1acff:    mov     r8, rsi
0x1ad02:    movabs  r10, 0x400000100000200
0x1ad0c:    jmp     0x1ad48
0x1ad48:
0x1ad0e:    nop
0x1ad10:    cmp     al, 0x3a
0x1ad12:    ja      0x1ad1e
0x1ad14:    bt      r10, rax
0x1ad18:    jb      0x1afd8
0x1ad1e:    movsx   edi, al
0x1ad21:    cmp     edi, r9d
0x1ad24:    je      0x1b010
0x1ad2a:    cmp     al, 0xa
0x1ad2c:    je      0x1ae10
0x1ad32:    mov     r8, rdx
0x1ad35:    movzx   eax, byte ptr [rdx]
0x1ad38:    lea     rdx, [rdx + 1]
0x1ad3c:    add     r12d, 1
0x1ad40:    test    al, al
0x1ad42:    je      0x1ae10
0x1ad48:    test    ebx, ebx
0x1ad4a:    je      0x1ad10
0x1ad4c:    cmp     al, 0xa
0x1ad4e:    je      0x1ae10
0x1ad54:    cmp     al, 0x3f
0x1ad56:    jne     0x1ad32
0x1ad58:    mov     r15d, r12d
0x1ad5b:    mov     qword ptr [rsp + 8], r8
0x1ad60:    sub     r15d, ecx
0x1ad63:    mov     qword ptr [rsp], rsi
0x1ad67:    lea     edi, [r15 + 1]
0x1ad6b:    movsxd  r15, r15d
0x1ad6e:    movsxd  rdi, edi
0x1ad71:    call    0x1d670
0x1ad76:    mov     rsi, qword ptr [rsp]
0x1ad7a:    mov     rdx, r15
0x1ad7d:    mov     rdi, rax
0x1ad80:    mov     r14, rax
0x1ad83:    call    0x4880
0x1ad88:    mov     r8, qword ptr [rsp + 8]
0x1ad8d:    xor     eax, eax
0x1ad8f:    mov     byte ptr [r14 + r15], 0
0x1ad94:    cmp     byte ptr [r8], 0x3f
0x1ad98:    sete    al

```



```

0x1ad9b:      add     r12d, eax
0x1ad9e:      nop
0x1ada0:      mov     dword ptr [r13], r12d
0x1ada4:      mov     r12d, ebx
0x1ada7:      and     r12d, 1
0x1adab:      jmp     0x1add3
0x1add3:
0x1adad:      nop     dword ptr [rax]
0x1adb0:      movsxd  rax, dword ptr [rip + 0x10d45]
0x1adb7:      je      0x1af78
0x1adbδ:      test    r12b, r12b
0x1adc0:      jne     0x1af78
0x1adc6:      test    eax, eax
0x1adc8:      je      0x1ade8
0x1adca:      sub     eax, 1
0x1adcd:      mov     dword ptr [rip + 0x10d29], eax
0x1add3:      xor     eax, eax
0x1add5:      mov     edx, ebp
0x1add7:      mov     esi, 0xffffffff
0x1addc:      mov     rdi, r14
0x1addf:      call    0x19de0
0x1ade4:      test    eax, eax
0x1ade6:      jns     0x1adb0
0x1ade8:      mov     eax, dword ptr [rip + 0x10cf6]
0x1adee:      mov     rdi, r14
0x1adf1:      mov     dword ptr [rip + 0x10d05], eax
0x1adf7:      call    0x18d10
0x1adfc:      add     rsp, 0x18
0x1ae00:      xor     eax, eax
0x1ae02:      pop     rbx
0x1ae03:      pop     rbp
0x1ae04:      pop     r12
0x1ae06:      pop     r13
0x1ae08:      pop     r14
0x1ae0a:      pop     r15
0x1ae0c:      ret
0x1ae0d:      nop     dword ptr [rax]
0x1ae10:      mov     r15d, r12d
0x1ae13:      sub     r15d, ecx
0x1ae16:      lea     edi, [r15 + 1]
0x1ae1a:      movsxd  r15, r15d
0x1ae1d:      movsxd  rdi, edi
0x1ae20:      mov     qword ptr [rsp + 8], r8
0x1ae25:      mov     qword ptr [rsp], rsi
0x1ae29:      call    0x1d670
0x1ae2e:      mov     rsi, qword ptr [rsp]
0x1ae32:      mov     rdx, r15
0x1ae35:      mov     rdi, rax

```

```

0x1ae38:    mov     r14, rax
0x1ae3b:    call   0x4880
0x1ae40:    test    ebx, ebx
0x1ae42:    mov     byte ptr [r14 + r15], 0
0x1ae47:    mov     r8, qword ptr [rsp + 8]
0x1ae4c:    je      0x1ada0
0x1ae52:    xor     eax, eax
0x1ae54:    cmp     byte ptr [r8], 0x3f
0x1ae58:    sete    al
0x1ae5b:    add     r12d, eax
0x1ae5e:    jmp     0x1ada0
0x1ada0:
0x1ae63:    nop     dword ptr [rax + rax]
0x1ae68:    lea     edx, [rcx + 1]
0x1ae6b:    xor     ecx, ecx
0x1ae6d:    movsxd  rdx, edx
0x1ae70:    sub     eax, 0x30
0x1ae73:    lea     ecx, [rcx + rcx*4]
0x1ae76:    mov     r9d, edx
0x1ae79:    movsx   eax, al
0x1ae7c:    lea     ecx, [rax + rcx*2]
0x1ae7f:    movzx   eax, byte ptr [rdi + rdx]
0x1ae83:    add     rdx, 1
0x1ae87:    lea     esi, [rax - 0x30]
0x1ae8a:    cmp     sil, 9
0x1ae8e:    jbe     0x1ae70
0x1ae90:    mov     dword ptr [r13], r9d
0x1ae94:    mov     eax, dword ptr [rip + 0x10c4a]
0x1ae9a:    mov     edx, dword ptr [rip + 0xf948]
0x1aea0:    cmp     r8d, -1
0x1aea4:    jne     0x1aead
0x1aea6:    lea     esi, [rax + rdx]
0x1aea9:    sub     esi, ecx
0x1aeab:    mov     ecx, esi
0x1aead:    sub     ecx, edx
0x1aeaf:    cmp     ecx, eax
0x1aeb1:    jge     0x1adfc
0x1aeb7:    test    ecx, ecx
0x1aeb9:    js      0x1adfc
0x1aebf:    mov     rax, qword ptr [rip + 0x10c3a]
0x1aec6:    test    rax, rax
0x1aec9:    je      0x1adfc
0x1aecf:    movsxd  rcx, ecx
0x1aed2:    mov     rax, qword ptr [rax + rcx*8]
0x1aed6:    test    rax, rax
0x1aed9:    je      0x1adfc
0x1aedf:    mov     rax, qword ptr [rax]
0x1aee2:    add     rsp, 0x18

```

```

0x1aee6:    pop     rbx
0x1aee7:    pop     rbp
0x1aee8:    pop     r12
0x1aeea:    pop     r13
0x1aeec:    pop     r14
0x1aeed:    pop     r15
0x1aef0:    ret
0x1aef1:    nop     dword ptr [rax]
0x1aef8:    lea     ecx, [rdx + 2]
0x1aefb:    mov     r8d, 0xffffffff
0x1af01:    movsxd  r10, ecx
0x1af04:    lea     rsi, [rdi + r10]
0x1af08:    movzx   eax, byte ptr [rsi]
0x1af0b:    lea     edx, [rax - 0x30]
0x1af0e:    cmp     dl, 9
0x1af11:    ja      0x1acd0
0x1af17:    jmp     0x1ae68
0x1ae68:
0x1af1c:    nop     dword ptr [rax]
0x1af20:    mov     ecx, dword ptr [rip + 0x10bbe]
0x1af26:    mov     eax, dword ptr [rip + 0xf8bc]
0x1af2c:    add     edx, 2
0x1af2f:    mov     dword ptr [r13], edx
0x1af33:    lea     eax, [rcx + rax - 1]
0x1af37:    sub     eax, dword ptr [rip + 0xf8ab]
0x1af3d:    cmp     eax, dword ptr [rip + 0x10ba1]
0x1af43:    jge     0x1adfc
0x1af49:    test    eax, eax
0x1af4b:    js      0x1adfc
0x1af51:    mov     rdx, qword ptr [rip + 0x10ba8]
0x1af58:    test    rdx, rdx
0x1af5b:    je      0x1adfc
0x1af61:    cdqe
0x1af63:    mov     rax, qword ptr [rdx + rax*8]
0x1af67:    test    rax, rax
0x1af6a:    jne     0x1aedf
0x1af70:    jmp     0x1adfc
0x1adfc:
0x1af75:    nop     dword ptr [rax]
0x1af78:    mov     edx, dword ptr [rip + 0x10b66]
0x1af7e:    cmp     edx, eax
0x1af80:    je      0x1b03b
0x1af86:    mov     rbp, qword ptr [rip + 0x10b73]
0x1af8d:    test    rbp, rbp
0x1af90:    je      0x1af97
0x1af92:    mov     rbp, qword ptr [rbp + rax*8]
0x1af97:    mov     dword ptr [rip + 0x10b5f], edx
0x1af9d:    test    ebx, ebx

```

```

0x1af9f:      je      0x1b02e
0x1afa5:      mov     rdi, qword ptr [rip + 0x10b44]
0x1afac:      test    rdi, rdi
0x1afaf:      je      0x1afb6
0x1afb1:      call    0x18d10
0x1afb6:      mov     qword ptr [rip + 0x10b33], r14
0x1afbd:      mov     rax, qword ptr [rbp]
0x1afc1:      add     rsp, 0x18
0x1afc5:      pop     rbx
0x1afc6:      pop     rbp
0x1afc7:      pop     r12
0x1afc9:      pop     r13
0x1afcb:      pop     r14
0x1afcd:      pop     r15
0x1afcf:      ret
0x1afd0:      xor     eax, eax
0x1afd2:      ret
0x1afd3:      nop     dword ptr [rax + rax]
0x1afd8:      mov     eax, r12d
0x1afdb:      mov     qword ptr [rsp], rsi
0x1afdf:      sub     eax, ecx
0x1afe1:      movsxd  r15, eax
0x1afe4:      add     eax, 1
0x1afe7:      movsxd  rdi, eax
0x1afea:      call    0x1d670
0x1afef:      mov     r14, rax
0x1aff2:      mov     rsi, qword ptr [rsp]
0x1aff6:      mov     rdx, r15
0x1aff9:      mov     rdi, r14
0x1affc:      call    0x4880
0x1b001:      mov     byte ptr [r14 + r15], 0
0x1b006:      jmp     0x1ada0
0x1ada0
0x1b00b:      nop     dword ptr [rax + rax]
0x1b010:      mov     r15d, r12d
0x1b013:      mov     qword ptr [rsp], rsi
0x1b017:      sub     r15d, ecx
0x1b01a:      lea     edi, [r15 + 1]
0x1b01e:      movsxd  r15, r15d
0x1b021:      movsxd  rdi, edi
0x1b024:      call    0x1d670
0x1b029:      mov     r14, rax
0x1b02c:      jmp     0x1aff2
0x1aff2
0x1b02e:      mov     rdi, r14
0x1b031:      call    0x18d10
0x1b036:      jmp     0x1afbd
0x1afbd

```

```

0x1b03b:    xor     ebp, ebp
0x1b03d:    jmp     0x1af97
0x1af97:
0x1b042:    mov     r12d, ecx
0x1b045:    mov     r8, rsi
0x1b048:    xor     r15d, r15d
0x1b04b:    mov     edi, 1
0x1b050:    jmp     0x1ae20
0x1ae20:
0x1b055:    nop     word ptr cs:[rax + rax]
0x1b060:    endbr64
0x1b064:    push    r15
0x1b066:    push    r14
0x1b068:    push    r13
0x1b06a:    push    r12
0x1b06c:    push    rbp
0x1b06d:    push    rbx
0x1b06e:    sub     rsp, 0x28
0x1b072:    movzx   r14d, byte ptr [rdi]
0x1b076:    test    r14b, r14b
0x1b079:    je      0x1b30d
0x1b07f:    mov     qword ptr [rsp], 0
0x1b087:    mov     r12, rdi
0x1b08a:    mov     r13, rdi
0x1b08d:    xor     ebx, ebx
0x1b08f:    mov     dword ptr [rsp + 8], 0
0x1b097:    xor     ebp, ebp
0x1b099:    mov     dword ptr [rsp + 0xc], 0
0x1b0a1:    jmp     0x1b0c0
0x1b0c0:
0x1b0a3:    nop     dword ptr [rax + rax]
0x1b0a8:    add     ebx, 1
0x1b0ab:    movsxd  rbp, ebx
0x1b0ae:    lea     r13, [r12 + rbp]
0x1b0b2:    movzx   r14d, byte ptr [r13]
0x1b0b7:    test    r14b, r14b
0x1b0ba:    je      0x1b315
0x1b0c0:    cmp     r14b, 0x20
0x1b0c4:    je      0x1b0a8
0x1b0c6:    cmp     r14b, 9
0x1b0ca:    je      0x1b0a8
0x1b0cc:    cmp     byte ptr [rip + 0x10a0d], r14b
0x1b0d3:    je      0x1b315
0x1b0d9:    movsx   r15d, r14b
0x1b0dd:    lea     rdi, [rip + 0x7379]
0x1b0e4:    mov     esi, r15d
0x1b0e7:    call    0x4a00
0x1b0ec:    test    rax, rax

```

```

0x1b0ef:      je      0x1b190
0x1b0f5:      add     ebx, 1
0x1b0f8:      mov     r14d, 1
0x1b0fe:      mov     r9d, 2
0x1b104:      movsxd   rbp, ebx
0x1b107:      lea     rcx, [r12 + rbp]
0x1b10b:      mov     eax, dword ptr [rsp + 8]
0x1b10f:      add     eax, 2
0x1b112:      cmp     eax, dword ptr [rsp + 0xc]
0x1b116:      jge     0x1b2a0
0x1b11c:      movsxd   r8, dword ptr [rsp + 8]
0x1b121:      mov     rdx, qword ptr [rsp]
0x1b125:      mov     rdi, r9
0x1b128:      mov     qword ptr [rsp + 0x18], rcx
0x1b12d:      shl     r8, 3
0x1b131:      lea     r15, [rdx + r8]
0x1b135:      mov     qword ptr [rsp + 0x10], r8
0x1b13a:      call    0x1d670
0x1b13f:      mov     rdx, r14
0x1b142:      mov     rsi, r13
0x1b145:      mov     qword ptr [r15], rax
0x1b148:      mov     rdi, rax
0x1b14b:      call    0x4880
0x1b150:      mov     rax, qword ptr [r15]
0x1b153:      mov     r8, qword ptr [rsp + 0x10]
0x1b158:      mov     rcx, qword ptr [rsp + 0x18]
0x1b15d:      add     dword ptr [rsp + 8], 1
0x1b162:      mov     byte ptr [rax + r14], 0
0x1b167:      mov     rax, qword ptr [rsp]
0x1b16b:      mov     qword ptr [rax + r8 + 8], 0
0x1b174:      movzx   r14d, byte ptr [rcx]
0x1b178:      test    r14b, r14b
0x1b17b:      je      0x1b315
0x1b181:      mov     r13, rcx
0x1b184:      jmp     0x1b0c0
0x1b0c0:
0x1b189:      nop     dword ptr [rax]
0x1b190:      mov     esi, r15d
0x1b193:      lea     rdi, [rip + 0x72c7]
0x1b19a:      call    0x4a00
0x1b19f:      test    rax, rax
0x1b1a2:      je      0x1b369
0x1b1a8:      movsx   edx, byte ptr [r12 + rbp + 1]
0x1b1ae:      cmp     dl, r14b
0x1b1b1:      je      0x1b2d8
0x1b1b7:      cmp     edx, 0x26
0x1b1ba:      je      0x1b340
0x1b1c0:      cmp     edx, 0x3e

```

```

0x1b1c3:    jne     0x1b3ac
0x1b1c9:    cmp     r14b, 0x26
0x1b1cd:    je      0x1b34e
0x1b1d3:    cmp     r14b, 0x24
0x1b1d7:    jne     0x1b0f5
0x1b1dd:    mov     edx, ebx
0x1b1df:    xor     r15d, r15d
0x1b1e2:    movsxd  r8, edx
0x1b1e5:    lea     rcx, [r12 + r8]
0x1b1e9:    mov     rbp, r8
0x1b1ec:    movzx   eax, byte ptr [rcx]
0x1b1ef:    test    al, al
0x1b1f1:    je      0x1b404
0x1b1f7:    mov     qword ptr [rsp + 0x10], r13
0x1b1fc:    lea     r14, [rip + 0x7265]
0x1b203:    mov     r13, r8
0x1b206:    mov     dword ptr [rsp + 0x18], ebx
0x1b20a:    mov     rbx, r12
0x1b20d:    mov     r12d, edx
0x1b210:    jmp     0x1b265
0x1b265:
0x1b212:    nop     word ptr [rax + rax]
0x1b218:    movsx   ebp, al
0x1b21b:    test    r15d, r15d
0x1b21e:    jne     0x1b290
0x1b220:    mov     esi, ebp
0x1b222:    lea     rdi, [rip + 0x7243]
0x1b229:    call    0x4a00
0x1b22e:    test    rax, rax
0x1b231:    jne     0x1b3c0
0x1b237:    mov     esi, ebp
0x1b239:    mov     rdi, r14
0x1b23c:    call    0x4a00
0x1b241:    lea     esi, [r12 + 1]
0x1b246:    test    rax, rax
0x1b249:    cmovne  r15d, ebp
0x1b24d:    movsxd  rbp, esi
0x1b250:    mov     r12d, esi
0x1b253:    lea     rcx, [rbx + rbp]
0x1b257:    mov     r13, rbp
0x1b25a:    movzx   eax, byte ptr [rcx]
0x1b25d:    test    al, al
0x1b25f:    je      0x1b389
0x1b265:    cmp     al, 0x5c
0x1b267:    jne     0x1b218
0x1b269:    cmp     byte ptr [rbx + r13 + 1], 0xa
0x1b26f:    lea     esi, [r12 + 1]
0x1b274:    je      0x1b330

```

```

0x1b27a:    xor     eax, eax
0x1b27c:    cmp     r15d, 0x27
0x1b280:    setne   al
0x1b283:    add     esi, eax
0x1b285:    jmp     0x1b24d
0x1b24d
0x1b287:    nop     word ptr [rax + rax]
0x1b290:    xor     eax, eax
0x1b292:    cmp     r15d, ebp
0x1b295:    lea     esi, [r12 + 1]
0x1b29a:    cmov     r15d, eax
0x1b29e:    jmp     0x1b24d
0x1b24d
0x1b2a0:    add     dword ptr [rsp + 0xc], 0xa
0x1b2a5:    movsxd  rsi, dword ptr [rsp + 0xc]
0x1b2aa:    mov     rdi, qword ptr [rsp]
0x1b2ae:    mov     qword ptr [rsp + 0x18], rcx
0x1b2b3:    shl     rsi, 3
0x1b2b7:    mov     qword ptr [rsp + 0x10], r9
0x1b2bc:    call    0x1d690
0x1b2c1:    mov     rcx, qword ptr [rsp + 0x18]
0x1b2c6:    mov     r9, qword ptr [rsp + 0x10]
0x1b2cb:    mov     qword ptr [rsp], rax
0x1b2cf:    jmp     0x1b11c
0x1b11c
0x1b2d4:    nop     dword ptr [rax]
0x1b2d8:    cmp     edx, 0x24
0x1b2db:    je      0x1b1d3
0x1b2e1:    mov     eax, ebx
0x1b2e3:    cmp     edx, 0x3c
0x1b2e6:    je      0x1b3f0
0x1b2ec:    add     eax, 2
0x1b2ef:    mov     r14d, eax
0x1b2f2:    movsxd  rbp, eax
0x1b2f5:    sub     r14d, ebx
0x1b2f8:    lea     rcx, [r12 + rbp]
0x1b2fc:    mov     ebx, eax
0x1b2fe:    lea     r9d, [r14 + 1]
0x1b302:    movsxd  r14, r14d
0x1b305:    movsxd  r9, r9d
0x1b308:    jmp     0x1b10b
0x1b10b
0x1b30d:    mov     qword ptr [rsp], 0
0x1b315:    mov     rax, qword ptr [rsp]
0x1b319:    add     rsp, 0x28
0x1b31d:    pop     rbx
0x1b31e:    pop     rbp
0x1b31f:    pop     r12

```



```

0x1b321:    pop     r13
0x1b323:    pop     r14
0x1b325:    pop     r15
0x1b327:    ret
0x1b328:    nop     dword ptr [rax + rax]
0x1b330:    lea     esi, [r12 + 2]
0x1b335:    jmp     0x1b24d
0x1b24d
0x1b33a:    nop     word ptr [rax + rax]
0x1b340:    mov     eax, r14d
0x1b343:    and     eax, 0xffffffff
0x1b346:    cmp     al, 0x3c
0x1b348:    jne     0x1b1d3
0x1b34e:    add     ebx, 2
0x1b351:    mov     r14d, 2
0x1b357:    mov     r9d, 3
0x1b35d:    movsxd  rbp, ebx
0x1b360:    lea     rcx, [r12 + rbp]
0x1b364:    jmp     0x1b10b
0x1b10b
0x1b369:    mov     esi, r15d
0x1b36c:    lea     rdi, [rip + 0x70f5]
0x1b373:    call    0x4a00
0x1b378:    test    rax, rax
0x1b37b:    je      0x1b1dd
0x1b381:    lea     edx, [rbx + 1]
0x1b384:    jmp     0x1b1e2
0x1b1e2
0x1b389:    mov     r12, rbx
0x1b38c:    mov     ebx, dword ptr [rsp + 0x18]
0x1b390:    mov     r14d, esi
0x1b393:    mov     r13, qword ptr [rsp + 0x10]
0x1b398:    sub     r14d, ebx
0x1b39b:    mov     ebx, esi
0x1b39d:    lea     r9d, [r14 + 1]
0x1b3a1:    movsxd  r14, r14d
0x1b3a4:    movsxd  r9, r9d
0x1b3a7:    jmp     0x1b10b
0x1b10b
0x1b3ac:    cmp     r14b, 0x24
0x1b3b0:    jne     0x1b0f5
0x1b3b6:    cmp     edx, 0x28
0x1b3b9:    je      0x1b34e
0x1b3bb:    jmp     0x1b1d3
0x1b1d3
0x1b3c0:    mov     edx, r12d
0x1b3c3:    mov     r12, rbx
0x1b3c6:    mov     ebx, dword ptr [rsp + 0x18]

```

```

0x1b3ca:      mov     r8, r13
0x1b3cd:      mov     r14d, edx
0x1b3d0:      mov     r13, qword ptr [rsp + 0x10]
0x1b3d5:      mov     rbp, r8
0x1b3d8:      lea     rcx, [r12 + r8]
0x1b3dc:      sub     r14d, ebx
0x1b3df:      mov     ebx, edx
0x1b3e1:      lea     r9d, [r14 + 1]
0x1b3e5:      movsxd  r14, r14d
0x1b3e8:      movsxd  r9, r9d
0x1b3eb:      jmp     0x1b10b
0x1b10b
0x1b3f0:      cmp     byte ptr [r12 + rbp + 2], 0x2d
0x1b3f6:      jne     0x1b2ec
0x1b3fc:      lea     eax, [rbx + 1]
0x1b3ff:      jmp     0x1b2ec
0x1b2ec
0x1b404:      mov     r14d, edx
0x1b407:      sub     r14d, ebx
0x1b40a:      mov     ebx, edx
0x1b40c:      lea     r9d, [r14 + 1]
0x1b410:      movsxd  r14, r14d
0x1b413:      movsxd  r9, r9d
0x1b416:      jmp     0x1b10b
0x1b10b
0x1b41b:      nop     dword ptr [rax + rax]
0x1b420:      endbr64
0x1b424:      push    r15
0x1b426:      xor     eax, eax
0x1b428:      push    r14
0x1b42a:      push    r13
0x1b42c:      push    r12
0x1b42e:      mov     r12d, esi
0x1b431:      push    rbp
0x1b432:      push    rbx
0x1b433:      mov     ebx, edi
0x1b435:      mov     rdi, rdx
0x1b438:      sub     rsp, 8
0x1b43c:      call    0x1b060
0x1b441:      test    rax, rax
0x1b444:      je      0x1b598
0x1b44a:      cmp     qword ptr [rax], 0
0x1b44e:      mov     rbp, rax
0x1b451:      je      0x1b590
0x1b457:      mov     eax, 1
0x1b45c:      nop     dword ptr [rax]
0x1b460:      mov     r14d, eax
0x1b463:      add     rax, 1

```

```

0x1b467:    cmp     qword ptr [rbp + rax*8 - 8], 0
0x1b46d:    jne     0x1b460
0x1b46f:    lea     eax, [r14 + r12 - 1]
0x1b474:    test    r12d, r12d
0x1b477:    cmovs   r12d, eax
0x1b47b:    lea     eax, [r14 + rbx - 1]
0x1b480:    test    ebx, ebx
0x1b482:    cmovs   ebx, eax
0x1b485:    lea     eax, [r12 + 1]
0x1b48a:    cmp     r12d, 0x24
0x1b48e:    jne     0x1b497
0x1b490:    lea     r12d, [r14 - 1]
0x1b494:    mov     eax, r14d
0x1b497:    lea     edx, [r14 - 1]
0x1b49b:    cmp     ebx, 0x24
0x1b49e:    cmov     ebx, edx
0x1b4a1:    mov     edx, eax
0x1b4a3:    cmp     ebx, eax
0x1b4a5:    cmovge   edx, ebx
0x1b4a8:    xor     r13d, r13d
0x1b4ab:    cmp     edx, r14d
0x1b4ae:    jg       0x1b547
0x1b4b4:    mov     ecx, ebx
0x1b4b6:    or      ecx, eax
0x1b4b8:    js       0x1b547
0x1b4be:    cmp     ebx, eax
0x1b4c0:    jge     0x1b59d
0x1b4c6:    movsxd   rbx, ebx
0x1b4c9:    xor     r15d, r15d
0x1b4cc:    mov     r13, rbx
0x1b4cf:    nop
0x1b4d0:    mov     rdi, qword ptr [rbp + r13*8]
0x1b4d5:    add     r13, 1
0x1b4d9:    call    0x49b0
0x1b4de:    lea     r15d, [r15 + rax + 1]
0x1b4e3:    lea     eax, [r13 - 1]
0x1b4e7:    cmp     r12d, eax
0x1b4ea:    jg       0x1b4d0
0x1b4ec:    lea     edi, [r15 + 1]
0x1b4f0:    xor     r15d, r15d
0x1b4f3:    movsxd   rdi, edi
0x1b4f6:    call    0x1d670
0x1b4fb:    mov     r13, rax
0x1b4fe:    nop
0x1b500:    mov     rsi, qword ptr [rbp + rbx*8]
0x1b505:    movsxd   rdi, r15d
0x1b508:    add     rdi, r13
0x1b50b:    call    0x48b0

```

```

0x1b510:    mov     rdi, qword ptr [rbp + rbx*8]
0x1b515:    call    0x49b0
0x1b51a:    add     r15d, eax
0x1b51d:    cmp     r12d, ebx
0x1b520:    jle     0x1b53b
0x1b522:    lea     eax, [r15 + 1]
0x1b526:    movsxd  r15, r15d
0x1b529:    movsxd  rdx, eax
0x1b52c:    mov     byte ptr [r13 + r15], 0x20
0x1b532:    mov     r15d, eax
0x1b535:    mov     byte ptr [r13 + rdx], 0
0x1b53b:    add     rbx, 1
0x1b53f:    lea     eax, [rbx - 1]
0x1b542:    cmp     r12d, eax
0x1b545:    jg      0x1b500
0x1b547:    test    r14d, r14d
0x1b54a:    je      0x1b571
0x1b54c:    lea     eax, [r14 - 1]
0x1b550:    mov     rbx, rbp
0x1b553:    lea     r12, [rbp + rax*8 + 8]
0x1b558:    nop     dword ptr [rax + rax]
0x1b560:    mov     rdi, qword ptr [rbx]
0x1b563:    add     rbx, 8
0x1b567:    call    0x18d10
0x1b56c:    cmp     r12, rbx
0x1b56f:    jne     0x1b560
0x1b571:    mov     rdi, rbp
0x1b574:    call    0x18d10
0x1b579:    add     rsp, 8
0x1b57d:    mov     rax, r13
0x1b580:    pop     rbx
0x1b581:    pop     rbp
0x1b582:    pop     r12
0x1b584:    pop     r13
0x1b586:    pop     r14
0x1b588:    pop     r15
0x1b58a:    ret
0x1b58b:    nop     dword ptr [rax + rax]
0x1b590:    xor     r14d, r14d
0x1b593:    jmp     0x1b46f
0x1b46f:
0x1b598:    xor     r13d, r13d
0x1b59b:    jmp     0x1b579
0x1b579:
0x1b59d:    mov     edi, 1
0x1b5a2:    call    0x1d670
0x1b5a7:    mov     r13, rax
0x1b5aa:    jmp     0x1b547

```

```

0x1b547
0x1b5ac:      nop      dword ptr [rax]
0x1b5b0:      endbr64
0x1b5b4:      push     r15
0x1b5b6:      push     r14
0x1b5b8:      push     r13
0x1b5ba:      push     r12
0x1b5bc:      push     rbp
0x1b5bd:      push     rbx
0x1b5be:      mov      rbx, rdi
0x1b5c1:      sub      rsp, 0x88
0x1b5c8:      mov      qword ptr [rsp + 0x20], rdi
0x1b5cd:      mov      qword ptr [rsp + 0x30], rsi
0x1b5d2:      mov      rax, qword ptr fs:[0x28]
0x1b5db:      mov      qword ptr [rsp + 0x78], rax
0x1b5e0:      xor      eax, eax
0x1b5e2:      cmp      byte ptr [rip + 0xf204], 0
0x1b5e9:      je       0x1b656
0x1b5eb:      mov      edi, 0x100
0x1b5f0:      call     0x1d670
0x1b5f5:      mov      rbx, qword ptr [rsp + 0x20]
0x1b5fa:      mov      byte ptr [rax], 0
0x1b5fd:      mov      r15, rax
0x1b600:      mov      rdi, rbx
0x1b603:      call     0x49b0
0x1b608:      mov      dword ptr [rsp + 0xc], eax
0x1b60c:      mov      r12, rax
0x1b60f:      movzx    eax, byte ptr [rbx]
0x1b612:      cmp      al, byte ptr [rip + 0xf1d4]
0x1b618:      je       0x1c1e0
0x1b61e:      movzx    ebp, byte ptr [rip + 0xf1c8]
0x1b625:      mov      r13, qword ptr [rip + 0xf1c4]
0x1b62c:      lea      rbx, [rbx + 1]
0x1b630:      test     al, al
0x1b632:      jne      0x1b6c7
0x1b638:      nop      dword ptr [rax + rax]
0x1b640:      test     bpl, bpl
0x1b643:      je       0x1b6d8
0x1b649:      mov      rdi, r15
0x1b64c:      call     0x18d10
0x1b651:      mov      rbx, qword ptr [rsp + 0x20]
0x1b656:      mov      rdi, rbx
0x1b659:      call     0x49b0
0x1b65e:      lea      rdi, [rax + 1]
0x1b662:      call     0x1d670
0x1b667:      mov      rsi, rbx
0x1b66a:      mov      rdi, rax
0x1b66d:      call     0x48b0

```

```

0x1b672:    mov     rcx, qword ptr [rsp + 0x30]
0x1b677:    mov     qword ptr [rcx], rax
0x1b67a:    xor     eax, eax
0x1b67c:    mov     rdx, qword ptr [rsp + 0x78]
0x1b681:    sub     rdx, qword ptr fs:[0x28]
0x1b68a:    jne     0x1c487
0x1b690:    add     rsp, 0x88
0x1b697:    pop     rbx
0x1b698:    pop     rbp
0x1b699:    pop     r12
0x1b69b:    pop     r13
0x1b69d:    pop     r14
0x1b69f:    pop     r15
0x1b6a1:    ret
0x1b6a2:    nop     word ptr [rax + rax]
0x1b6a8:    mov     rdi, r13
0x1b6ab:    call    0x4a00
0x1b6b0:    test    rax, rax
0x1b6b3:    je      0x1b6d8
0x1b6b5:    nop     dword ptr [rax]
0x1b6b8:    movzx   eax, byte ptr [rbx]
0x1b6bb:    add     rbx, 1
0x1b6bf:    test    al, al
0x1b6c1:    je      0x1b640
0x1b6c7:    movsx   esi, byte ptr [rbx]
0x1b6ca:    cmp     al, bpl
0x1b6cd:    jne     0x1b6b8
0x1b6cf:    test    esi, esi
0x1b6d1:    je      0x1b6b8
0x1b6d3:    jmp     0x1b6a8
0x1b6a8:
0x1b6d5:    nop     dword ptr [rax]
0x1b6d8:    mov     rbp, qword ptr [rsp + 0x20]
0x1b6dd:    test    r12d, r12d
0x1b6e0:    jle     0x1c40a
0x1b6e6:    mov     dword ptr [rsp + 4], 0x100
0x1b6ee:    xor     r13d, r13d
0x1b6f1:    xor     ebx, ebx
0x1b6f3:    mov     dword ptr [rsp + 0x28], 0
0x1b6fb:    mov     dword ptr [rsp + 0x3c], 0
0x1b703:    mov     dword ptr [rsp + 8], 0
0x1b70b:    jmp     0x1b758
0x1b758:
0x1b70d:    nop     dword ptr [rax]
0x1b710:    cmp     r12b, byte ptr [rip + 0xf0d6]
0x1b717:    je      0x1b7d0
0x1b71d:    cmp     r12b, 0xfd
0x1b721:    je      0x1b7d0

```

```

0x1b727:    cmp     r12b, 0x5c
0x1b72b:    je      0x1bde0
0x1b731:    cmp     ebx, dword ptr [rsp + 4]
0x1b735:    jge     0x1bfe0
0x1b73b:    lea     eax, [rbx + 1]
0x1b73e:    mov     byte ptr [r15 + rbx], r12b
0x1b742:    movsxd  rdx, eax
0x1b745:    movsxd  rbx, eax
0x1b748:    mov     byte ptr [r15 + rdx], 0
0x1b74d:    add     r13d, 1
0x1b751:    cmp     r13d, dword ptr [rsp + 0xc]
0x1b756:    jge     0x1b79d
0x1b758:    movsxd  r14, r13d
0x1b75b:    mov     r10d, dword ptr [rsp + 8]
0x1b760:    lea     rdx, [rbp + r14]
0x1b765:    movzx   r12d, byte ptr [rdx]
0x1b769:    test    r10d, r10d
0x1b76c:    je      0x1b710
0x1b76e:    cmp     ebx, dword ptr [rsp + 4]
0x1b772:    jge     0x1be10
0x1b778:    lea     eax, [rbx + 1]
0x1b77b:    mov     byte ptr [r15 + rbx], r12b
0x1b77f:    add     r13d, 1
0x1b783:    movsxd  rdx, eax
0x1b786:    mov     dword ptr [rsp + 8], 0
0x1b78e:    movsxd  rbx, eax
0x1b791:    mov     byte ptr [r15 + rdx], 0
0x1b796:    cmp     r13d, dword ptr [rsp + 0xc]
0x1b79b:    jl      0x1b758
0x1b79d:    mov     rax, qword ptr [rsp + 0x30]
0x1b7a2:    mov     qword ptr [rax], r15
0x1b7a5:    cmp     qword ptr [rsp + 0x20], rbp
0x1b7aa:    je      0x1b7b4
0x1b7ac:    mov     rdi, rbp
0x1b7af:    call    0x18d10
0x1b7b4:    mov     ecx, dword ptr [rsp + 0x28]
0x1b7b8:    test    ecx, ecx
0x1b7ba:    jne     0x1c35b
0x1b7c0:    mov     edx, dword ptr [rsp + 0x3c]
0x1b7c4:    xor     eax, eax
0x1b7c6:    test    edx, edx
0x1b7c8:    setne   al
0x1b7cb:    jmp     0x1b67c
0x1b67c:
0x1b7d0:    lea     rcx, [rbp + r14 + 1]
0x1b7d5:    mov     qword ptr [rsp + 0x10], rdx
0x1b7da:    movsx   esi, byte ptr [rcx]
0x1b7dd:    test    esi, esi

```

```

0x1b7df:      je      0x1b830
0x1b7e1:      mov     rdi, qword ptr [rip + 0xf008]
0x1b7e8:      mov     qword ptr [rsp + 0x18], rcx
0x1b7ed:      call    0x4a00
0x1b7f2:      mov     rcx, qword ptr [rsp + 0x18]
0x1b7f7:      test    rax, rax
0x1b7fa:      je      0x1b830
0x1b7fc:      cmp     ebx, dword ptr [rsp + 4]
0x1b800:      jl      0x1b73b
0x1b806:      add     dword ptr [rsp + 4], 0x40
0x1b80b:      movsxd   rsi, dword ptr [rsp + 4]
0x1b810:      mov     rdi, r15
0x1b813:      call    0x1d690
0x1b818:      mov     rdx, qword ptr [rsp + 0x10]
0x1b81d:      mov     r15, rax
0x1b820:      movzx    r12d, byte ptr [rdx]
0x1b824:      jmp     0x1b73b
0x1b73b:
0x1b829:      nop     dword ptr [rax]
0x1b830:      mov     edi, 0x80
0x1b835:      mov     qword ptr [rsp + 0x18], rcx
0x1b83a:      call    0x1d670
0x1b83f:      mov     rcx, qword ptr [rsp + 0x18]
0x1b844:      mov     dword ptr [rsp + 0x6c], r13d
0x1b849:      mov     qword ptr [rsp + 0x10], rax
0x1b84e:      movzx    r12d, byte ptr [rcx]
0x1b852:      test    r12b, r12b
0x1b855:      je      0x1be30
0x1b85b:      movsx    esi, r12b
0x1b85f:      lea     rdi, [rip + 0x6c11]
0x1b866:      call    0x4a00
0x1b86b:      test    rax, rax
0x1b86e:      je      0x1bfc0
0x1b874:      lea     eax, [r13 + 1]
0x1b878:      lea     rsi, [rsp + 0x70]
0x1b87d:      xor     edx, edx
0x1b87f:      mov     byte ptr [rsp + 0x77], 0
0x1b884:      mov     dword ptr [rsp + 0x6c], eax
0x1b888:      movzx    eax, byte ptr [rip + 0xef5e]
0x1b88f:      lea     rdi, [rsp + 0x75]
0x1b894:      mov     dword ptr [rsp + 0x70], 0
0x1b89c:      mov     byte ptr [rsp + 0x76], al
0x1b8a0:      mov     byte ptr [rsp + 0x75], al
0x1b8a4:      xor     eax, eax
0x1b8a6:      call    0x1ac70
0x1b8ab:      mov     r12d, dword ptr [rsp + 0x6c]
0x1b8b0:      mov     r14, rax
0x1b8b3:      test    r14, r14

```



```

0x1b8b6:      je      0x1c435
0x1b8bc:      movsxd  rax, r12d
0x1b8bf:      xor     esi, esi
0x1b8c1:      mov     edx, r12d
0x1b8c4:      movzx   eax, byte ptr [rbp + rax]
0x1b8c9:      cmp     al, 0x3a
0x1b8cb:      jne     0x1b8df
0x1b8cd:      lea     edx, [r12 + 1]
0x1b8d2:      mov     esi, 1
0x1b8d7:      movsxd  rax, edx
0x1b8da:      movzx   eax, byte ptr [rbp + rax]
0x1b8df:      cmp     al, 0x25
0x1b8e1:      je      0x1c010
0x1b8e7:      cmp     al, 0x2d
0x1b8e9:      je      0x1c100
0x1b8ef:      jg      0x1bd48
0x1b8f5:      cmp     al, 0x24
0x1b8f7:      je      0x1bf80
0x1b8fd:      cmp     al, 0x2a
0x1b8ff:      jne     0x1bd54
0x1b905:      add     edx, 1
0x1b908:      mov     esi, 0x24
0x1b90d:      mov     edi, 1
0x1b912:      xor     eax, eax
0x1b914:      mov     dword ptr [rsp + 0x6c], edx
0x1b918:      mov     rdx, r14
0x1b91b:      call    0x1b420
0x1b920:      mov     r13, rax
0x1b923:      test    rax, rax
0x1b926:      je      0x1c080
0x1b92c:      lea     rax, [rip + 0x101ed]
0x1b933:      cmp     r13, rax
0x1b936:      je      0x1bdade
0x1b93c:      mov     rdi, r13
0x1b93f:      call    0x49b0
0x1b944:      lea     rdi, [rax + 1]
0x1b948:      call    0x1d670
0x1b94d:      mov     rsi, r13
0x1b950:      mov     rdi, rax
0x1b953:      call    0x48b0
0x1b958:      mov     rdi, r13
0x1b95b:      mov     r14, rax
0x1b95e:      call    0x18d10
0x1b963:      movsxd  rcx, dword ptr [rsp + 0x6c]
0x1b968:      cmp     byte ptr [rbp + rcx], 0x3a
0x1b96d:      mov     dword ptr [rsp + 0x44], ecx
0x1b971:      jne     0x1c3d4
0x1b977:      mov     dword ptr [rsp + 0x18], 0

```

```

0x1b97f:    lea     r12, [rip + 0x6afe]
0x1b986:    mov     dword ptr [rsp + 0x38], 0
0x1b98e:    mov     dword ptr [rsp + 0x40], ebx
0x1b992:    mov     qword ptr [rsp + 0x28], r15
0x1b997:    mov     r15, r14
0x1b99a:    mov     r14d, ecx
0x1b99d:    nop     dword ptr [rax]
0x1b9a0:    movzx   eax, byte ptr [rbp + rcx + 1]
0x1b9a5:    cmp     al, 0x67
0x1b9a7:    jne     0x1b9bf
0x1b9a9:    add     r14d, 1
0x1b9ad:    mov     dword ptr [rsp + 0x18], 1
0x1b9b5:    mov     dword ptr [rsp + 0x6c], r14d
0x1b9ba:    movzx   eax, byte ptr [rbp + rcx + 2]
0x1b9bf:    cmp     al, 0x26
0x1b9c1:    je      0x1be70
0x1b9c7:    sub     eax, 0x65
0x1b9ca:    cmp     al, 0xf
0x1b9cc:    ja      0x1b9e0
0x1b9ce:    movzx   eax, al
0x1b9d1:    movsxd  rax, dword ptr [r12 + rax*4]
0x1b9d5:    add     rax, r12
0x1b9d8:    jmp     rax
rax
->->->                                indirect JMP
0x13ed7:    mov     ecx, dword ptr [rbp + 0x68]
0x13eda:    test    ecx, ecx
0x13edc:    jle     0x14cab
0x13ee2:    mov     rdx, qword ptr [rbp + 0x38]
0x13ee6:    xor     eax, eax
0x13ee8:    jmp     0x13efb
0x13efb
0x13eea:    nop     word ptr [rax + rax]
0x13ef0:    add     eax, 1
0x13ef3:    add     rdx, 0x50
0x13ef7:    cmp     eax, ecx
0x13ef9:    je      0x13f00
0x13efb:    cmp     qword ptr [rdx], rbx
0x13efe:    jne     0x13ef0
0x13f00:    mov     dword ptr [rbp + 0x28], eax
0x13f03:    mov     rdi, rbp
0x13f06:    xor     eax, eax
0x13f08:    call    0xf110
0x13f0d:    mov     dword ptr [rbp + 0x2c], eax
0x13f10:    jmp     0x13f40
0x13f40
0x13f12:    nop     word ptr [rax + rax]
0x13f18:    sub     r15d, 9

```

```

0x13f1c:    cmp     dword ptr [rbp + 0x64], 0x4f
0x13f20:    mov     dword ptr [rbp + 0x80], r15d
0x13f27:    jg      0x13f40
0x13f29:    cmp     r15d, 5
0x13f2d:    jne     0x13f40
0x13f2f:    mov     dword ptr [rbp + 0x80], 0
0x13f39:    nop     dword ptr [rax]
0x13f40:    cmp     byte ptr [rbp + 0x6d], 0
0x13f44:    je      0x13f50
0x13f46:    mov     rdi, rbp
0x13f49:    xor     eax, eax
0x13f4b:    call    0x10d70
0x13f50:    mov     rax, qword ptr [rsp + 0x38]
0x13f55:    sub     rax, qword ptr fs:[0x28]
0x13f5e:    jne     0x14ce9
0x13f64:    add     rsp, 0x48
0x13f68:    xor     eax, eax
0x13f6a:    pop     rbx
0x13f6b:    pop     rbp
0x13f6c:    pop     r12
0x13f6e:    pop     r13
0x13f70:    pop     r14
0x13f72:    pop     r15
0x13f74:    ret
0x13f75:    nop     dword ptr [rax]
0x13f78:    movsxd  rax, dword ptr [rdi + 0x28]
0x13f7c:    mov     esi, 0x2e
0x13f81:    lea     rax, [rax + rax*4]
0x13f85:    shl     rax, 4
0x13f89:    add     rax, qword ptr [rdi + 0x38]
0x13f8d:    mov     r12, qword ptr [rax]
0x13f90:    mov     rdi, r12
0x13f93:    call    0x4a20
0x13f98:    mov     rbx, rax
0x13f9b:    test    rax, rax
0x13f9e:    je      0x13f50
0x13fa0:    cmp     r12, rax
0x13fa3:    je      0x13f50
0x13fa5:    mov     rdi, rax
0x13fa8:    call    0x49b0
0x13fad:    lea     rdi, [rax + 2]
0x13fb1:    call    0x4c20
0x13fb6:    mov     r8, rbx
0x13fb9:    mov     esi, 1
0x13fbe:    xor     ebx, ebx
0x13fc0:    mov     rdi, rax
0x13fc3:    mov     r13, rax
0x13fc6:    xor     eax, eax

```

```

0x13fc8:      mov     rdx, -1
0x13fcf:      lea     rcx, [rip + 0xd3be]
0x13fd6:      call    0x4f10
0x13fdb:      mov     eax, dword ptr [rbp + 0x68]
0x13fde:      test    eax, eax
0x13fe0:      jle     0x14132
0x13fe6:      nop     word ptr cs:[rax + rax]
0x13ff0:      mov     rax, qword ptr [rbp + 0x38]
0x13ff4:      lea     r12, [rbx + rbx*4]
0x13ff8:      shl     r12, 4
0x13ffc:      add     rax, r12
0x13fff:      cmp     byte ptr [rax + 0x48], 0
0x14003:      je      0x14048
0x14005:      cmp     dword ptr [rip + 0x17618], 1
0x1400c:      mov     rsi, qword ptr [rax]
0x1400f:      mov     rdi, r13
0x14012:      sbb     edx, edx
0x14014:      and     edx, 4
0x14017:      add     edx, 1
0x1401a:      call    0x4a40
0x1401f:      test    eax, eax
0x14021:      jne     0x14048
0x14023:      add     r12, qword ptr [rbp + 0x38]
0x14027:      movzx   eax, byte ptr [r12 + 0x49]
0x1402d:      cmp     r15d, 0x32
0x14031:      je      0x149b3
0x14037:      test    al, al
0x14039:      je      0x14048
0x1403b:      mov     byte ptr [r12 + 0x49], 0
0x14041:      sub     dword ptr [rbp + 0x70], 1
0x14045:      nop     dword ptr [rax]
0x14048:      add     rbx, 1
0x1404c:      cmp     dword ptr [rbp + 0x68], ebx
0x1404f:      jg      0x13ff0
0x14051:      jmp     0x14132
0x14132:
0x14056:      nop     word ptr cs:[rax + rax]
0x14060:      xor     eax, eax
0x14062:      mov     rdi, rcx
0x14065:      call    0x13ca0
0x1406a:      cmp     qword ptr [rax], 0
0x1406e:      mov     r14, rax
0x14071:      mov     r13, rax
0x14074:      je      0x1412a
0x1407a:      nop     word ptr [rax + rax]
0x14080:      mov     edx, dword ptr [rbp + 0x68]
0x14083:      xor     ebx, ebx
0x14085:      test    edx, edx

```

```

0x14087:    jle     0x140f9
0x14089:    nop     dword ptr [rax]
0x14090:    mov     rax, qword ptr [rbp + 0x38]
0x14094:    lea     r12, [rbx + rbx*4]
0x14098:    shl     r12, 4
0x1409c:    add     rax, r12
0x1409f:    cmp     byte ptr [rax + 0x48], 0
0x140a3:    je      0x140f0
0x140a5:    cmp     dword ptr [rip + 0x17578], 1
0x140ac:    mov     rsi, qword ptr [rax]
0x140af:    sbb     edx, edx
0x140b1:    mov     rdi, qword ptr [r13]
0x140b5:    and     edx, 4
0x140b8:    add     edx, 1
0x140bb:    call    0x4a40
0x140c0:    test    eax, eax
0x140c2:    jne     0x140f0
0x140c4:    add     r12, qword ptr [rbp + 0x38]
0x140c8:    movzx   eax, byte ptr [r12 + 0x49]
0x140ce:    cmp     r15d, 0x24
0x140d2:    je      0x14840
0x140d8:    test    al, al
0x140da:    je      0x140f0
0x140dc:    mov     byte ptr [r12 + 0x49], 0
0x140e2:    sub     dword ptr [rbp + 0x70], 1
0x140e6:    nop     word ptr cs:[rax + rax]
0x140f0:    add     rbx, 1
0x140f4:    cmp     dword ptr [rbp + 0x68], ebx
0x140f7:    jg      0x14090
0x140f9:    add     r13, 8
0x140fd:    cmp     qword ptr [r13], 0
0x14102:    jne     0x14080
0x14108:    mov     rdi, qword ptr [r14]
0x1410b:    test    rdi, rdi
0x1410e:    je      0x1412a
0x14110:    mov     rbx, r14
0x14113:    nop     dword ptr [rax + rax]
0x14118:    call    0x18d10
0x1411d:    mov     rdi, qword ptr [rbx + 8]
0x14121:    add     rbx, 8
0x14125:    test    rdi, rdi
0x14128:    jne     0x14118
0x1412a:    mov     rdi, r14
0x1412d:    call    0x18d10
0x14132:    cmp     byte ptr [rbp + 0x6d], 0
0x14136:    je      0x13f50
0x1413c:    mov     rdi, rbp
0x1413f:    xor     eax, eax

```

```

0x14141:      call    0x10d70
0x14146:      cmp     byte ptr [rbp + 0x6d], 0
0x1414a:      je      0x13f50
0x14150:      mov     rdi, rbp
0x14153:      xor     eax, eax
0x14155:      call    0xec50
0x1415a:      jmp     0x13f50
0x13f50
0x1415f:      mov     rax, qword ptr [rdi + 0x18]
0x14163:      cmp     byte ptr [rax], 0x2f
0x14166:      je      0x149a4
0x1416c:      mov     esi, dword ptr [rbp + 0x28]
0x1416f:      test    esi, esi
0x14171:      je      0x14187
0x14173:      mov     dword ptr [rbp + 0x28], 0
0x1417a:      mov     rdi, rbp
0x1417d:      xor     eax, eax
0x1417f:      call    0xf110
0x14184:      mov     dword ptr [rbp + 0x2c], eax
0x14187:      mov     rax, qword ptr [rsp + 0x38]
0x1418c:      sub     rax, qword ptr fs:[0x28]
0x14195:      jne     0x14ce9
0x1419b:      add     rsp, 0x48
0x1419f:      mov     rsi, r12
0x141a2:      mov     rdi, rbp
0x141a5:      xor     eax, eax
0x141a7:      pop     rbx
0x141a8:      pop     rbp
0x141a9:      pop     r12
0x141ab:      pop     r13
0x141ad:      pop     r14
0x141af:      pop     r15
0x141b1:      jmp     0x162b0
0x162b0
0x141b6:      mov     edx, dword ptr [rdi + 0x68]
0x141b9:      test    edx, edx
0x141bb:      jle     0x141ec
0x141bd:      sub     edx, 1
0x141c0:      mov     rcx, qword ptr [rdi + 0x38]
0x141c4:      lea     rdx, [rdx + rdx*4]
0x141c8:      shl     rdx, 4
0x141cc:      lea     rax, [rcx + 0x49]
0x141d0:      lea     rdx, [rcx + rdx + 0x99]
0x141d8:      nop     dword ptr [rax + rax]
0x141e0:      mov     byte ptr [rax], 0
0x141e3:      add     rax, 0x50
0x141e7:      cmp     rdx, rax
0x141ea:      jne     0x141e0

```

```

0x141ec:      cmp     byte ptr [rbp + 0x6d], 0
0x141f0:      mov     dword ptr [rbp + 0x70], 0
0x141f7:      je      0x13f50
0x141fd:      jmp     0x1413c
0x1413c
0x14202:      movzx   edx, byte ptr [rdi + 0x6d]
0x14206:      mov     ebx, r8d
0x14209:      xor     r13d, r13d
0x1420c:      xor     r12d, r12d
0x1420f:      test    r8d, r8d
0x14212:      jne     0x14249
0x14214:      jmp     0x1488f
0x1488f
0x14219:      nop     dword ptr [rax]
0x14220:      test    r12d, r12d
0x14223:      jne     0x14240
0x14225:      mov     r12d, 1
0x1422b:      test    dl, dl
0x1422d:      je      0x14240
0x1422f:      mov     rdi, rbp
0x14232:      call    0x10760
0x14237:      movzx   edx, byte ptr [rbp + 0x6d]
0x1423b:      nop     dword ptr [rax + rax]
0x14240:      sub     ebx, 1
0x14243:      je      0x14870
0x14249:      mov     esi, dword ptr [rbp + 0x28]
0x1424c:      test    esi, esi
0x1424e:      je      0x14870
0x14254:      lea     eax, [rsi - 1]
0x14257:      mov     dword ptr [rbp + 0x28], eax
0x1425a:      cmp     esi, dword ptr [rbp + 0x2c]
0x1425d:      jne     0x14220
0x1425f:      sub     esi, dword ptr [rbp + 0x88]
0x14265:      mov     eax, 0
0x1426a:      mov     r13d, 1
0x14270:      cmovs   esi, eax
0x14273:      mov     dword ptr [rbp + 0x2c], esi
0x14276:      jmp     0x14240
0x14240
0x14278:      mov     rsi, r12
0x1427b:      xor     eax, eax
0x1427d:      call    0x15bb0
0x14282:      jmp     0x13f50
0x13f50
0x14287:      mov     rsi, r12
0x1428a:      xor     eax, eax
0x1428c:      call    0x15910
0x14291:      jmp     0x13f50

```

```

0x13f50
0x14296:    mov     edx, dword ptr [rcx]
0x14298:    mov     rsi, r12
0x1429b:    xor     eax, eax
0x1429d:    call    0x167a0
0x142a2:    jmp     0x13f50
0x13f50
0x142a7:    mov     edx, dword ptr [rcx]
0x142a9:    mov     rsi, r12
0x142ac:    xor     eax, eax
0x142ae:    call    0x12fd0
0x142b3:    jmp     0x13f50
0x13f50
0x142b8:    mov     rdi, qword ptr [rdi + 0x48]
0x142bc:    call    0x19120
0x142c1:    mov     qword ptr [rbp + 0x48], 0
0x142c9:    jmp     0x13f50
0x13f50
0x142ce:    mov     eax, dword ptr [rdi + 0x30]
0x142d1:    mov     edx, dword ptr [rdi + 0x64]
0x142d4:    mov     esi, 4
0x142d9:    add     r8d, dword ptr [rdi + 0x74]
0x142dd:    add     eax, 1
0x142e0:    cmp     edx, 0x27
0x142e3:    jle     0x14300
0x142e5:    movsxd  rcx, dword ptr [rdi + 0x80]
0x142ec:    cmp     edx, 0x49
0x142ef:    jg      0x142f6
0x142f1:    cmp     ecx, 5
0x142f4:    je      0x14300
0x142f6:    lea     rsi, [rip + 0xd6e3]
0x142fd:    mov     esi, dword ptr [rsi + rcx*4]
0x14300:    sub     edx, esi
0x14302:    sub     eax, edx
0x14304:    mov     edx, 0
0x14309:    cmovs   eax, edx
0x1430c:    cmp     eax, r8d
0x1430f:    cmovg   eax, r8d
0x14313:    mov     dword ptr [rbp + 0x74], eax
0x14316:    mov     rdi, rbp
0x14319:    xor     eax, eax
0x1431b:    call    0x10ec0
0x14320:    xor     eax, eax
0x14322:    mov     rdi, rbp
0x14325:    call    0x10160
0x1432a:    cmp     byte ptr [rbp + 0x6d], 0
0x1432e:    je      0x1433a
0x14330:    mov     rdi, rbp

```



```

0x14333:      xor     eax, eax
0x14335:      call   0xec50
0x1433a:      mov     rdi, rbp
0x1433d:      xor     eax, eax
0x1433f:      call   0x10240
0x14344:      jmp     0x13f40
0x13f40
0x14349:      sub     dword ptr [rdi + 0x74], r8d
0x1434d:      mov     eax, dword ptr [rdi + 0x74]
0x14350:      xor     edx, edx
0x14352:      test    eax, eax
0x14354:      cmovs   eax, edx
0x14357:      jmp     0x14313
0x14313
0x14359:      xor     esi, esi
0x1435b:      mov     rdi, rcx
0x1435e:      mov     edx, 0xa
0x14363:      call   0x4ba0
0x14368:      mov     rsi, r12
0x1436b:      mov     rdi, rbp
0x1436e:      movsxd  rdx, eax
0x14371:      xor     eax, eax
0x14373:      call   0x137c0
0x14378:      jmp     0x13f50
0x13f50
0x1437d:      mov     rsi, r12
0x14380:      xor     eax, eax
0x14382:      call   0x12ca0
0x14387:      jmp     0x13f50
0x13f50
0x1438c:      mov     eax, dword ptr [rdi + 0x60]
0x1438f:      mov     edx, dword ptr [r12 + 0x60]
0x14394:      mov     dword ptr [rdi + 0x60], edx
0x14397:      mov     edx, dword ptr [r12 + 0x64]
0x1439c:      mov     dword ptr [r12 + 0x60], eax
0x143a1:      mov     eax, dword ptr [rdi + 0x64]
0x143a4:      mov     dword ptr [rdi + 0x64], edx
0x143a7:      mov     edx, dword ptr [r12 + 0x58]
0x143ac:      mov     dword ptr [r12 + 0x64], eax
0x143b1:      mov     eax, dword ptr [rdi + 0x58]
0x143b4:      mov     dword ptr [rdi + 0x58], edx
0x143b7:      mov     edx, dword ptr [r12 + 0x5c]
0x143bc:      mov     dword ptr [r12 + 0x58], eax
0x143c1:      mov     eax, dword ptr [rdi + 0x5c]
0x143c4:      mov     dword ptr [rdi + 0x5c], edx
0x143c7:      mov     dword ptr [r12 + 0x5c], eax
0x143cc:      movsxd  rcx, dword ptr [rdi + 0x60]
0x143d0:      movsxd  rdx, dword ptr [rdi + 0x5c]

```

```

0x143d4:      movsxd  rsi, dword ptr [rdi + 0x58]
0x143d8:      movsxd  r8, dword ptr [rdi + 0x64]
0x143dc:      call   0xf590
0x143e1:      movsxd  rcx, dword ptr [r12 + 0x60]
0x143e6:      movsxd  rdx, dword ptr [r12 + 0x5c]
0x143eb:      mov     rdi, r12
0x143ee:      movsxd  rsi, dword ptr [r12 + 0x58]
0x143f3:      movsxd  r8, dword ptr [r12 + 0x64]
0x143f8:      call   0xf590
0x143fd:      jmp     0x13f50
0x13f50
0x14402:      mov     rsi, r12
0x14405:      xor     eax, eax
0x14407:      call   0x14e20
0x1440c:      jmp     0x13f50
0x13f50
0x14411:      mov     eax, dword ptr [rdi + 0x84]
0x14417:      add     eax, 1
0x1441a:      movsxd  r15, eax
0x1441d:      cdq
0x1441e:      imul    r15, r15, 0x38e38e39
0x14425:      sar     r15, 0x21
0x14429:      sub     r15d, edx
0x1442c:      lea     edx, [r15 + r15*8]
0x14430:      sub     eax, edx
0x14432:      mov     r15d, eax
0x14435:      jmp     0x13e84
0x13e84
0x1443a:      mov     eax, dword ptr [rdi + 0x80]
0x14440:      add     eax, 1
0x14443:      movsxd  r15, eax
0x14446:      cdq
0x14447:      imul    r15, r15, 0x2aaaaaab
0x1444e:      shr     r15, 0x20
0x14452:      sub     r15d, edx
0x14455:      lea     edx, [r15 + r15*2]
0x14459:      add     edx, edx
0x1445b:      sub     eax, edx
0x1445d:      mov     r15d, eax
0x14460:      jmp     0x13f1c
0x13f1c
0x14465:      mov     edx, dword ptr [rdi + 0x68]
0x14468:      mov     dword ptr [rdi + 0x70], 0
0x1446f:      test    edx, edx
0x14471:      jle     0x14132
0x14477:      sub     edx, 1
0x1447a:      mov     rcx, qword ptr [rdi + 0x38]
0x1447e:      lea     rdx, [rdx + rdx*4]

```

```

0x14482:      shl     rdx, 4
0x14486:      lea     rax, [rcx + 0x49]
0x1448a:      lea     rcx, [rcx + rdx + 0x99]
0x14492:      nop     word ptr [rax + rax]
0x14498:      cmp     byte ptr [rax - 1], 0
0x1449c:      je      0x144ac
0x1449e:      cmp     byte ptr [rax], 0
0x144a1:      sete    dl
0x144a4:      mov     byte ptr [rax], dl
0x144a6:      movzx   edx, dl
0x144a9:      add     dword ptr [rbp + 0x70], edx
0x144ac:      add     rax, 0x50
0x144b0:      cmp     rcx, rax
0x144b3:      jne     0x14498
0x144b5:      jmp     0x14132
0x14132
0x144ba:      mov     rsi, rcx
0x144bd:      xor     eax, eax
0x144bf:      call    0x161d0
0x144c4:      jmp     0x13f50
0x13f50
0x144c9:      mov     rdx, rcx
0x144cc:      mov     rsi, r12
0x144cf:      xor     eax, eax
0x144d1:      call    0x165f0
0x144d6:      jmp     0x13f50
0x13f50
0x144db:      mov     rcx, qword ptr [rcx + 8]
0x144df:      mov     rdi, rcx
0x144e2:      mov     qword ptr [rsp + 8], rcx
0x144e7:      call    0x49b0
0x144ec:      mov     rcx, qword ptr [rsp + 8]
0x144f1:      mov     qword ptr [rsp + 0x10], rax
0x144f6:      mov     r14, rax
0x144f9:      mov     eax, dword ptr [r13]
0x144fd:      cmp     eax, 2
0x14500:      je      0x14bb4
0x14506:      cmp     eax, 3
0x14509:      je      0x14aea
0x1450f:      cmp     eax, 1
0x14512:      je      0x148db
0x14518:      mov     rax, qword ptr [rbp + 0x40]
0x1451c:      mov     qword ptr [r13 + 0x10], rax
0x14520:      jmp     0x13f50
0x13f50
0x14525:      mov     rdi, qword ptr [rcx + 8]
0x14529:      mov     qword ptr [rsp + 8], rdi
0x1452e:      call    0x49b0

```

```

0x14533:      mov     rdi, qword ptr [rsp + 8]
0x14538:      mov     qword ptr [rsp + 0x10], rax
0x1453d:      mov     r14, rax
0x14540:      mov     eax, dword ptr [r13]
0x14544:      cmp     eax, 2
0x14547:      je      0x14a96
0x1454d:      cmp     eax, 3
0x14550:      je      0x149de
0x14556:      cmp     eax, 1
0x14559:      je      0x148db
0x1455f:      mov     rax, qword ptr [rbp + 0x40]
0x14563:      mov     qword ptr [rsp + 0x10], rax
0x14568:      mov     rax, qword ptr [rsp + 0x10]
0x1456d:      mov     qword ptr [r13 + 0x10], rax
0x14571:      jmp     0x13f50
0x13f50
0x14576:      mov     edi, 0x10
0x1457b:      call    0x19100
0x14580:      lea     rsi, [rsp + 0x20]
0x14585:      mov     qword ptr [rsp + 0x28], 0
0x1458e:      mov     qword ptr [rbp + 0x48], rax
0x14592:      mov     rdi, rax
0x14595:      mov     eax, dword ptr [rbp + 0x28]
0x14598:      mov     dword ptr [rsp + 0x20], eax
0x1459c:      call    0x19150
0x145a1:      mov     qword ptr [rbp + 0x40], 0
0x145a9:      mov     byte ptr [rbp + 0x6e], 0
0x145ad:      jmp     0x13f50
0x13f50
0x145b2:      xor     esi, esi
0x145b4:      mov     edx, 0xa
0x145b9:      mov     rdi, rcx
0x145bc:      call    0x4ba0
0x145c1:      test    eax, eax
0x145c3:      jle     0x13f50
0x145c9:      mov     ecx, dword ptr [rbp + 0x60]
0x145cc:      lea     edx, [rcx - 1]
0x145cf:      cmp     edx, eax
0x145d1:      jle     0x13f50
0x145d7:      mov     dword ptr [r12 + 0x88], eax
0x145df:      mov     dword ptr [rbp + 0x88], eax
0x145e5:      jmp     0x13f50
0x13f50
0x145ea:      mov     eax, dword ptr [rdi + 0x68]
0x145ed:      sub     eax, 1
0x145f0:      cmp     dword ptr [rdi + 0x28], eax
0x145f3:      je      0x13f50
0x145f9:      mov     dword ptr [rdi + 0x28], eax

```

```

0x145fc:      mov     rdi, rbp
0x145ff:      xor     eax, eax
0x14601:      call    0xf110
0x14606:      mov     dword ptr [rbp + 0x2c], eax
0x14609:      jmp     0x14132
0x14132:
0x1460e:      mov     ecx, dword ptr [rdi + 0x28]
0x14611:      test    ecx, ecx
0x14613:      je      0x13f50
0x14619:      mov     dword ptr [rdi + 0x28], 0
0x14620:      jmp     0x145fc
0x145fc:
0x14622:      mov     r9d, dword ptr [rdi + 0x68]
0x14626:      mov     r10d, dword ptr [rdi + 0x28]
0x1462a:      lea     eax, [r9 - 1]
0x1462e:      cmp     r10d, eax
0x14631:      je      0x13f50
0x14637:      mov     r8d, dword ptr [rdi + 0x60]
0x1463b:      mov     ecx, eax
0x1463d:      sub     ecx, dword ptr [rdi + 0x2c]
0x14640:      lea     edx, [r8 - 2]
0x14644:      cmp     ecx, edx
0x14646:      jl      0x14656
0x14648:      mov     ecx, eax
0x1464a:      sub     ecx, r10d
0x1464d:      cmp     edx, ecx
0x1464f:      jg      0x14656
0x14651:      lea     eax, [r10 + r8 - 2]
0x14656:      mov     dword ptr [rbp + 0x28], eax
0x14659:      mov     rdi, rbp
0x1465c:      xor     eax, eax
0x1465e:      sub     r8d, 1
0x14662:      call    0xf110
0x14667:      mov     dword ptr [rbp + 0x2c], eax
0x1466a:      movzx   eax, byte ptr [rbp + 0x6d]
0x1466e:      cmp     r9d, r8d
0x14671:      jge     0x148ce
0x14677:      test    al, al
0x14679:      je      0x13f50
0x1467f:      mov     esi, r10d
0x14682:      mov     rdi, rbp
0x14685:      call    0x10760
0x1468a:      cmp     byte ptr [rbp + 0x6d], 0
0x1468e:      mov     esi, dword ptr [rbp + 0x28]
0x14691:      je      0x13f50
0x14697:      mov     rdi, rbp
0x1469a:      call    0x10760
0x1469f:      jmp     0x14146

```

```

0x14146
0x146a4:      mov     r8d, dword ptr [rdi + 0x28]
0x146a8:      test    r8d, r8d
0x146ab:      je       0x13f50
0x146b1:      mov     r9d, dword ptr [rdi + 0x60]
0x146b5:      lea     eax, [r9 - 2]
0x146b9:      cmp     r8d, eax
0x146bc:      jge     0x149ca
0x146c2:      mov     dword ptr [rdi + 0x28], 0
0x146c9:      xor     eax, eax
0x146cb:      call    0xf110
0x146d0:      mov     dword ptr [rbp + 0x2c], eax
0x146d3:      sub     r9d, 1
0x146d7:      movzx   eax, byte ptr [rbp + 0x6d]
0x146db:      cmp     r9d, dword ptr [rbp + 0x68]
0x146df:      jle     0x148ce
0x146e5:      mov     esi, r8d
0x146e8:      test    al, al
0x146ea:      je       0x13f50
0x146f0:      jmp     0x14682
0x14682
0x146f2:      movzx   edi, byte ptr [rdi + 0x6d]
0x146f6:      mov     ebx, r8d
0x146f9:      xor     r13d, r13d
0x146fc:      xor     r12d, r12d
0x146ff:      test    r8d, r8d
0x14702:      jne     0x14739
0x14704:      jmp     0x148c0
0x148c0
0x14709:      nop     dword ptr [rax]
0x14710:      test    r12d, r12d
0x14713:      jne     0x14730
0x14715:      mov     r12d, 1
0x1471b:      test    dil, dil
0x1471e:      je       0x14730
0x14720:      mov     rdi, rbp
0x14723:      mov     esi, r9d
0x14726:      call    0x10760
0x1472b:      movzx   edi, byte ptr [rbp + 0x6d]
0x1472f:      nop
0x14730:      sub     ebx, 1
0x14733:      je       0x148a0
0x14739:      mov     eax, dword ptr [rbp + 0x68]
0x1473c:      mov     r9d, dword ptr [rbp + 0x28]
0x14740:      sub     eax, 1
0x14743:      cmp     r9d, eax
0x14746:      jge     0x148a0
0x1474c:      mov     esi, dword ptr [rbp + 0x2c]

```

```

0x1474f:    mov     ecx, dword ptr [rbp + 0x60]
0x14752:    lea     edx, [r9 + 1]
0x14756:    mov     dword ptr [rbp + 0x28], edx
0x14759:    sub     ecx, 2
0x1475c:    sub     edx, esi
0x1475e:    cmp     edx, ecx
0x14760:    jl      0x14710
0x14762:    sub     eax, ecx
0x14764:    add     esi, dword ptr [rbp + 0x88]
0x1476a:    mov     r13d, 1
0x14770:    add     eax, 1
0x14773:    cmp     esi, eax
0x14775:    cmovg   esi, eax
0x14778:    mov     dword ptr [rbp + 0x2c], esi
0x1477b:    jmp     0x14730
0x14730:
0x1477d:    mov     rax, qword ptr [rdi + 0x18]
0x14781:    mov     esi, dword ptr [rdi + 0x28]
0x14784:    cmp     byte ptr [rax], 0x2f
0x14787:    je      0x1497c
0x1478d:    test    esi, esi
0x1478f:    je      0x147d0
0x14791:    movsxd  rax, esi
0x14794:    lea     rax, [rax + rax*4]
0x14798:    shl     rax, 4
0x1479c:    add     rax, qword ptr [rbp + 0x38]
0x147a0:    movzx   edx, byte ptr [rax + 0x49]
0x147a4:    test    dl, dl
0x147a6:    sete    byte ptr [rax + 0x49]
0x147aa:    cmp     dl, 1
0x147ad:    sbb     eax, eax
0x147af:    and     eax, 2
0x147b2:    sub     eax, 1
0x147b5:    add     dword ptr [rbp + 0x70], eax
0x147b8:    cmp     byte ptr [rbp + 0x6d], 0
0x147bc:    je      0x147d0
0x147be:    mov     rdi, rbp
0x147c1:    mov     dword ptr [rsp + 8], r8d
0x147c6:    call    0x10760
0x147cb:    mov     r8d, dword ptr [rsp + 8]
0x147d0:    xor     ecx, ecx
0x147d2:    mov     rdx, r12
0x147d5:    mov     esi, 0x1c
0x147da:    mov     rdi, rbp
0x147dd:    call    0x13e30
0x147e2:    jmp     0x13f50
0x13f50:
0x147e7:    mov     rsi, r12

```

```

0x147ea:    xor     eax, eax
0x147ec:    call   0x15400
0x147f1:    jmp     0x13f50
0x13f50:
0x147f6:    mov     edx, dword ptr [rdi + 0x68]
0x147f9:    test    edx, edx
0x147fb:    jle     0x14132
0x14801:    sub     edx, 1
0x14804:    mov     rcx, qword ptr [rdi + 0x38]
0x14808:    lea     rdx, [rdx + rdx*4]
0x1480c:    shl     rdx, 4
0x14810:    lea     rax, [rcx + 0x49]
0x14814:    lea     rdx, [rcx + rdx + 0x99]
0x1481c:    nop     dword ptr [rax]
0x14820:    cmp     byte ptr [rax], 0
0x14823:    jne     0x14832
0x14825:    cmp     byte ptr [rax - 1], 0
0x14829:    je      0x14832
0x1482b:    mov     byte ptr [rax], 1
0x1482e:    add     dword ptr [rbp + 0x70], 1
0x14832:    add     rax, 0x50
0x14836:    cmp     rax, rdx
0x14839:    jne     0x14820
0x1483b:    jmp     0x14132
0x14132:
0x14840:    test    al, al
0x14842:    jne     0x140f0
0x14848:    mov     byte ptr [r12 + 0x49], 1
0x1484e:    add     dword ptr [rbp + 0x70], 1
0x14852:    jmp     0x140f0
0x140f0:
0x14857:    lea     rdi, [rip + 0xcb3a]
0x1485e:    call   0x8b00
0x14863:    jmp     0x13f50
0x13f50:
0x14868:    nop     dword ptr [rax + rax]
0x14870:    test    r13d, r13d
0x14873:    je      0x14931
0x14879:    test    dl, dl
0x1487b:    je      0x13f50
0x14881:    mov     rdi, rbp
0x14884:    xor     eax, eax
0x14886:    call   0x10d70
0x1488b:    movzx   edx, byte ptr [rbp + 0x6d]
0x1488f:    test    dl, dl
0x14891:    jne     0x14150
0x14897:    jmp     0x13f50
0x13f50:

```



```

0x1489c:      nop      dword ptr [rax]
0x148a0:      test     r13d, r13d
0x148a3:      je       0x14956
0x148a9:      test     dil, dil
0x148ac:      je       0x13f50
0x148b2:      mov      rdi, rbp
0x148b5:      xor      eax, eax
0x148b7:      call     0x10d70
0x148bc:      movzx     edi, byte ptr [rbp + 0x6d]
0x148c0:      test     dil, dil
0x148c3:      jne      0x14150
0x148c9:      jmp      0x13f50
0x13f50
0x148ce:      test     al, al
0x148d0:      je       0x13f50
0x148d6:      jmp      0x1413c
0x1413c
0x148db:      mov      rdi, qword ptr [rbp + 0x48]
0x148df:      lea      r14, [rsp + 0x20]
0x148e4:      mov      rsi, r14
0x148e7:      call     0x191a0
0x148ec:      mov      edx, dword ptr [rsp + 0x20]
0x148f0:      mov      rax, qword ptr [rbp + 0x40]
0x148f4:      cmp      rax, qword ptr [rsp + 0x10]
0x148f9:      jne      0x1451c
0x148ff:      mov      r8d, dword ptr [rbp + 0x28]
0x14903:      cmp      r8d, edx
0x14906:      jl       0x14cb2
0x1490c:      jg       0x14ccf
0x14912:      mov      rdi, qword ptr [rbp + 0x48]
0x14916:      mov      edx, 1
0x1491b:      mov      rsi, r14
0x1491e:      call     0x19200
0x14923:      mov      rax, qword ptr [rsp + 0x28]
0x14928:      mov      qword ptr [rbp + 0x40], rax
0x1492c:      jmp      0x1451c
0x1451c
0x14931:      test     r12d, r12d
0x14934:      je       0x1488f
0x1493a:      mov      esi, dword ptr [rbp + 0x28]
0x1493d:      test     dl, dl
0x1493f:      je       0x13f50
0x14945:      mov      rdi, rbp
0x14948:      call     0x10760
0x1494d:      movzx     edx, byte ptr [rbp + 0x6d]
0x14951:      jmp      0x1488f
0x1488f
0x14956:      test     r12d, r12d

```

```

0x14959:      je      0x148c0
0x1495f:      mov     esi, dword ptr [rbp + 0x28]
0x14962:      test    dil, dil
0x14965:      je      0x13f50
0x1496b:      mov     rdi, rbp
0x1496e:      call    0x10760
0x14973:      movzx    edi, byte ptr [rbp + 0x6d]
0x14977:      jmp     0x148c0
0x148c0
0x1497c:      cmp     byte ptr [rax + 1], 0
0x14980:      je      0x14791
0x14986:      jmp     0x1478d
0x1478d
0x1498b:      movsxd   rsi, eax
0x1498e:      lea     rcx, [rip - 0x6315]
0x14995:      mov     edx, 0x50
0x1499a:      call    0x48e0
0x1499f:      jmp     0x13ed7
0x13ed7
0x149a4:      cmp     byte ptr [rax + 1], 0
0x149a8:      je      0x13f50
0x149ae:      jmp     0x1416c
0x1416c
0x149b3:      test    al, al
0x149b5:      jne     0x14048
0x149bb:      mov     byte ptr [r12 + 0x49], 1
0x149c1:      add     dword ptr [rbp + 0x70], 1
0x149c5:      jmp     0x14048
0x14048
0x149ca:      mov     edx, r8d
0x149cd:      sub     edx, eax
0x149cf:      xor     eax, eax
0x149d1:      mov     dword ptr [rdi + 0x28], edx
0x149d4:      call    0xf110
0x149d9:      jmp     0x146d0
0x146d0
0x149de:      mov     eax, dword ptr [rbp + 0x28]
0x149e1:      mov     rdi, qword ptr [rbp + 0x48]
0x149e5:      lea     rsi, [rsp + 0x20]
0x149ea:      mov     qword ptr [rsp + 0x28], r14
0x149ef:      mov     dword ptr [rsp + 0x20], eax
0x149f3:      call    0x19150
0x149f8:      mov     rax, qword ptr [r13 + 8]
0x149fc:      mov     ebx, dword ptr [rbp + 0x28]
0x149ff:      mov     qword ptr [rsp + 8], rax
0x14a04:      test    ebx, ebx
0x14a06:      js      0x14a80
0x14a08:      movsxd   rax, ebx

```

```

0x14a0b:    lea     r8, [rax + rax*4]
0x14a0f:    shl     r8, 4
0x14a13:    add     r8, qword ptr [rbp + 0x38]
0x14a17:    mov     r15, r8
0x14a1a:    jmp     0x14a2c
0x14a2c:
0x14a1c:    nop     dword ptr [rax]
0x14a20:    sub     ebx, 1
0x14a23:    sub     r15, 0x50
0x14a27:    cmp     ebx, -1
0x14a2a:    je      0x14a80
0x14a2c:    mov     rsi, qword ptr [r15]
0x14a2f:    mov     rdi, qword ptr [rsp + 8]
0x14a34:    mov     rdx, r14
0x14a37:    call    0x4c30
0x14a3c:    test    eax, eax
0x14a3e:    jne     0x14a20
0x14a40:    cmp     byte ptr [rbp + 0x6e], 0
0x14a44:    mov     qword ptr [rbp + 0x40], r14
0x14a48:    je      0x14c8d
0x14a4e:    mov     dword ptr [rbp + 0x28], ebx
0x14a51:    xor     eax, eax
0x14a53:    mov     rdi, rbp
0x14a56:    call    0xf110
0x14a5b:    cmp     byte ptr [rbp + 0x6d], 0
0x14a5f:    mov     dword ptr [rbp + 0x2c], eax
0x14a62:    je      0x14a77
0x14a64:    xor     eax, eax
0x14a66:    mov     rdi, rbp
0x14a69:    call    0x10d70
0x14a6e:    mov     rax, qword ptr [rbp + 0x40]
0x14a72:    mov     qword ptr [rsp + 0x10], rax
0x14a77:    mov     byte ptr [rbp + 0x6e], 0
0x14a7b:    jmp     0x14568
0x14568:
0x14a80:    mov     rax, qword ptr [rbp + 0x40]
0x14a84:    mov     dword ptr [r13], 0xffffffff
0x14a8c:    mov     qword ptr [rsp + 0x10], rax
0x14a91:    jmp     0x14568
0x14568:
0x14a96:    cmp     byte ptr [rbp + 0x6e], 0
0x14a9a:    je      0x14c0e
0x14aa0:    mov     eax, dword ptr [rbp + 0x68]
0x14aa3:    lea     ebx, [rax - 1]
0x14aa6:    test    ebx, ebx
0x14aa8:    js      0x14a80
0x14aaa:    movsxd  rax, ebx
0x14aad:    lea     r15, [rax + rax*4]

```

```

0x14ab1:    shl     r15, 4
0x14ab5:    add     r15, qword ptr [rbp + 0x38]
0x14ab9:    jmp     0x14ad1
0x14ad1:
0x14abb:    nop     dword ptr [rax + rax]
0x14ac0:    sub     ebx, 1
0x14ac3:    sub     r15, 0x50
0x14ac7:    mov     rdi, qword ptr [rsp + 8]
0x14acc:    cmp     ebx, -1
0x14acf:    je      0x14a80
0x14ad1:    mov     rsi, qword ptr [r15]
0x14ad4:    mov     rdx, r14
0x14ad7:    mov     qword ptr [rsp + 8], rdi
0x14adc:    call    0x4c30
0x14ae1:    test    eax, eax
0x14ae3:    jne     0x14ac0
0x14ae5:    jmp     0x14a40
0x14a40:
0x14aea:    mov     eax, dword ptr [rbp + 0x28]
0x14aed:    mov     rdi, qword ptr [rbp + 0x48]
0x14af1:    lea     rsi, [rsp + 0x20]
0x14af6:    mov     qword ptr [rsp + 0x28], r14
0x14afb:    mov     dword ptr [rsp + 0x20], eax
0x14aff:    call    0x19150
0x14b04:    mov     rax, qword ptr [r13 + 8]
0x14b08:    mov     ebx, dword ptr [rbp + 0x28]
0x14b0b:    mov     qword ptr [rsp + 8], rax
0x14b10:    mov     eax, dword ptr [rbp + 0x68]
0x14b13:    mov     dword ptr [rsp + 0x1c], eax
0x14b17:    cmp     ebx, eax
0x14b19:    jge     0x14ba3
0x14b1f:    movsxd  rax, ebx
0x14b22:    lea     rcx, [rax + rax*4]
0x14b26:    shl     rcx, 4
0x14b2a:    add     rcx, qword ptr [rbp + 0x38]
0x14b2e:    mov     r15, rcx
0x14b31:    jmp     0x14b45
0x14b45:
0x14b33:    nop     dword ptr [rax + rax]
0x14b38:    add     ebx, 1
0x14b3b:    add     r15, 0x50
0x14b3f:    cmp     ebx, dword ptr [rsp + 0x1c]
0x14b43:    je      0x14ba3
0x14b45:    mov     rsi, qword ptr [r15]
0x14b48:    mov     rdi, qword ptr [rsp + 8]
0x14b4d:    mov     rdx, r14
0x14b50:    call    0x4c30
0x14b55:    test    eax, eax

```

```

0x14b57:    jne     0x14b38
0x14b59:    cmp     ebx, -1
0x14b5c:    je      0x14ba3
0x14b5e:    cmp     byte ptr [rbp + 0x6e], 0
0x14b62:    mov     qword ptr [rbp + 0x40], r14
0x14b66:    je      0x14c70
0x14b6c:    mov     dword ptr [rbp + 0x28], ebx
0x14b6f:    xor     eax, eax
0x14b71:    mov     rdi, rbp
0x14b74:    call    0xf110
0x14b79:    cmp     byte ptr [rbp + 0x6d], 0
0x14b7d:    mov     dword ptr [rbp + 0x2c], eax
0x14b80:    je      0x14b95
0x14b82:    xor     eax, eax
0x14b84:    mov     rdi, rbp
0x14b87:    call    0x10d70
0x14b8c:    mov     rax, qword ptr [rbp + 0x40]
0x14b90:    mov     qword ptr [rsp + 0x10], rax
0x14b95:    mov     byte ptr [rbp + 0x6e], 0
0x14b99:    mov     rax, qword ptr [rsp + 0x10]
0x14b9e:    jmp     0x1451c
0x1451c:
0x14ba3:    mov     dword ptr [r13], 0xffffffff
0x14bab:    mov     rax, qword ptr [rbp + 0x40]
0x14baf:    jmp     0x1451c
0x1451c:
0x14bb4:    xor     ebx, ebx
0x14bb6:    cmp     byte ptr [rbp + 0x6e], 0
0x14bba:    je      0x14c41
0x14bc0:    mov     eax, dword ptr [rbp + 0x68]
0x14bc3:    mov     dword ptr [rsp + 0x1c], eax
0x14bc7:    cmp     ebx, eax
0x14bc9:    jge     0x14ba3
0x14bcb:    movsxd  rax, ebx
0x14bce:    lea     r15, [rax + rax*4]
0x14bd2:    shl     r15, 4
0x14bd6:    add     r15, qword ptr [rbp + 0x38]
0x14bda:    jmp     0x14bf2
0x14bf2:
0x14bdc:    nop     dword ptr [rax]
0x14be0:    add     ebx, 1
0x14be3:    add     r15, 0x50
0x14be7:    cmp     ebx, dword ptr [rsp + 0x1c]
0x14beb:    mov     rcx, qword ptr [rsp + 8]
0x14bf0:    je      0x14ba3
0x14bf2:    mov     rsi, qword ptr [r15]
0x14bf5:    mov     rdx, r14
0x14bf8:    mov     rdi, rcx

```

```

0x14bfb:    mov     qword ptr [rsp + 8], rcx
0x14c00:    call   0x4c30
0x14c05:    test   eax, eax
0x14c07:    jne     0x14be0
0x14c09:    jmp     0x14b59
0x14b59:
0x14c0e:    mov     eax, dword ptr [rbp + 0x28]
0x14c11:    mov     rdi, qword ptr [rbp + 0x48]
0x14c15:    lea     rsi, [rsp + 0x20]
0x14c1a:    mov     qword ptr [rsp + 0x28], r14
0x14c1f:    mov     dword ptr [rsp + 0x20], eax
0x14c23:    call   0x19150
0x14c28:    cmp     byte ptr [rbp + 0x6e], 0
0x14c2c:    jne     0x14cee
0x14c32:    mov     eax, dword ptr [rbp + 0x28]
0x14c35:    mov     rdi, qword ptr [r13 + 8]
0x14c39:    lea     ebx, [rax - 1]
0x14c3c:    jmp     0x14aa6
0x14aa6:
0x14c41:    mov     eax, dword ptr [rbp + 0x28]
0x14c44:    mov     rdi, qword ptr [rbp + 0x48]
0x14c48:    lea     rsi, [rsp + 0x20]
0x14c4d:    mov     qword ptr [rsp + 0x28], r14
0x14c52:    mov     dword ptr [rsp + 0x20], eax
0x14c56:    call   0x19150
0x14c5b:    cmp     byte ptr [rbp + 0x6e], 0
0x14c5f:    jne     0x14c67
0x14c61:    mov     eax, dword ptr [rbp + 0x28]
0x14c64:    lea     ebx, [rax + 1]
0x14c67:    mov     rcx, qword ptr [r13 + 8]
0x14c6b:    jmp     0x14bc0
0x14bc0:
0x14c70:    sub     ebx, dword ptr [rbp + 0x28]
0x14c73:    xor     ecx, ecx
0x14c75:    mov     rdx, r12
0x14c78:    mov     esi, 0x1c
0x14c7d:    mov     r8d, ebx
0x14c80:    mov     rdi, rbp
0x14c83:    call   0x13e30
0x14c88:    jmp     0x14518
0x14518:
0x14c8d:    mov     r8d, dword ptr [rbp + 0x28]
0x14c91:    xor     ecx, ecx
0x14c93:    mov     rdx, r12
0x14c96:    mov     esi, 0x1b
0x14c9b:    mov     rdi, rbp
0x14c9e:    sub     r8d, ebx
0x14ca1:    call   0x13e30

```

```

0x14ca6:      jmp      0x1455f
0x1455f
0x14cab:      xor      eax, eax
0x14cad:      jmp      0x13f00
0x13f00
0x14cb2:      sub      edx, r8d
0x14cb5:      xor      ecx, ecx
0x14cb7:      mov      esi, 0x1c
0x14cbc:      mov      rdi, rbp
0x14cbf:      mov      r8d, edx
0x14cc2:      mov      rdx, r12
0x14cc5:      call     0x13e30
0x14cca:      jmp      0x14912
0x14912
0x14ccf:      sub      r8d, edx
0x14cd2:      xor      ecx, ecx
0x14cd4:      mov      rdx, r12
0x14cd7:      mov      esi, 0x1b
0x14cdc:      mov      rdi, rbp
0x14cdf:      call     0x13e30
0x14ce4:      jmp      0x14912
0x14912
0x14ce9:      call     0x49d0
0x14cee:      mov      rdi, qword ptr [r13 + 8]
0x14cf2:      jmp      0x14aa0
0x14aa0
0x14cf7:      nop      word ptr [rax + rax]
0x14d00:      endbr64
0x14d04:      push     rbp
0x14d05:      xor      eax, eax
0x14d07:      mov      rbp, rdi
0x14d0a:      mov      dword ptr [rdi + 0x28], 0
0x14d11:      call     0xf110
0x14d16:      mov      rsi, qword ptr [rbp + 0x18]
0x14d1a:      xor      edx, edx
0x14d1c:      mov      ecx, 0x1e
0x14d21:      mov      dword ptr [rbp + 0x2c], eax
0x14d24:      lea      rdi, [rip + 0xc677]
0x14d2b:      xor      eax, eax
0x14d2d:      call     0xf760
0x14d32:      mov      rax, qword ptr [rbp + 0x18]
0x14d36:      cmp      byte ptr [rax], 0x2f
0x14d39:      jne      0x14d51
0x14d3b:      cmp      byte ptr [rax + 1], 0
0x14d3f:      jne      0x14d51
0x14d41:      lea      rdi, [rip + 0xc672]
0x14d48:      call     0x8b00
0x14d4d:      mov      rax, qword ptr [rbp + 0x18]

```

```

0x14d51:      mov     edx, 0x2f
0x14d56:      mov     word ptr [rax], dx
0x14d59:      mov     rdi, qword ptr [rbp + 0x18]
0x14d5d:      mov     qword ptr [rbp + 0x78], 1
0x14d65:      call    0x49c0
0x14d6a:      mov     rdi, rbp
0x14d6d:      mov     r8d, 1
0x14d73:      xor     ecx, ecx
0x14d75:      xor     edx, edx
0x14d77:      mov     esi, 0x22
0x14d7c:      pop     rbp
0x14d7d:      jmp     0x13e30
0x13e30
0x14d82:      nop     word ptr cs:[rax + rax]
0x14d8d:      nop     dword ptr [rax]
0x14d90:      endbr64
0x14d94:      push    rbp
0x14d95:      cmp     byte ptr [rdi + 0x6d], 0
0x14d99:      mov     rbp, rdi
0x14d9c:      mov     eax, esi
0x14d9e:      mov     byte ptr [rdi + 0x6c], sil
0x14da2:      je      0x14db0
0x14da4:      mov     esi, dword ptr [rdi + 0x28]
0x14da7:      call    0x10760
0x14dac:      movzx   eax, byte ptr [rbp + 0x6c]
0x14db0:      test    al, al
0x14db2:      jne     0x14dc0
0x14db4:      pop     rbp
0x14db5:      ret
0x14db6:      nop     word ptr cs:[rax + rax]
0x14dc0:      mov     rdi, qword ptr [rbp + 0x18]
0x14dc4:      call    0x49c0
0x14dc9:      cmp     eax, -1
0x14dcc:      jne     0x14db4
0x14dce:      xor     eax, eax
0x14dd0:      mov     rdi, rbp
0x14dd3:      call    0x14d00
0x14dd8:      xor     eax, eax
0x14dda:      mov     rdi, rbp
0x14ddd:      call    0x10ec0
0x14de2:      xor     eax, eax
0x14de4:      mov     rdi, rbp
0x14de7:      call    0x10160
0x14dec:      cmp     byte ptr [rbp + 0x6d], 0
0x14df0:      jne     0x14e10
0x14df2:      xor     eax, eax
0x14df4:      mov     rdi, rbp
0x14df7:      call    0x10240

```



```

0x14dfc:    cmp     byte ptr [rbp + 0x6d], 0
0x14e00:    je      0x14db4
0x14e02:    mov     rdi, rbp
0x14e05:    xor     eax, eax
0x14e07:    pop     rbp
0x14e08:    jmp     0x10d70
0x10d70
0x14e0d:    nop     dword ptr [rax]
0x14e10:    mov     rdi, rbp
0x14e13:    xor     eax, eax
0x14e15:    call    0xec50
0x14e1a:    jmp     0x14df2
0x14df2
0x14e1c:    nop     dword ptr [rax]
0x14e20:    endbr64
0x14e24:    push    r15
0x14e26:    mov     r15, rsi
0x14e29:    push    r14
0x14e2b:    push    r13
0x14e2d:    push    r12
0x14e2f:    push    rbp
0x14e30:    mov     rbp, rdi
0x14e33:    push    rbx
0x14e34:    sub     rsp, 0x58
0x14e38:    mov     edx, dword ptr [rdi + 0x70]
0x14e3b:    mov     rax, qword ptr fs:[0x28]
0x14e44:    mov     qword ptr [rsp + 0x48], rax
0x14e49:    xor     eax, eax
0x14e4b:    mov     qword ptr [rsp + 0x38], 0
0x14e54:    mov     qword ptr [rsp + 0x40], 0
0x14e5d:    mov     dword ptr [rdi + 0x90], 1
0x14e67:    test    edx, edx
0x14e69:    jne     0x15138
0x14e6f:    movsxd  rax, dword ptr [rdi + 0x28]
0x14e73:    mov     rdx, rax
0x14e76:    lea     rax, [rax + rax*4]
0x14e7a:    shl     rax, 4
0x14e7e:    add     rax, qword ptr [rdi + 0x38]
0x14e82:    mov     r12, qword ptr [rax]
0x14e85:    test    edx, edx
0x14e87:    jne     0x14ea0
0x14e89:    mov     rax, qword ptr [rdi + 0x18]
0x14e8d:    cmp     byte ptr [rax], 0x2f
0x14e90:    jne     0x15110
0x14e96:    cmp     byte ptr [rax + 1], 0
0x14e9a:    jne     0x15110
0x14ea0:    mov     rdi, r12
0x14ea3:    call    0x49b0

```

```

0x14ea8:    lea     rdi, [rax + 0x11]
0x14eac:    call   0x1d670
0x14eb1:    mov     edx, 0xe
0x14eb6:    mov     rsi, r12
0x14eb9:    lea     rdi, [rip + 0x16720]
0x14ec0:    mov     r13, rax
0x14ec3:    call   0x4880
0x14ec8:    lea     rcx, [rip + 0xc500]
0x14ecf:    mov     esi, 1
0x14ed4:    mov     rdi, r13
0x14ed7:    mov     r8, rax
0x14eda:    mov     rdx, -1
0x14ee1:    xor     eax, eax
0x14ee3:    call   0x4f10
0x14ee8:    mov     rdi, r12
0x14eeb:    call   0x49b0
0x14ef0:    mov     rdi, qword ptr [r15 + 0x18]
0x14ef4:    mov     rbx, rax
0x14ef7:    call   0x49b0
0x14efc:    lea     rdi, [rbx + rax + 2]
0x14f01:    call   0x1d670
0x14f06:    mov     r8, qword ptr [r15 + 0x18]
0x14f0a:    mov     r9, r12
0x14f0d:    lea     rcx, [rip + 0xb7ed]
0x14f14:    mov     rdi, rax
0x14f17:    mov     r14, rax
0x14f1a:    mov     esi, 1
0x14f1f:    xor     eax, eax
0x14f21:    mov     rdx, -1
0x14f28:    call   0x4f10
0x14f2d:    mov     rdi, r13
0x14f30:    lea     rsi, [rsp + 0x40]
0x14f35:    mov     rdx, r14
0x14f38:    mov     rcx, qword ptr [rip + 0x166d9]
0x14f3f:    call   0x83e0
0x14f44:    mov     rdi, r13
0x14f47:    test    rax, rax
0x14f4a:    je      0x153c3
0x14f50:    call   0x18d10
0x14f55:    movsxd  rax, dword ptr [rbp + 0x28]
0x14f59:    lea     rax, [rax + rax*4]
0x14f5d:    shl     rax, 4
0x14f61:    add     rax, qword ptr [rbp + 0x38]
0x14f65:    mov     eax, dword ptr [rax + 0x10]
0x14f68:    and     eax, 0xf000
0x14f6d:    cmp     eax, 0x4000
0x14f72:    je      0x152f0
0x14f78:    mov     rdi, qword ptr [rip + 0x15509]

```

```

0x14f7f:      call    0xdb60
0x14f84:      xor     eax, eax
0x14f86:      call    0xa820
0x14f8b:      mov     rdi, qword ptr [rsp + 0x40]
0x14f90:      xor     eax, eax
0x14f92:      call    0x19940
0x14f97:      mov     rdi, qword ptr [rsp + 0x40]
0x14f9c:      mov     r13, rax
0x14f9f:      call    0x18d10
0x14fa4:      mov     rdi, r12
0x14fa7:      xor     eax, eax
0x14fa9:      mov     rsi, r13
0x14fac:      mov     qword ptr [rsp + 0x40], r13
0x14fb1:      call    0x11190
0x14fb6:      mov     rdi, r14
0x14fb9:      mov     ebx, eax
0x14fbb:      call    0x18d10
0x14fc0:      test    ebx, ebx
0x14fc2:      jne     0x153a0
0x14fc8:      call    0x4bf0
0x14fcd:      mov     ebx, eax
0x14fcf:      call    0x49e0
0x14fd4:      movsxd  rdx, dword ptr [rbp + 0x28]
0x14fd8:      mov     r9d, ebx
0x14fdb:      mov     rsi, r12
0x14fde:      mov     r8d, eax
0x14fe1:      mov     rdi, rbp
0x14fe4:      xor     eax, eax
0x14fe6:      lea     rdx, [rdx + rdx*4]
0x14fea:      shl     rdx, 4
0x14fee:      add     rdx, qword ptr [rbp + 0x38]
0x14ff2:      mov     ecx, dword ptr [rdx + 0x10]
0x14ff5:      mov     rdx, qword ptr [rsp + 0x40]
0x14ffa:      call    0x11330
0x14fff:      mov     rdi, qword ptr [rsp + 0x40]
0x15004:      mov     ebx, eax
0x15006:      call    0x18d10
0x1500b:      lea     eax, [rbx - 1]
0x1500e:      cmp     eax, 1
0x15011:      jbe     0x15039
0x15013:      cdqe
0x15015:      lea     rdx, [rip + 0x15404]
0x1501c:      xor     ecx, ecx
0x1501e:      mov     rsi, r12
0x15021:      mov     rdx, qword ptr [rdx + rax*8]
0x15025:      mov     r8d, 0x1e
0x1502b:      lea     rdi, [rip + 0xc277]
0x15032:      xor     eax, eax

```

```

0x15034:      call    0xf7e0
0x15039:      xor     eax, eax
0x1503b:      call    0xe4e0
0x15040:      mov     rdi, rbp
0x15043:      xor     eax, eax
0x15045:      call    0x10240
0x1504a:      xor     eax, eax
0x1504c:      mov     rdi, r15
0x1504f:      call    0x10240
0x15054:      xor     eax, eax
0x15056:      call    0xa820
0x1505b:      mov     rsi, qword ptr [r15 + 0x18]
0x1505f:      xor     eax, eax
0x15061:      mov     edx, 1
0x15066:      mov     rdi, r15
0x15069:      call    0xf880
0x1506e:      test    eax, eax
0x15070:      je      0x153e0
0x15076:      xor     eax, eax
0x15078:      mov     rdi, r15
0x1507b:      call    0x10ec0
0x15080:      xor     eax, eax
0x15082:      mov     rdi, r15
0x15085:      call    0x10160
0x1508a:      cmp     byte ptr [r15 + 0x6d], 0
0x1508f:      je      0x1509b
0x15091:      mov     rdi, r15
0x15094:      xor     eax, eax
0x15096:      call    0xec50
0x1509b:      xor     eax, eax
0x1509d:      mov     rdi, r15
0x150a0:      call    0x10240
0x150a5:      cmp     byte ptr [r15 + 0x6d], 0
0x150aa:      je      0x150b6
0x150ac:      mov     rdi, r15
0x150af:      xor     eax, eax
0x150b1:      call    0x10d70
0x150b6:      mov     rsi, qword ptr [rbp + 0x18]
0x150ba:      xor     eax, eax
0x150bc:      mov     edx, 1
0x150c1:      mov     rdi, rbp
0x150c4:      call    0xf880
0x150c9:      test    eax, eax
0x150cb:      je      0x153d0
0x150d1:      xor     eax, eax
0x150d3:      mov     rdi, rbp
0x150d6:      call    0x10ec0
0x150db:      xor     eax, eax

```

```

0x150dd:    mov     rdi, rbp
0x150e0:    call   0x10160
0x150e5:    cmp     byte ptr [rbp + 0x6d], 0
0x150e9:    je      0x150f5
0x150eb:    mov     rdi, rbp
0x150ee:    xor     eax, eax
0x150f0:    call   0xec50
0x150f5:    xor     eax, eax
0x150f7:    mov     rdi, rbp
0x150fa:    call   0x10240
0x150ff:    cmp     byte ptr [rbp + 0x6d], 0
0x15103:    je      0x15110
0x15105:    mov     rdi, rbp
0x15108:    xor     eax, eax
0x1510a:    call   0x10d70
0x1510f:    nop
0x15110:    mov     rax, qword ptr [rsp + 0x48]
0x15115:    sub     rax, qword ptr fs:[0x28]
0x1511e:    jne     0x153ed
0x15124:    add     rsp, 0x58
0x15128:    pop     rbx
0x15129:    pop     rbp
0x1512a:    pop     r12
0x1512c:    pop     r13
0x1512e:    pop     r14
0x15130:    pop     r15
0x15132:    ret
0x15133:    nop     dword ptr [rax + rax]
0x15138:    mov     rdx, qword ptr [rsi + 0x18]
0x1513c:    mov     rcx, qword ptr [rip + 0x164d5]
0x15143:    lea     rsi, [rsp + 0x38]
0x15148:    lea     rdi, [rip + 0xc28d]
0x1514f:    call   0x83e0
0x15154:    test    rax, rax
0x15157:    je      0x15110
0x15159:    mov     rdi, qword ptr [rbp + 0x18]
0x1515d:    mov     rsi, qword ptr [rsp + 0x38]
0x15162:    xor     eax, eax
0x15164:    call   0x11190
0x15169:    test    eax, eax
0x1516b:    jne     0x15308
0x15171:    mov     r12, qword ptr [rsp + 0x38]
0x15176:    mov     rdi, r12
0x15179:    call   0x49b0
0x1517e:    mov     rdi, r12
0x15181:    mov     rbx, rax
0x15184:    lea     rax, [rax + 1]
0x15188:    mov     qword ptr [rsp + 0x18], rax

```

```

0x1518d:    lea     rax, [rbx + 2]
0x15191:    mov     rsi, rax
0x15194:    mov     qword ptr [rsp + 0x10], rax
0x15199:    call    0x1d690
0x1519e:    mov     qword ptr [rsp + 0x38], rax
0x151a3:    mov     byte ptr [rax + rbx], 0x2f
0x151a7:    mov     rax, qword ptr [rsp + 0x38]
0x151ac:    mov     byte ptr [rax + rbx + 1], 0
0x151b1:    mov     eax, dword ptr [rbp + 0x70]
0x151b4:    mov     dword ptr [rbp + 0x50], 0xffffffff
0x151bb:    mov     dword ptr [rbp + 0x54], eax
0x151be:    mov     rdi, rbp
0x151c1:    call    0x100e0
0x151c6:    cmp     eax, -1
0x151c9:    je      0x152ce
0x151cf:    cdqe
0x151d1:    lea     rax, [rax + rax*4]
0x151d5:    shl     rax, 4
0x151d9:    mov     qword ptr [rsp + 8], rax
0x151de:    add     rax, qword ptr [rbp + 0x38]
0x151e2:    mov     r12, qword ptr [rax]
0x151e5:    mov     r14d, dword ptr [rax + 0x10]
0x151e9:    call    0x49e0
0x151ee:    mov     ebx, eax
0x151f0:    call    0x4bf0
0x151f5:    mov     rdi, r12
0x151f8:    mov     r13d, eax
0x151fb:    call    0x49b0
0x15200:    mov     rcx, qword ptr [rsp + 0x10]
0x15205:    mov     rdi, qword ptr [rsp + 0x38]
0x1520a:    lea     rsi, [rcx + rax]
0x1520e:    call    0x1d690
0x15213:    mov     rcx, qword ptr [rsp + 0x18]
0x15218:    mov     rsi, r12
0x1521b:    mov     qword ptr [rsp + 0x38], rax
0x15220:    lea     rdi, [rax + rcx]
0x15224:    call    0x48b0
0x15229:    mov     eax, dword ptr [rip + 0x16881]
0x1522f:    test    eax, eax
0x15231:    jne     0x15338
0x15237:    mov     rdi, qword ptr [rip + 0x15252]
0x1523e:    call    0xdb60
0x15243:    xor     eax, eax
0x15245:    call    0xa820
0x1524a:    mov     rdx, qword ptr [rsp + 0x38]
0x1524f:    xor     eax, eax
0x15251:    mov     r9d, r13d
0x15254:    mov     r8d, ebx

```

```

0x15257:      mov     ecx, r14d
0x1525a:      mov     rsi, r12
0x1525d:      mov     rdi, rbp
0x15260:      call    0x11330
0x15265:      cmp     eax, 1
0x15268:      je      0x15320
0x1526e:      cmp     eax, 2
0x15271:      je      0x152ce
0x15273:      cmp     eax, 3
0x15276:      je      0x151be
0x1527c:      sub     eax, 1
0x1527f:      lea     rbx, [rip + 0x1519a]
0x15286:      xor     ecx, ecx
0x15288:      mov     rsi, r12
0x1528b:      cdqe
0x1528d:      mov     r8d, 0xe
0x15293:      lea     rdi, [rip + 0xc00f]
0x1529a:      mov     rdx, qword ptr [rbx + rax*8]
0x1529e:      xor     eax, eax
0x152a0:      call    0xf7e0
0x152a5:      test    al, al
0x152a7:      je      0x152ce
0x152a9:      xor     eax, eax
0x152ab:      mov     rdi, rbp
0x152ae:      call    0x10240
0x152b3:      mov     rdi, r15
0x152b6:      xor     eax, eax
0x152b8:      call    0x10240
0x152bd:      mov     rdi, rbp
0x152c0:      call    0x100e0
0x152c5:      cmp     eax, -1
0x152c8:      jne     0x151cf
0x152ce:      mov     rdi, qword ptr [rsp + 0x38]
0x152d3:      test    rdi, rdi
0x152d6:      je      0x152dd
0x152d8:      call    0x18d10
0x152dd:      xor     eax, eax
0x152df:      call    0xe4e0
0x152e4:      xor     eax, eax
0x152e6:      call    0xa820
0x152eb:      jmp     0x1505b
0x1505b:
0x152f0:      mov     rdi, qword ptr [rip + 0x151a1]
0x152f7:      call    0xdb60
0x152fc:      jmp     0x14f84
0x14f84:
0x15301:      nop     dword ptr [rax]
0x15308:      mov     edx, 0xc

```

```

0x1530d:    xor     esi, esi
0x1530f:    lea     rdi, [rip + 0x14fea]
0x15316:    call    0x8070
0x1531b:    jmp     0x15110
0x15110:
0x15320:    mov     rax, qword ptr [rbp + 0x38]
0x15324:    mov     rbx, qword ptr [rsp + 8]
0x15329:    mov     byte ptr [rax + rbx + 0x49], 0
0x1532e:    jmp     0x152a9
0x152a9:
0x15333:    nop     dword ptr [rax + rax]
0x15338:    mov     dword ptr [rip + 0x1676e], 0
0x15342:    xor     eax, eax
0x15344:    call    0xc8c0
0x15349:    mov     edx, 4
0x1534e:    lea     rsi, [rip + 0xb1dd]
0x15355:    lea     rdi, [rip + 0xbe9d]
0x1535c:    mov     qword ptr [rsp + 0x28], rax
0x15361:    call    0x8070
0x15366:    mov     rdi, qword ptr [rsp + 0x28]
0x1536b:    mov     byte ptr [rsp + 0x27], al
0x1536f:    call    0xc970
0x15374:    xor     eax, eax
0x15376:    call    0xd870
0x1537b:    xor     eax, eax
0x1537d:    call    0xd740
0x15382:    xor     eax, eax
0x15384:    call    0xa820
0x15389:    movzx   eax, byte ptr [rsp + 0x27]
0x1538e:    and     eax, 0xffffffff
0x15391:    cmp     al, 0x4e
0x15393:    je      0x15237
0x15399:    jmp     0x152ce
0x152ce:
0x1539e:    nop
0x153a0:    mov     rdx, qword ptr [rsp + 0x40]
0x153a5:    lea     rdi, [rip + 0xc8a4]
0x153ac:    xor     ecx, ecx
0x153ae:    xor     eax, eax
0x153b0:    mov     r8d, 0x1e
0x153b6:    mov     rsi, r12
0x153b9:    call    0xf7e0
0x153be:    mov     rdi, qword ptr [rsp + 0x40]
0x153c3:    call    0x18d10
0x153c8:    jmp     0x15110
0x15110:
0x153cd:    nop     dword ptr [rax]
0x153d0:    mov     rdi, rbp

```



```

0x153d3:      call    0x14d00
0x153d8:      jmp     0x150d1
0x150d1
0x153dd:      nop     dword ptr [rax]
0x153e0:      mov     rdi, r15
0x153e3:      call    0x14d00
0x153e8:      jmp     0x15076
0x15076
0x153ed:      call    0x49d0
0x153f2:      nop     word ptr cs:[rax + rax]
0x153fd:      nop     dword ptr [rax]
0x15400:      endbr64
0x15404:      push    r15
0x15406:      push    r14
0x15408:      push    r13
0x1540a:      push    r12
0x1540c:      push    rbp
0x1540d:      mov     rbp, rdi
0x15410:      push    rbx
0x15411:      sub     rsp, 0x28
0x15415:      mov     qword ptr [rsp + 0x10], rsi
0x1541a:      mov     esi, dword ptr [rdi + 0x70]
0x1541d:      test    esi, esi
0x1541f:      jne     0x15450
0x15421:      mov     ecx, dword ptr [rdi + 0x28]
0x15424:      test    ecx, ecx
0x15426:      jne     0x15450
0x15428:      mov     rax, qword ptr [rdi + 0x18]
0x1542c:      cmp     byte ptr [rax], 0x2f
0x1542f:      je      0x15440
0x15431:      add     rsp, 0x28
0x15435:      pop     rbx
0x15436:      pop     rbp
0x15437:      pop     r12
0x15439:      pop     r13
0x1543b:      pop     r14
0x1543d:      pop     r15
0x1543f:      ret
0x15440:      cmp     byte ptr [rax + 1], 0
0x15444:      jne     0x15431
0x15446:      nop     word ptr cs:[rax + rax]
0x15450:      xor     edx, edx
0x15452:      lea     rsi, [rip + 0xb0d9]
0x15459:      lea     rdi, [rip + 0xbf97]
0x15460:      call    0x8070
0x15465:      cmp     al, 0x79
0x15467:      jne     0x15431
0x15469:      mov     edx, dword ptr [rbp + 0x68]

```

```

0x1546c:    test    edx, edx
0x1546e:    jle     0x158fc
0x15474:    mov     rax, qword ptr [rbp + 0x38]
0x15478:    xor     ecx, ecx
0x1547a:    add     rax, 0x49
0x1547e:    jmp     0x1548b
0x1548b:
0x15480:    add     ecx, 1
0x15483:    add     rax, 0x50
0x15487:    cmp     ecx, edx
0x15489:    je      0x15490
0x1548b:    cmp     byte ptr [rax], 0
0x1548e:    je      0x15480
0x15490:    mov     dword ptr [rsp + 0x1c], ecx
0x15494:    mov     eax, dword ptr [rbp + 0x70]
0x15497:    mov     dword ptr [rbp + 0x50], 0xffffffff
0x1549e:    xor     r13d, r13d
0x154a1:    lea     r15, [rip + 0xbf69]
0x154a8:    mov     dword ptr [rsp + 0x18], 1
0x154b0:    mov     dword ptr [rbp + 0x54], eax
0x154b3:    nop     dword ptr [rax + rax]
0x154b8:    mov     rdi, rbp
0x154bb:    call    0x100e0
0x154c0:    cmp     eax, -1
0x154c3:    je      0x155c6
0x154c9:    cdqe
0x154cb:    lea     rbx, [rax + rax*4]
0x154cf:    mov     rax, qword ptr [rbp + 0x38]
0x154d3:    shl     rbx, 4
0x154d7:    mov     r12, qword ptr [rax + rbx]
0x154db:    xor     eax, eax
0x154dd:    call    0x18ab0
0x154e2:    mov     rdi, r12
0x154e5:    call    0x49b0
0x154ea:    lea     rdi, [rax + 0x20]
0x154ee:    call    0x1d670
0x154f3:    mov     r8, r12
0x154f6:    mov     rcx, r15
0x154f9:    mov     esi, 1
0x154fe:    mov     r14, rax
0x15501:    mov     rdi, rax
0x15504:    mov     rdx, -1
0x1550b:    xor     eax, eax
0x1550d:    call    0x4f10
0x15512:    mov     edx, 1
0x15517:    mov     esi, 2
0x1551c:    mov     rdi, r14
0x1551f:    call    0xe480

```

```

0x15524:    xor     eax, eax
0x15526:    call   0xa820
0x1552b:    mov     rdi, r14
0x1552e:    call   0x18d10
0x15533:    mov     edx, dword ptr [rip + 0x16577]
0x15539:    test    edx, edx
0x1553b:    jne     0x15760
0x15541:    mov     eax, dword ptr [rsp + 0x18]
0x15545:    test    eax, eax
0x15547:    jne     0x15738
0x1554d:    mov     rdi, qword ptr [rip + 0x14f54]
0x15554:    call   0xdb60
0x15559:    xor     eax, eax
0x1555b:    call   0xa820
0x15560:    cmp     r13d, 0x61
0x15564:    je      0x15728
0x1556a:    cmp     r13d, 0x73
0x1556e:    je      0x154b8
0x15574:    cmp     r13d, 0x79
0x15578:    jne     0x155c6
0x1557a:    mov     rax, qword ptr [rbp + 0x38]
0x1557e:    cmp     byte ptr [rax + rbx + 0x48], 0
0x15583:    jne     0x156e8
0x15589:    mov     rdi, qword ptr [rip + 0x14f10]
0x15590:    call   0xdb60
0x15595:    xor     eax, eax
0x15597:    call   0xa820
0x1559c:    mov     rdi, r12
0x1559f:    call   0x4990
0x155a4:    test    eax, eax
0x155a6:    jne     0x157c8
0x155ac:    mov     rax, qword ptr [rbp + 0x38]
0x155b0:    mov     rdi, rbp
0x155b3:    mov     byte ptr [rax + rbx + 0x49], 0
0x155b8:    call   0x100e0
0x155bd:    cmp     eax, -1
0x155c0:    jne     0x154c9
0x155c6:    mov     eax, dword ptr [rsp + 0x1c]
0x155ca:    cmp     dword ptr [rbp + 0x68], eax
0x155cd:    je      0x155e3
0x155cf:    mov     eax, dword ptr [rsp + 0x1c]
0x155d3:    mov     rdi, rbp
0x155d6:    mov     dword ptr [rbp + 0x28], eax
0x155d9:    xor     eax, eax
0x155db:    call   0xf110
0x155e0:    mov     dword ptr [rbp + 0x2c], eax
0x155e3:    xor     eax, eax
0x155e5:    mov     rdi, rbp

```

```

0x155e8:      call    0x10240
0x155ed:      mov     rdi, qword ptr [rsp + 0x10]
0x155f2:      xor     eax, eax
0x155f4:      call    0x10240
0x155f9:      xor     eax, eax
0x155fb:      call    0xe4e0
0x15600:      xor     eax, eax
0x15602:      call    0xa820
0x15607:      mov     rsi, qword ptr [rbp + 0x18]
0x1560b:      xor     eax, eax
0x1560d:      mov     edx, 1
0x15612:      mov     rdi, rbp
0x15615:      call    0xf880
0x1561a:      test    eax, eax
0x1561c:      je      0x158e0
0x15622:      xor     eax, eax
0x15624:      mov     rdi, rbp
0x15627:      call    0x10ec0
0x1562c:      xor     eax, eax
0x1562e:      mov     rdi, rbp
0x15631:      call    0x10160
0x15636:      cmp     byte ptr [rbp + 0x6d], 0
0x1563a:      jne     0x15890
0x15640:      xor     eax, eax
0x15642:      mov     rdi, rbp
0x15645:      call    0x10240
0x1564a:      cmp     byte ptr [rbp + 0x6d], 0
0x1564e:      jne     0x15880
0x15654:      mov     rax, qword ptr [rsp + 0x10]
0x15659:      mov     rdi, qword ptr [rbp + 0x18]
0x1565d:      mov     r12, qword ptr [rax + 0x18]
0x15661:      mov     rsi, r12
0x15664:      call    0x4b30
0x15669:      test    eax, eax
0x1566b:      jne     0x15431
0x15671:      mov     rdi, qword ptr [rsp + 0x10]
0x15676:      mov     edx, 1
0x1567b:      mov     rsi, r12
0x1567e:      call    0xf880
0x15683:      test    eax, eax
0x15685:      je      0x158ed
0x1568b:      mov     rbx, qword ptr [rsp + 0x10]
0x15690:      xor     eax, eax
0x15692:      mov     rdi, rbx
0x15695:      call    0x10ec0
0x1569a:      xor     eax, eax
0x1569c:      mov     rdi, rbx
0x1569f:      call    0x10160

```

```

0x156a4:    cmp     byte ptr [rbx + 0x6d], 0
0x156a8:    je      0x156b4
0x156aa:    mov     rdi, rbx
0x156ad:    xor     eax, eax
0x156af:    call    0xec50
0x156b4:    mov     rbx, qword ptr [rsp + 0x10]
0x156b9:    xor     eax, eax
0x156bb:    mov     rdi, rbx
0x156be:    call    0x10240
0x156c3:    cmp     byte ptr [rbx + 0x6d], 0
0x156c7:    je      0x15431
0x156cd:    add     rsp, 0x28
0x156d1:    mov     rdi, rbx
0x156d4:    xor     eax, eax
0x156d6:    pop     rbx
0x156d7:    pop     rbp
0x156d8:    pop     r12
0x156da:    pop     r13
0x156dc:    pop     r14
0x156de:    pop     r15
0x156e0:    jmp     0x10d70
0x10d70:
0x156e5:    nop     dword ptr [rax]
0x156e8:    mov     rdi, r12
0x156eb:    call    0x4870
0x156f0:    test    eax, eax
0x156f2:    je      0x155ac
0x156f8:    xor     eax, eax
0x156fa:    mov     ecx, 0xe
0x156ff:    mov     rsi, r12
0x15702:    lea     rdx, [rip + 0xae29]
0x15709:    lea     rdi, [rip + 0xc5e8]
0x15710:    call    0xf760
0x15715:    cmp     al, 0x79
0x15717:    je      0x154b8
0x1571d:    jmp     0x155c6
0x155c6:
0x15722:    nop     word ptr [rax + rax]
0x15728:    mov     dword ptr [rsp + 0x18], 0
0x15730:    jmp     0x1557a
0x1557a:
0x15735:    nop     dword ptr [rax]
0x15738:    mov     ecx, 2
0x1573d:    lea     rdx, [rip + 0xbcd9]
0x15744:    mov     rsi, r12
0x15747:    xor     eax, eax
0x15749:    lea     rdi, [rip + 0xc528]
0x15750:    call    0xf760

```

```

0x15755:      movsx    r13d, al
0x15759:      jmp     0x1554d
0x1554d:
0x1575e:      nop
0x15760:      mov     dword ptr [rip + 0x16346], 0
0x1576a:      xor     eax, eax
0x1576c:      call    0xc8c0
0x15771:      mov     edx, 4
0x15776:      lea     rsi, [rip + 0xad5]
0x1577d:      lea     rdi, [rip + 0xba75]
0x15784:      mov     qword ptr [rsp + 8], rax
0x15789:      call    0x8070
0x1578e:      mov     rdi, qword ptr [rsp + 8]
0x15793:      mov     r14d, eax
0x15796:      call    0xc970
0x1579b:      xor     eax, eax
0x1579d:      and     r14d, 0xffffffff
0x157a1:      call    0xd870
0x157a6:      xor     eax, eax
0x157a8:      call    0xd740
0x157ad:      xor     eax, eax
0x157af:      call    0xa820
0x157b4:      cmp     r14b, 0x4e
0x157b8:      je      0x15541
0x157be:      jmp     0x155c6
0x155c6:
0x157c3:      nop     dword ptr [rax + rax]
0x157c8:      xor     eax, eax
0x157ca:      mov     ecx, 0x12
0x157cf:      mov     rsi, r12
0x157d2:      lea     rdx, [rip + 0xad59]
0x157d9:      lea     rdi, [rip + 0xc4c0]
0x157e0:      call    0xf760
0x157e5:      cmp     al, 0x79
0x157e7:      jne     0x156f8
0x157ed:      mov     rdi, r12
0x157f0:      call    0x49b0
0x157f5:      lea     rdi, [rax + 0x21]
0x157f9:      call    0x1d670
0x157fe:      mov     rdx, -1
0x15805:      mov     r8, r12
0x15808:      lea     rcx, [rip + 0xbb38]
0x1580f:      mov     r14, rax
0x15812:      mov     rdi, rax
0x15815:      mov     esi, 1
0x1581a:      xor     eax, eax
0x1581c:      call    0x4f10
0x15821:      mov     esi, 1

```

```

0x15826:      mov     rdi, r14
0x15829:      call   0x193f0
0x1582e:      mov     rdi, r14
0x15831:      mov     dword ptr [rsp + 8], eax
0x15835:      call   0x18d10
0x1583a:      mov     rdi, qword ptr [rbp + 0x18]
0x1583e:      call   0xc4b0
0x15843:      mov     eax, dword ptr [rsp + 8]
0x15847:      mov     edx, eax
0x15849:      and     edx, 0x7f
0x1584c:      add     edx, 1
0x1584f:      cmp     dl, 1
0x15852:      jg      0x158b0
0x15854:      test    al, 0x7f
0x15856:      jne     0x156f8
0x1585c:      test    ah, 0xff
0x1585f:      je      0x155ac
0x15865:      lea     rdi, [rip + 0xb37e]
0x1586c:      call   0x195e0
0x15871:      jmp     0x156f8
0x156f8
0x15876:      nop     word ptr cs:[rax + rax]
0x15880:      mov     rdi, rbp
0x15883:      xor     eax, eax
0x15885:      call   0x10d70
0x1588a:      jmp     0x15654
0x15654
0x1588f:      nop
0x15890:      xor     eax, eax
0x15892:      mov     rdi, rbp
0x15895:      call   0xec50
0x1589a:      xor     eax, eax
0x1589c:      mov     rdi, rbp
0x1589f:      call   0x10240
0x158a4:      cmp     byte ptr [rbp + 0x6d], 0
0x158a8:      je      0x15654
0x158ae:      jmp     0x15880
0x15880
0x158b0:      xor     eax, eax
0x158b2:      mov     ecx, 0xe
0x158b7:      mov     rsi, r12
0x158ba:      lea     rdx, [rip + 0xac71]
0x158c1:      lea     rdi, [rip + 0xc408]
0x158c8:      call   0xf760
0x158cd:      cmp     al, 0x79
0x158cf:      je      0x154b8
0x158d5:      jmp     0x155c6
0x155c6

```

```

0x158da:      nop      word ptr [rax + rax]
0x158e0:      mov      rdi, rbp
0x158e3:      call     0x14d00
0x158e8:      jmp      0x15622
0x15622
0x158ed:      mov      rdi, qword ptr [rsp + 0x10]
0x158f2:      call     0x14d00
0x158f7:      jmp      0x1568b
0x1568b
0x158fc:      mov      dword ptr [rsp + 0x1c], 0
0x15904:      jmp      0x15494
0x15494
0x15909:      nop      dword ptr [rax]
0x15910:      endbr64
0x15914:      push     r13
0x15916:      xor      edx, edx
0x15918:      push     r12
0x1591a:      push     rbp
0x1591b:      mov      rbp, rsi
0x1591e:      push     rbx
0x1591f:      mov      rbx, rdi
0x15922:      lea      rdi, [rip + 0xbaf9]
0x15929:      sub      rsp, 0x18
0x1592d:      mov      rcx, qword ptr [rip + 0x15cd4]
0x15934:      mov      rax, qword ptr fs:[0x28]
0x1593d:      mov      qword ptr [rsp + 8], rax
0x15942:      xor      eax, eax
0x15944:      mov      rsi, rsp
0x15947:      mov      qword ptr [rsp], 0
0x1594f:      call     0x83e0
0x15954:      test     rax, rax
0x15957:      je       0x15a0b
0x1595d:      mov      rdi, qword ptr [rsp]
0x15961:      cmp      byte ptr [rdi], 0
0x15964:      je       0x15a06
0x1596a:      xor      eax, eax
0x1596c:      call     0x19940
0x15971:      mov      rdi, qword ptr [rsp]
0x15975:      mov      r13, rax
0x15978:      call     0x18d10
0x1597d:      mov      rdi, r13
0x15980:      mov      qword ptr [rsp], r13
0x15984:      call     0x49b0
0x15989:      mov      rdi, r13
0x1598c:      lea      rsi, [rax + 2]
0x15990:      mov      r12, rax
0x15993:      call     0x4ce0
0x15998:      mov      esi, 0x41ff

```



```

0x1599d:    mov     qword ptr [rsp], rax
0x159a1:    mov     byte ptr [rax + r12], 0x2f
0x159a6:    mov     rax, qword ptr [rsp]
0x159aa:    mov     byte ptr [rax + r12 + 1], 0
0x159b0:    mov     rdi, qword ptr [rsp]
0x159b4:    xor     eax, eax
0x159b6:    call    0x124d0
0x159bb:    cmp     eax, -1
0x159be:    je      0x15a90
0x159c4:    mov     rsi, qword ptr [rbx + 0x18]
0x159c8:    xor     eax, eax
0x159ca:    mov     edx, 1
0x159cf:    mov     rdi, rbx
0x159d2:    call    0xf880
0x159d7:    test    eax, eax
0x159d9:    jne     0x15a30
0x159db:    mov     rdi, rbx
0x159de:    call    0x14d00
0x159e3:    mov     rsi, qword ptr [rbp + 0x18]
0x159e7:    mov     rdi, qword ptr [rbx + 0x18]
0x159eb:    call    0x4b30
0x159f0:    test    eax, eax
0x159f2:    je      0x15af0
0x159f8:    mov     rdi, rbp
0x159fb:    xor     eax, eax
0x159fd:    call    0x10240
0x15a02:    mov     rdi, qword ptr [rsp]
0x15a06:    call    0x18d10
0x15a0b:    mov     rax, qword ptr [rsp + 8]
0x15a10:    sub     rax, qword ptr fs:[0x28]
0x15a19:    jne     0x15ba4
0x15a1f:    add     rsp, 0x18
0x15a23:    pop     rbx
0x15a24:    pop     rbp
0x15a25:    pop     r12
0x15a27:    pop     r13
0x15a29:    ret
0x15a2a:    nop     word ptr [rax + rax]
0x15a30:    mov     rsi, qword ptr [rsp]
0x15a34:    mov     rdi, rbx
0x15a37:    xor     eax, eax
0x15a39:    call    0x122d0
0x15a3e:    mov     edx, dword ptr [rbx + 0x60]
0x15a41:    mov     esi, dword ptr [rbx + 0x68]
0x15a44:    mov     edi, 0
0x15a49:    mov     dword ptr [rbx + 0x28], eax
0x15a4c:    sub     edx, 2
0x15a4f:    mov     ecx, edx

```

```

0x15a51:      shr     ecx, 0x1f
0x15a54:      add     ecx, edx
0x15a56:      sar     ecx, 1
0x15a58:      sub     eax, ecx
0x15a5a:      lea     ecx, [rdx + rax]
0x15a5d:      cmovs   eax, edi
0x15a60:      cmp     esi, ecx
0x15a62:      jg      0x15a70
0x15a64:      mov     eax, esi
0x15a66:      sub     eax, edx
0x15a68:      mov     edx, 0
0x15a6d:      cmovs   eax, edx
0x15a70:      cmp     byte ptr [rbx + 0x6d], 0
0x15a74:      mov     dword ptr [rbx + 0x2c], eax
0x15a77:      jne     0x15ac0
0x15a79:      mov     rdi, rbx
0x15a7c:      xor     eax, eax
0x15a7e:      call    0x10240
0x15a83:      jmp     0x159e3
0x159e3:
0x15a88:      nop     dword ptr [rax + rax]
0x15a90:      mov     rsi, qword ptr [rsp]
0x15a94:      mov     ecx, 0x1c
0x15a99:      xor     edx, edx
0x15a9b:      xor     eax, eax
0x15a9d:      lea     rdi, [rip + 0xb993]
0x15aa4:      call    0xf760
0x15aa9:      mov     rdi, qword ptr [rsp]
0x15aad:      call    0x18d10
0x15ab2:      jmp     0x15a0b
0x15a0b:
0x15ab7:      nop     word ptr [rax + rax]
0x15ac0:      xor     eax, eax
0x15ac2:      mov     rdi, rbx
0x15ac5:      call    0x10d70
0x15aca:      cmp     byte ptr [rbx + 0x6d], 0
0x15ace:      je      0x15a79
0x15ad0:      mov     rdi, rbx
0x15ad3:      xor     eax, eax
0x15ad5:      call    0xec50
0x15ada:      mov     rdi, rbx
0x15add:      xor     eax, eax
0x15adf:      call    0x10240
0x15ae4:      jmp     0x159e3
0x159e3:
0x15ae9:      nop     dword ptr [rax]
0x15af0:      movsxd  rax, dword ptr [rbp + 0x28]
0x15af4:      lea     rax, [rax + rax*4]

```

```

0x15af8:      shl     rax, 4
0x15afc:      add     rax, qword ptr [rbp + 0x38]
0x15b00:      mov     rdi, qword ptr [rax]
0x15b03:      call    0x1d7a0
0x15b08:      mov     rsi, qword ptr [rbp + 0x18]
0x15b0c:      mov     edx, 1
0x15b11:      mov     rdi, rbp
0x15b14:      mov     r12, rax
0x15b17:      xor     eax, eax
0x15b19:      call    0xf880
0x15b1e:      test    eax, eax
0x15b20:      jne     0x15b40
0x15b22:      mov     rdi, rbp
0x15b25:      call    0x14d00
0x15b2a:      mov     rdi, r12
0x15b2d:      call    0x18d10
0x15b32:      jmp     0x159f8
0x159f8
0x15b37:      nop     word ptr [rax + rax]
0x15b40:      mov     rsi, r12
0x15b43:      mov     rdi, rbp
0x15b46:      xor     eax, eax
0x15b48:      call    0x122d0
0x15b4d:      mov     edx, dword ptr [rbp + 0x60]
0x15b50:      mov     esi, dword ptr [rbp + 0x68]
0x15b53:      mov     edi, 0
0x15b58:      mov     dword ptr [rbp + 0x28], eax
0x15b5b:      sub     edx, 2
0x15b5e:      mov     ecx, edx
0x15b60:      shr     ecx, 0x1f
0x15b63:      add     ecx, edx
0x15b65:      sar     ecx, 1
0x15b67:      sub     eax, ecx
0x15b69:      lea     ecx, [rdx + rax]
0x15b6c:      cmovs   eax, edi
0x15b6f:      cmp     esi, ecx
0x15b71:      jg      0x15b7f
0x15b73:      mov     eax, esi
0x15b75:      sub     eax, edx
0x15b77:      mov     edx, 0
0x15b7c:      cmovs   eax, edx
0x15b7f:      cmp     byte ptr [rbp + 0x6d], 0
0x15b83:      mov     dword ptr [rbp + 0x2c], eax
0x15b86:      je      0x15b2a
0x15b88:      xor     eax, eax
0x15b8a:      mov     rdi, rbp
0x15b8d:      call    0x10d70
0x15b92:      cmp     byte ptr [rbp + 0x6d], 0

```

```

0x15b96:    je      0x15b2a
0x15b98:    mov     rdi, rbp
0x15b9b:    xor     eax, eax
0x15b9d:    call    0xec50
0x15ba2:    jmp     0x15b2a
0x15b2a:
0x15ba4:    call    0x49d0
0x15ba9:    nop     dword ptr [rax]
0x15bb0:    endbr64
0x15bb4:    push    r15
0x15bb6:    mov     r15, rdi
0x15bb9:    push    r14
0x15bbb:    push    r13
0x15bbd:    push    r12
0x15bbf:    mov     r12, rsi
0x15bc2:    push    rbp
0x15bc3:    push    rbx
0x15bc4:    sub     rsp, 0x48
0x15bc8:    mov     edx, dword ptr [rdi + 0x70]
0x15bcb:    mov     rax, qword ptr fs:[0x28]
0x15bd4:    mov     qword ptr [rsp + 0x38], rax
0x15bd9:    xor     eax, eax
0x15bdb:    mov     qword ptr [rsp + 0x28], 0
0x15be4:    mov     qword ptr [rsp + 0x30], 0
0x15bed:    mov     dword ptr [rdi + 0x90], 1
0x15bf7:    test    edx, edx
0x15bf9:    jne     0x15ec8
0x15bff:    movsxd  rax, dword ptr [rdi + 0x28]
0x15c03:    mov     rdx, rax
0x15c06:    lea     rax, [rax + rax*4]
0x15c0a:    shl     rax, 4
0x15c0e:    add     rax, qword ptr [rdi + 0x38]
0x15c12:    mov     rbp, qword ptr [rax]
0x15c15:    test    edx, edx
0x15c17:    jne     0x15c30
0x15c19:    mov     rax, qword ptr [rdi + 0x18]
0x15c1d:    cmp     byte ptr [rax], 0x2f
0x15c20:    jne     0x15ea0
0x15c26:    cmp     byte ptr [rax + 1], 0
0x15c2a:    jne     0x15ea0
0x15c30:    mov     rdi, rbp
0x15c33:    call    0x49b0
0x15c38:    lea     rdi, [rax + 0x11]
0x15c3c:    call    0x1d670
0x15c41:    mov     edx, 0xe
0x15c46:    mov     rsi, rbp
0x15c49:    lea     rdi, [rip + 0x15990]
0x15c50:    mov     r13, rax

```

```

0x15c53:      call    0x4880
0x15c58:      lea     rcx, [rip + 0xb7ef]
0x15c5f:      mov     esi, 1
0x15c64:      mov     rdi, r13
0x15c67:      mov     r8, rax
0x15c6a:      mov     rdx, -1
0x15c71:      xor     eax, eax
0x15c73:      call    0x4f10
0x15c78:      mov     rdi, rbp
0x15c7b:      call    0x49b0
0x15c80:      mov     rdi, qword ptr [r12 + 0x18]
0x15c85:      mov     rbx, rax
0x15c88:      call    0x49b0
0x15c8d:      lea     rdi, [rbx + rax + 2]
0x15c92:      call    0x1d670
0x15c97:      mov     r8, qword ptr [r12 + 0x18]
0x15c9c:      mov     r9, rbp
0x15c9f:      lea     rcx, [rip + 0xaa5b]
0x15ca6:      mov     rdi, rax
0x15ca9:      mov     r14, rax
0x15cac:      mov     esi, 1
0x15cb1:      xor     eax, eax
0x15cb3:      mov     rdx, -1
0x15cba:      call    0x4f10
0x15cbf:      mov     rdi, r13
0x15cc2:      lea     rsi, [rsp + 0x30]
0x15cc7:      mov     rdx, r14
0x15cca:      mov     rcx, qword ptr [rip + 0x1593f]
0x15cd1:      call    0x83e0
0x15cd6:      mov     rdi, r13
0x15cd9:      test    rax, rax
0x15cdc:      je      0x16173
0x15ce2:      call    0x18d10
0x15ce7:      movsxd  rax, dword ptr [r15 + 0x28]
0x15ceb:      lea     rax, [rax + rax*4]
0x15cef:      shl     rax, 4
0x15cf3:      add     rax, qword ptr [r15 + 0x38]
0x15cf7:      mov     eax, dword ptr [rax + 0x10]
0x15cfa:      and     eax, 0xf000
0x15cff:      cmp     eax, 0x4000
0x15d04:      je      0x160a0
0x15d0a:      mov     rdi, qword ptr [rip + 0x1479f]
0x15d11:      call    0xdb60
0x15d16:      xor     eax, eax
0x15d18:      call    0xa820
0x15d1d:      mov     rdi, qword ptr [rsp + 0x30]
0x15d22:      xor     eax, eax
0x15d24:      call    0x19940

```

```

0x15d29:    mov     rdi, qword ptr [rsp + 0x30]
0x15d2e:    mov     r13, rax
0x15d31:    call    0x18d10
0x15d36:    mov     rdi, rbp
0x15d39:    xor     eax, eax
0x15d3b:    mov     rsi, r13
0x15d3e:    mov     qword ptr [rsp + 0x30], r13
0x15d43:    call    0x11190
0x15d48:    mov     rdi, r14
0x15d4b:    mov     ebx, eax
0x15d4d:    call    0x18d10
0x15d52:    test    ebx, ebx
0x15d54:    jne     0x16150
0x15d5a:    movsxd  rax, dword ptr [r15 + 0x28]
0x15d5e:    mov     rdx, qword ptr [rsp + 0x30]
0x15d63:    mov     rsi, rbp
0x15d66:    mov     rdi, r15
0x15d69:    lea     rax, [rax + rax*4]
0x15d6d:    shl     rax, 4
0x15d71:    add     rax, qword ptr [r15 + 0x38]
0x15d75:    mov     ecx, dword ptr [rax + 0x10]
0x15d78:    xor     eax, eax
0x15d7a:    call    0x11d10
0x15d7f:    cmp     eax, 1
0x15d82:    je      0x15db6
0x15d84:    cmp     eax, 2
0x15d87:    je      0x161a0
0x15d8d:    sub     eax, 1
0x15d90:    lea     rdx, [rip + 0x14629]
0x15d97:    xor     ecx, ecx
0x15d99:    mov     rsi, rbp
0x15d9c:    cdqe
0x15d9e:    mov     r8d, 0xe
0x15da4:    lea     rdi, [rip + 0xb6b0]
0x15dab:    mov     rdx, qword ptr [rdx + rax*8]
0x15daf:    xor     eax, eax
0x15db1:    call    0xf7e0
0x15db6:    mov     rdi, qword ptr [rsp + 0x30]
0x15dbb:    call    0x18d10
0x15dc0:    xor     eax, eax
0x15dc2:    call    0xe4e0
0x15dc7:    mov     rdi, r15
0x15dca:    xor     eax, eax
0x15dcc:    call    0x10240
0x15dd1:    xor     eax, eax
0x15dd3:    mov     rdi, r12
0x15dd6:    call    0x10240
0x15ddb:    xor     eax, eax

```

```

0x15ddd:    call    0xa820
0x15de2:    mov     rsi, qword ptr [r12 + 0x18]
0x15de7:    xor     eax, eax
0x15de9:    mov     edx, 1
0x15dee:    mov     rdi, r12
0x15df1:    call    0xf880
0x15df6:    test    eax, eax
0x15df8:    je      0x16190
0x15dfe:    xor     eax, eax
0x15e00:    mov     rdi, r12
0x15e03:    call    0x10ec0
0x15e08:    xor     eax, eax
0x15e0a:    mov     rdi, r12
0x15e0d:    call    0x10160
0x15e12:    cmp     byte ptr [r12 + 0x6d], 0
0x15e18:    je      0x15e24
0x15e1a:    mov     rdi, r12
0x15e1d:    xor     eax, eax
0x15e1f:    call    0xec50
0x15e24:    xor     eax, eax
0x15e26:    mov     rdi, r12
0x15e29:    call    0x10240
0x15e2e:    cmp     byte ptr [r12 + 0x6d], 0
0x15e34:    je      0x15e40
0x15e36:    mov     rdi, r12
0x15e39:    xor     eax, eax
0x15e3b:    call    0x10d70
0x15e40:    mov     rsi, qword ptr [r15 + 0x18]
0x15e44:    xor     eax, eax
0x15e46:    mov     edx, 1
0x15e4b:    mov     rdi, r15
0x15e4e:    call    0xf880
0x15e53:    test    eax, eax
0x15e55:    je      0x16180
0x15e5b:    xor     eax, eax
0x15e5d:    mov     rdi, r15
0x15e60:    call    0x10ec0
0x15e65:    xor     eax, eax
0x15e67:    mov     rdi, r15
0x15e6a:    call    0x10160
0x15e6f:    cmp     byte ptr [r15 + 0x6d], 0
0x15e74:    je      0x15e80
0x15e76:    mov     rdi, r15
0x15e79:    xor     eax, eax
0x15e7b:    call    0xec50
0x15e80:    xor     eax, eax
0x15e82:    mov     rdi, r15
0x15e85:    call    0x10240

```

```

0x15e8a:    cmp     byte ptr [r15 + 0x6d], 0
0x15e8f:    je      0x15ea0
0x15e91:    mov     rdi, r15
0x15e94:    xor     eax, eax
0x15e96:    call    0x10d70
0x15e9b:    nop     dword ptr [rax + rax]
0x15ea0:    mov     rax, qword ptr [rsp + 0x38]
0x15ea5:    sub     rax, qword ptr fs:[0x28]
0x15eae:    jne     0x161bd
0x15eb4:    add     rsp, 0x48
0x15eb8:    pop     rbx
0x15eb9:    pop     rbp
0x15eba:    pop     r12
0x15ebc:    pop     r13
0x15ebe:    pop     r14
0x15ec0:    pop     r15
0x15ec2:    ret
0x15ec3:    nop     dword ptr [rax + rax]
0x15ec8:    mov     rdx, qword ptr [rsi + 0x18]
0x15ecc:    mov     rcx, qword ptr [rip + 0x1573d]
0x15ed3:    lea     rsi, [rsp + 0x28]
0x15ed8:    lea     rdi, [rip + 0xb591]
0x15edf:    call    0x83e0
0x15ee4:    test    rax, rax
0x15ee7:    je      0x15ea0
0x15ee9:    mov     rdi, qword ptr [r15 + 0x18]
0x15eed:    mov     rsi, qword ptr [rsp + 0x28]
0x15ef2:    xor     eax, eax
0x15ef4:    call    0x11190
0x15ef9:    mov     r13d, eax
0x15efc:    test    eax, eax
0x15efe:    jne     0x160b8
0x15f04:    mov     rbp, qword ptr [rsp + 0x28]
0x15f09:    mov     rdi, rbp
0x15f0c:    call    0x49b0
0x15f11:    mov     rdi, rbp
0x15f14:    mov     rbx, rax
0x15f17:    lea     rax, [rax + 1]
0x15f1b:    lea     r14, [rbx + 2]
0x15f1f:    mov     qword ptr [rsp + 8], rax
0x15f24:    mov     rsi, r14
0x15f27:    call    0x1d690
0x15f2c:    mov     qword ptr [rsp + 0x28], rax
0x15f31:    mov     byte ptr [rax + rbx], 0x2f
0x15f35:    mov     rax, qword ptr [rsp + 0x28]
0x15f3a:    mov     byte ptr [rax + rbx + 1], 0
0x15f3f:    mov     edx, dword ptr [r15 + 0x68]
0x15f43:    test    edx, edx

```



```

0x15f45:      jle      0x15f6a
0x15f47:      mov     rax, qword ptr [r15 + 0x38]
0x15f4b:      add     rax, 0x49
0x15f4f:      jmp     0x15f65
0x15f65:
0x15f51:      nop     dword ptr [rax]
0x15f58:      add     r13d, 1
0x15f5c:      add     rax, 0x50
0x15f60:      cmp     r13d, edx
0x15f63:      je      0x15f6a
0x15f65:      cmp     byte ptr [rax], 0
0x15f68:      je      0x15f58
0x15f6a:      mov     eax, dword ptr [r15 + 0x70]
0x15f6e:      mov     dword ptr [r15 + 0x50], 0xffffffff
0x15f76:      mov     dword ptr [r15 + 0x54], eax
0x15f7a:      nop     word ptr [rax + rax]
0x15f80:      mov     rdi, r15
0x15f83:      call    0x100e0
0x15f88:      cmp     eax, -1
0x15f8b:      je      0x16060
0x15f91:      cdqe
0x15f93:      lea     rbx, [rax + rax*4]
0x15f97:      mov     rax, qword ptr [r15 + 0x38]
0x15f9b:      shl     rbx, 4
0x15f9f:      mov     rbp, qword ptr [rax + rbx]
0x15fa3:      xor     eax, eax
0x15fa5:      call    0x18ab0
0x15faa:      mov     rdi, rbp
0x15fad:      call    0x49b0
0x15fb2:      mov     rdi, qword ptr [rsp + 0x28]
0x15fb7:      lea     rsi, [r14 + rax]
0x15fbb:      call    0x1d690
0x15fc0:      mov     rcx, qword ptr [rsp + 8]
0x15fc5:      mov     rsi, rbp
0x15fc8:      mov     qword ptr [rsp + 0x28], rax
0x15fcd:      lea     rdi, [rax + rcx]
0x15fd1:      call    0x48b0
0x15fd6:      mov     eax, dword ptr [rip + 0x15ad4]
0x15fdc:      test    eax, eax
0x15fde:      jne     0x160e0
0x15fe4:      mov     rdi, qword ptr [rip + 0x144cd]
0x15feb:      call    0xdb60
0x15ff0:      xor     eax, eax
0x15ff2:      call    0xa820
0x15ff7:      mov     rax, qword ptr [r15 + 0x38]
0x15ffb:      mov     rsi, rbp
0x15ffe:      mov     rdi, r15
0x16001:      mov     rdx, qword ptr [rsp + 0x28]

```

```

0x16006:      mov     ecx, dword ptr [rax + rbx + 0x10]
0x1600a:      xor     eax, eax
0x1600c:      call    0x11d10
0x16011:      cmp     eax, 1
0x16014:      je      0x160d0
0x1601a:      cmp     eax, 2
0x1601d:      je      0x16060
0x1601f:      cmp     eax, 3
0x16022:      je      0x15f80
0x16028:      sub     eax, 1
0x1602b:      lea     rdx, [rip + 0x1438e]
0x16032:      xor     ecx, ecx
0x16034:      mov     rsi, rbp
0x16037:      cdqe
0x16039:      mov     r8d, 0xe
0x1603f:      lea     rdi, [rip + 0xb415]
0x16046:      mov     rdx, qword ptr [rdx + rax*8]
0x1604a:      xor     eax, eax
0x1604c:      call    0xf7e0
0x16051:      test    al, al
0x16053:      jne     0x15f80
0x16059:      nop     dword ptr [rax]
0x16060:      mov     rdi, qword ptr [rsp + 0x28]
0x16065:      test    rdi, rdi
0x16068:      je      0x1606f
0x1606a:      call    0x18d10
0x1606f:      cmp     dword ptr [r15 + 0x68], r13d
0x16073:      je      0x16087
0x16075:      mov     dword ptr [r15 + 0x28], r13d
0x16079:      mov     rdi, r15
0x1607c:      xor     eax, eax
0x1607e:      call    0xf110
0x16083:      mov     dword ptr [r15 + 0x2c], eax
0x16087:      xor     eax, eax
0x16089:      call    0xe4e0
0x1608e:      xor     eax, eax
0x16090:      call    0xa820
0x16095:      jmp     0x15de2
0x15de2
0x1609a:      nop     word ptr [rax + rax]
0x160a0:      mov     rdi, qword ptr [rip + 0x14419]
0x160a7:      call    0xdb60
0x160ac:      jmp     0x15d16
0x15d16
0x160b1:      nop     dword ptr [rax]
0x160b8:      mov     edx, 0xc
0x160bd:      xor     esi, esi
0x160bf:      lea     rdi, [rip + 0x1423a]

```

```

0x160c6:      call    0x8070
0x160cb:      jmp     0x15ea0
0x15ea0
0x160d0:      mov     rax, qword ptr [r15 + 0x38]
0x160d4:      mov     byte ptr [rax + rbx + 0x49], 0
0x160d9:      jmp     0x15f80
0x15f80
0x160de:      nop
0x160e0:      mov     dword ptr [rip + 0x159c6], 0
0x160ea:      xor     eax, eax
0x160ec:      call    0xc8c0
0x160f1:      mov     edx, 4
0x160f6:      lea     rsi, [rip + 0xa435]
0x160fd:      lea     rdi, [rip + 0xb0f5]
0x16104:      mov     qword ptr [rsp + 0x18], rax
0x16109:      call    0x8070
0x1610e:      mov     rdi, qword ptr [rsp + 0x18]
0x16113:      mov     byte ptr [rsp + 0x17], al
0x16117:      call    0xc970
0x1611c:      xor     eax, eax
0x1611e:      call    0xd870
0x16123:      xor     eax, eax
0x16125:      call    0xd740
0x1612a:      xor     eax, eax
0x1612c:      call    0xa820
0x16131:      movzx   eax, byte ptr [rsp + 0x17]
0x16136:      and     eax, 0xffffffff
0x16139:      cmp     al, 0x4e
0x1613b:      je      0x15fe4
0x16141:      jmp     0x16060
0x16060
0x16146:      nop     word ptr cs:[rax + rax]
0x16150:      mov     rdx, qword ptr [rsp + 0x30]
0x16155:      lea     rdi, [rip + 0xbaf4]
0x1615c:      xor     ecx, ecx
0x1615e:      xor     eax, eax
0x16160:      mov     r8d, 0x1e
0x16166:      mov     rsi, rbp
0x16169:      call    0xf7e0
0x1616e:      mov     rdi, qword ptr [rsp + 0x30]
0x16173:      call    0x18d10
0x16178:      jmp     0x15ea0
0x15ea0
0x1617d:      nop     dword ptr [rax]
0x16180:      mov     rdi, r15
0x16183:      call    0x14d00
0x16188:      jmp     0x15e5b
0x15e5b

```

```

0x1618d:      nop      dword ptr [rax]
0x16190:      mov      rdi, r12
0x16193:      call     0x14d00
0x16198:      jmp      0x15dfe
0x15dfe
0x1619d:      nop      dword ptr [rax]
0x161a0:      mov      rdi, qword ptr [rsp + 0x30]
0x161a5:      call     0x18d10
0x161aa:      xor      eax, eax
0x161ac:      call     0xe4e0
0x161b1:      xor      eax, eax
0x161b3:      call     0xa820
0x161b8:      jmp      0x15ea0
0x15ea0
0x161bd:      call     0x49d0
0x161c2:      nop      word ptr cs:[rax + rax]
0x161cd:      nop      dword ptr [rax]
0x161d0:      endbr64
0x161d4:      push     r13
0x161d6:      xor      r13d, r13d
0x161d9:      push     r12
0x161db:      lea      r12, [rip + 0xc203]
0x161e2:      push     rbp
0x161e3:      mov      rbp, rdi
0x161e6:      push     rbx
0x161e7:      mov      rbx, rsi
0x161ea:      sub      rsp, 8
0x161ee:      mov      rdx, qword ptr [rdi + 0x38]
0x161f2:      test     rdx, rdx
0x161f5:      je       0x1621a
0x161f7:      movsxd   rax, dword ptr [rdi + 0x28]
0x161fb:      lea      rax, [rax + rax*4]
0x161ff:      shl      rax, 4
0x16203:      mov      rdi, qword ptr [rdx + rax]
0x16207:      test     rdi, rdi
0x1620a:      je       0x1621a
0x1620c:      call     0x1d7a0
0x16211:      mov      r13d, 1
0x16217:      mov      r12, rax
0x1621a:      xor      edx, edx
0x1621c:      mov      rsi, qword ptr [rbp + 0x18]
0x16220:      cmp      rbx, -1
0x16224:      mov      rdi, rbp
0x16227:      sete     dl
0x1622a:      xor      eax, eax
0x1622c:      call     0xf880
0x16231:      test     eax, eax
0x16233:      je       0x16278

```

```

0x16235:      cmp     rbx, -1
0x16239:      je      0x16288
0x1623b:      mov     dword ptr [rbp + 0x28], 0
0x16242:      mov     rdi, rbp
0x16245:      xor     eax, eax
0x16247:      call    0xf110
0x1624c:      mov     dword ptr [rbp + 0x2c], eax
0x1624f:      test    r13d, r13d
0x16252:      jne     0x16260
0x16254:      add     rsp, 8
0x16258:      pop     rbx
0x16259:      pop     rbp
0x1625a:      pop     r12
0x1625c:      pop     r13
0x1625e:      ret
0x1625f:      nop
0x16260:      add     rsp, 8
0x16264:      mov     rdi, r12
0x16267:      pop     rbx
0x16268:      pop     rbp
0x16269:      pop     r12
0x1626b:      pop     r13
0x1626d:      jmp     0x18d10
0x18d10
0x16272:      nop     word ptr [rax + rax]
0x16278:      mov     rdi, rbp
0x1627b:      xor     eax, eax
0x1627d:      call    0x14d00
0x16282:      jmp     0x1624f
0x1624f
0x16284:      nop     dword ptr [rax]
0x16288:      mov     rsi, r12
0x1628b:      mov     rdi, rbp
0x1628e:      xor     eax, eax
0x16290:      call    0x122d0
0x16295:      mov     rdi, rbp
0x16298:      mov     dword ptr [rbp + 0x28], eax
0x1629b:      xor     eax, eax
0x1629d:      call    0xf110
0x162a2:      mov     dword ptr [rbp + 0x2c], eax
0x162a5:      jmp     0x1624f
0x1624f
0x162a7:      nop     word ptr [rax + rax]
0x162b0:      endbr64
0x162b4:      push    r15
0x162b6:      push    r14
0x162b8:      push    r13
0x162ba:      push    r12

```

```

0x162bc:    mov     r12, rdi
0x162bf:    push   rbp
0x162c0:    movsxd  rax, dword ptr [rdi + 0x28]
0x162c4:    mov     rbp, rsi
0x162c7:    lea     rax, [rax + rax*4]
0x162cb:    shl     rax, 4
0x162cf:    add     rax, qword ptr [rdi + 0x38]
0x162d3:    movzx   edx, byte ptr [rax + 0x48]
0x162d7:    mov     r14, qword ptr [rax]
0x162da:    test    dl, dl
0x162dc:    je      0x16300
0x162de:    cmp     dl, 1
0x162e1:    jne     0x162ed
0x162e3:    cmp     byte ptr [rax + 0x4a], 0
0x162e7:    jne     0x16420
0x162ed:    xor     r13d, r13d
0x162f0:    pop     rbp
0x162f1:    mov     eax, r13d
0x162f4:    pop     r12
0x162f6:    pop     r13
0x162f8:    pop     r14
0x162fa:    pop     r15
0x162fc:    ret
0x162fd:    nop     dword ptr [rax]
0x16300:    cmp     byte ptr [r14], 0x2e
0x16304:    jne     0x163c0
0x1630a:    cmp     byte ptr [r14 + 1], 0x2e
0x1630f:    jne     0x163c0
0x16315:    cmp     byte ptr [r14 + 2], 0
0x1631a:    jne     0x163c0
0x16320:    mov     rax, qword ptr [rdi + 0x18]
0x16324:    cmp     byte ptr [rax], 0x2f
0x16327:    jne     0x16330
0x16329:    cmp     byte ptr [rax + 1], 0
0x1632d:    je      0x162ed
0x1632f:    nop
0x16330:    mov     rax, qword ptr [r12 + 0x78]
0x16335:    lea     rdi, [rax + 1]
0x16339:    call    0x1d670
0x1633e:    mov     rsi, qword ptr [r12 + 0x18]
0x16343:    mov     rdi, rax
0x16346:    mov     r15, rax
0x16349:    call    0x48b0
0x1634e:    mov     edx, 1
0x16353:    mov     rsi, r14
0x16356:    mov     rdi, r12
0x16359:    xor     eax, eax
0x1635b:    call    0xf880

```

```

0x16360:      mov     r13d, eax
0x16363:      test    eax, eax
0x16365:      jne     0x165c0
0x1636b:      xor     eax, eax
0x1636d:      mov     rdi, r12
0x16370:      call    0x14d00
0x16375:      xor     eax, eax
0x16377:      mov     rdi, r12
0x1637a:      call    0x10ec0
0x1637f:      xor     eax, eax
0x16381:      mov     rdi, r12
0x16384:      call    0x10160
0x16389:      cmp     byte ptr [r12 + 0x6d], 0
0x1638f:      jne     0x165b0
0x16395:      xor     eax, eax
0x16397:      mov     rdi, r12
0x1639a:      call    0x10240
0x1639f:      cmp     byte ptr [r12 + 0x6d], 0
0x163a5:      je      0x162ed
0x163ab:      mov     rdi, r12
0x163ae:      xor     eax, eax
0x163b0:      call    0x10d70
0x163b5:      jmp     0x162f0
0x162f0:
0x163ba:      nop     word ptr [rax + rax]
0x163c0:      mov     rax, qword ptr [r12 + 0x78]
0x163c5:      lea     rdi, [rax + 1]
0x163c9:      call    0x1d670
0x163ce:      mov     rsi, qword ptr [r12 + 0x18]
0x163d3:      mov     rdi, rax
0x163d6:      mov     r15, rax
0x163d9:      call    0x48b0
0x163de:      mov     edx, 1
0x163e3:      mov     rsi, r14
0x163e6:      mov     rdi, r12
0x163e9:      xor     eax, eax
0x163eb:      call    0xf880
0x163f0:      mov     r13d, eax
0x163f3:      test    eax, eax
0x163f5:      jne     0x164b0
0x163fb:      mov     ecx, 0x1c
0x16400:      xor     edx, edx
0x16402:      mov     rsi, r14
0x16405:      xor     eax, eax
0x16407:      lea     rdi, [rip + 0xaf94]
0x1640e:      call    0xf760
0x16413:      jmp     0x162f0
0x162f0:

```

```

0x16418:      nop      dword ptr [rax + rax]
0x16420:      mov      rdi, r14
0x16423:      call     0x49b0
0x16428:      lea      rdi, [rax + 0x21]
0x1642c:      call     0x1d670
0x16431:      mov      r8, r14
0x16434:      mov      esi, 1
0x16439:      lea      rcx, [rip + 0xb04b]
0x16440:      mov      rdx, -1
0x16447:      mov      r13, rax
0x1644a:      mov      rdi, rax
0x1644d:      xor      eax, eax
0x1644f:      call     0x4f10
0x16454:      xor      esi, esi
0x16456:      mov      rdi, r13
0x16459:      call     0x193f0
0x1645e:      mov      rdi, r13
0x16461:      mov      r13d, 1
0x16467:      mov      dword ptr [rip + 0x1446b], 1
0x16471:      call     0x18d10
0x16476:      xor      eax, eax
0x16478:      call     0xa480
0x1647d:      mov      rdi, qword ptr [rip + 0x14444]
0x16484:      call     0xbb10
0x16489:      mov      rdi, r14
0x1648c:      mov      dword ptr [r12 + 0x14], 0x3fffffff
0x16495:      mov      dword ptr [rbp + 0x14], 0x3fffffff
0x1649c:      call     0xd610
0x164a1:      mov      rdi, qword ptr [r12 + 0x18]
0x164a6:      call     0xc4b0
0x164ab:      jmp      0x162f0
0x162f0
0x164b0:      mov      dword ptr [r12 + 0x28], 0
0x164b9:      mov      rdi, r12
0x164bc:      xor      eax, eax
0x164be:      call     0xf110
0x164c3:      mov      dword ptr [r12 + 0x2c], eax
0x164c8:      mov      rdi, r15
0x164cb:      call     0x18d10
0x164d0:      xor      eax, eax
0x164d2:      mov      rdi, r12
0x164d5:      call     0x10ec0
0x164da:      xor      eax, eax
0x164dc:      mov      rdi, r12
0x164df:      call     0x10160
0x164e4:      cmp      byte ptr [r12 + 0x6d], 0
0x164ea:      je       0x164f6
0x164ec:      mov      rdi, r12

```



```

0x164ef:    xor     eax, eax
0x164f1:    call   0xec50
0x164f6:    xor     eax, eax
0x164f8:    mov     rdi, r12
0x164fb:    call   0x10240
0x16500:    cmp     byte ptr [r12 + 0x6d], 0
0x16506:    je      0x16512
0x16508:    mov     rdi, r12
0x1650b:    xor     eax, eax
0x1650d:    call   0x10d70
0x16512:    mov     rsi, qword ptr [rbp + 0x18]
0x16516:    mov     rdi, qword ptr [r12 + 0x18]
0x1651b:    call   0x4b30
0x16520:    mov     r13d, eax
0x16523:    test    eax, eax
0x16525:    jne     0x16588
0x16527:    mov     r8d, 1
0x1652d:    mov     rdx, r12
0x16530:    mov     esi, 0x22
0x16535:    mov     rdi, rbp
0x16538:    mov     rcx, -1
0x1653f:    call   0x13e30
0x16544:    xor     eax, eax
0x16546:    mov     rdi, rbp
0x16549:    call   0x10ec0
0x1654e:    xor     eax, eax
0x16550:    mov     rdi, rbp
0x16553:    call   0x10160
0x16558:    cmp     byte ptr [rbp + 0x6d], 0
0x1655c:    jne     0x165a0
0x1655e:    xor     eax, eax
0x16560:    mov     rdi, rbp
0x16563:    call   0x10240
0x16568:    cmp     byte ptr [rbp + 0x6d], 0
0x1656c:    je      0x162ed
0x16572:    mov     rdi, rbp
0x16575:    xor     eax, eax
0x16577:    call   0x10d70
0x1657c:    jmp     0x162f0
0x162f0:
0x16581:    nop     dword ptr [rax]
0x16588:    mov     rdi, rbp
0x1658b:    xor     eax, eax
0x1658d:    call   0x10240
0x16592:    jmp     0x162ed
0x162ed:
0x16597:    nop     word ptr [rax + rax]
0x165a0:    mov     rdi, rbp

```

```

0x165a3:    xor     eax, eax
0x165a5:    call   0xec50
0x165aa:    jmp     0x1655e
0x1655e:
0x165ac:    nop     dword ptr [rax]
0x165b0:    mov     rdi, r12
0x165b3:    xor     eax, eax
0x165b5:    call   0xec50
0x165ba:    jmp     0x16395
0x16395:
0x165bf:    nop
0x165c0:    mov     rdi, r15
0x165c3:    mov     esi, 0x2f
0x165c8:    call   0x4a20
0x165cd:    mov     rdi, r12
0x165d0:    lea     rsi, [rax + 1]
0x165d4:    xor     eax, eax
0x165d6:    call   0x122d0
0x165db:    mov     rdi, r12
0x165de:    mov     dword ptr [r12 + 0x28], eax
0x165e3:    xor     eax, eax
0x165e5:    call   0xf110
0x165ea:    jmp     0x164c3
0x164c3:
0x165ef:    nop
0x165f0:    endbr64
0x165f4:    push    r13
0x165f6:    push    r12
0x165f8:    mov     r12, rdx
0x165fb:    push    rbp
0x165fc:    mov     rbp, rsi
0x165ff:    push    rbx
0x16600:    mov     rbx, rdi
0x16603:    mov     rdi, rdx
0x16606:    sub     rsp, 8
0x1660a:    call   0x49b0
0x1660f:    cmp     byte ptr [r12], 0x2f
0x16614:    je      0x166f0
0x1661a:    add     rax, qword ptr [rbx + 0x78]
0x1661e:    mov     rdi, qword ptr [rbx + 0x18]
0x16622:    lea     rdx, [rax + 1]
0x16626:    lea     rsi, [rax + 2]
0x1662a:    mov     qword ptr [rbx + 0x78], rdx
0x1662e:    call   0x1d690
0x16633:    mov     qword ptr [rbx + 0x18], rax
0x16637:    mov     rdi, rax
0x1663a:    mov     r13, rax
0x1663d:    call   0x49b0

```

```

0x16642:    mov     edx, 0x2f
0x16647:    mov     rsi, r12
0x1664a:    mov     word ptr [r13 + rax], dx
0x16650:    mov     rdi, qword ptr [rbx + 0x18]
0x16654:    call    0x4df0
0x16659:    mov     rdi, qword ptr [rbx + 0x18]
0x1665d:    call    0x18040
0x16662:    mov     dword ptr [rbx + 0x28], 0
0x16669:    mov     rdi, rbx
0x1666c:    xor     eax, eax
0x1666e:    mov     r8d, 1
0x16674:    call    0xf110
0x16679:    xor     ecx, ecx
0x1667b:    xor     edx, edx
0x1667d:    mov     esi, 0x22
0x16682:    mov     dword ptr [rbx + 0x2c], eax
0x16685:    mov     rdi, rbx
0x16688:    call    0x13e30
0x1668d:    xor     eax, eax
0x1668f:    mov     rdi, rbx
0x16692:    call    0x10ec0
0x16697:    xor     eax, eax
0x16699:    mov     rdi, rbx
0x1669c:    call    0x10160
0x166a1:    cmp     byte ptr [rbx + 0x6d], 0
0x166a5:    je      0x166b1
0x166a7:    mov     rdi, rbx
0x166aa:    xor     eax, eax
0x166ac:    call    0xec50
0x166b1:    xor     eax, eax
0x166b3:    mov     rdi, rbx
0x166b6:    call    0x10240
0x166bb:    cmp     byte ptr [rbx + 0x6d], 0
0x166bf:    je      0x166cb
0x166c1:    mov     rdi, rbx
0x166c4:    xor     eax, eax
0x166c6:    call    0x10d70
0x166cb:    mov     rsi, qword ptr [rbp + 0x18]
0x166cf:    mov     rdi, qword ptr [rbx + 0x18]
0x166d3:    call    0x4b30
0x166d8:    test    eax, eax
0x166da:    je      0x16718
0x166dc:    add     rsp, 8
0x166e0:    pop     rbx
0x166e1:    pop     rbp
0x166e2:    pop     r12
0x166e4:    pop     r13
0x166e6:    ret

```

```

0x166e7:      nop      word ptr [rax + rax]
0x166f0:      mov      qword ptr [rbx + 0x78], rax
0x166f4:      mov      rdi, qword ptr [rbx + 0x18]
0x166f8:      lea      rsi, [rax + 1]
0x166fc:      call     0x1d690
0x16701:      mov      rsi, r12
0x16704:      mov      qword ptr [rbx + 0x18], rax
0x16708:      mov      rdi, rax
0x1670b:      call     0x48b0
0x16710:      jmp      0x16662
0x16662
0x16715:      nop      dword ptr [rax]
0x16718:      xor      edx, edx
0x1671a:      mov      r8d, 1
0x16720:      mov      esi, 0x22
0x16725:      mov      rdi, rbp
0x16728:      mov      rcx, -1
0x1672f:      call     0x13e30
0x16734:      xor      eax, eax
0x16736:      mov      rdi, rbp
0x16739:      call     0x10ec0
0x1673e:      xor      eax, eax
0x16740:      mov      rdi, rbp
0x16743:      call     0x10160
0x16748:      cmp      byte ptr [rbp + 0x6d], 0
0x1674c:      jne      0x16780
0x1674e:      xor      eax, eax
0x16750:      mov      rdi, rbp
0x16753:      call     0x10240
0x16758:      cmp      byte ptr [rbp + 0x6d], 0
0x1675c:      je       0x166dc
0x16762:      add      rsp, 8
0x16766:      mov      rdi, rbp
0x16769:      xor      eax, eax
0x1676b:      pop      rbx
0x1676c:      pop      rbp
0x1676d:      pop      r12
0x1676f:      pop      r13
0x16771:      jmp      0x10d70
0x10d70
0x16776:      nop      word ptr cs:[rax + rax]
0x16780:      xor      eax, eax
0x16782:      mov      rdi, rbp
0x16785:      call     0xec50
0x1678a:      xor      eax, eax
0x1678c:      mov      rdi, rbp
0x1678f:      call     0x10240
0x16794:      cmp      byte ptr [rbp + 0x6d], 0

```

```

0x16798:      je      0x166dc
0x1679e:      jmp     0x16762
0x16762:
0x167a0:      endbr64
0x167a4:      push    r15
0x167a6:      lea     r15, [rip + 0x13bd3]
0x167ad:      push    r14
0x167af:      lea     r14, [rip + 0xacdc]
0x167b6:      push    r13
0x167b8:      mov     r13, rsi
0x167bb:      push    r12
0x167bd:      mov     r12d, edx
0x167c0:      push    rbp
0x167c1:      mov     rbp, rdi
0x167c4:      push    rbx
0x167c5:      sub     rsp, 8
0x167c9:      mov     rdi, qword ptr [rip + 0x13d08]
0x167d0:      call    0xdb60
0x167d5:      xor     eax, eax
0x167d7:      call    0xa820
0x167dc:      mov     eax, dword ptr [rbp + 0x70]
0x167df:      mov     dword ptr [rbp + 0x50], 0xffffffff
0x167e6:      mov     dword ptr [rbp + 0x90], 1
0x167f0:      mov     dword ptr [rbp + 0x54], eax
0x167f3:      nop     dword ptr [rax + rax]
0x167f8:      mov     rdi, rbp
0x167fb:      call    0x100e0
0x16800:      movsxd  rbx, eax
0x16803:      cmp     ebx, -1
0x16806:      je      0x1686b
0x16808:      xor     eax, eax
0x1680a:      call    0x18ab0
0x1680f:      xor     eax, eax
0x16811:      mov     edx, r12d
0x16814:      mov     esi, ebx
0x16816:      mov     rdi, rbp
0x16819:      call    0x134d0
0x1681e:      cmp     eax, 1
0x16821:      je      0x168f0
0x16827:      cmp     eax, 2
0x1682a:      je      0x1686b
0x1682c:      cmp     eax, 3
0x1682f:      je      0x167f8
0x16831:      sub     eax, 1
0x16834:      xor     ecx, ecx
0x16836:      mov     rdi, r14
0x16839:      mov     r8d, 0x1e
0x1683f:      cdqe

```

```

0x16841:    mov     rdx, qword ptr [r15 + rax*8]
0x16845:    lea     rax, [rbx + rbx*4]
0x16849:    shl     rax, 4
0x1684d:    add     rax, qword ptr [rbp + 0x38]
0x16851:    mov     rsi, qword ptr [rax]
0x16854:    xor     eax, eax
0x16856:    call    0xf7e0
0x1685b:    mov     rdi, rbp
0x1685e:    call    0x100e0
0x16863:    movsxd  rbx, eax
0x16866:    cmp     ebx, -1
0x16869:    jne     0x16808
0x1686b:    xor     eax, eax
0x1686d:    call    0xe4e0
0x16872:    xor     eax, eax
0x16874:    call    0xa820
0x16879:    xor     ecx, ecx
0x1687b:    xor     edx, edx
0x1687d:    mov     r8d, 1
0x16883:    mov     esi, 0x22
0x16888:    mov     rdi, rbp
0x1688b:    call    0x13e30
0x16890:    xor     eax, eax
0x16892:    mov     rdi, rbp
0x16895:    call    0x10ec0
0x1689a:    xor     eax, eax
0x1689c:    mov     rdi, rbp
0x1689f:    call    0x10160
0x168a4:    cmp     byte ptr [rbp + 0x6d], 0
0x168a8:    je      0x168b4
0x168aa:    mov     rdi, rbp
0x168ad:    xor     eax, eax
0x168af:    call    0xec50
0x168b4:    xor     eax, eax
0x168b6:    mov     rdi, rbp
0x168b9:    call    0x10240
0x168be:    cmp     byte ptr [rbp + 0x6d], 0
0x168c2:    je      0x168ce
0x168c4:    mov     rdi, rbp
0x168c7:    xor     eax, eax
0x168c9:    call    0x10d70
0x168ce:    mov     rsi, qword ptr [r13 + 0x18]
0x168d2:    mov     rdi, qword ptr [rbp + 0x18]
0x168d6:    call    0x4b30
0x168db:    test    eax, eax
0x168dd:    je      0x16908
0x168df:    add     rsp, 8
0x168e3:    pop     rbx

```

```

0x168e4:      pop     rbp
0x168e5:      pop     r12
0x168e7:      pop     r13
0x168e9:      pop     r14
0x168eb:      pop     r15
0x168ed:      ret
0x168ee:      nop
0x168f0:      lea     rax, [rbx + rbx*4]
0x168f4:      shl     rax, 4
0x168f8:      add     rax, qword ptr [rbp + 0x38]
0x168fc:      mov     byte ptr [rax + 0x49], 0
0x16900:      jmp     0x167f8
0x167f8
0x16905:      nop     dword ptr [rax]
0x16908:      xor     ecx, ecx
0x1690a:      xor     edx, edx
0x1690c:      mov     r8d, 1
0x16912:      mov     esi, 0x22
0x16917:      mov     rdi, r13
0x1691a:      call    0x13e30
0x1691f:      xor     eax, eax
0x16921:      mov     rdi, r13
0x16924:      call    0x10ec0
0x16929:      xor     eax, eax
0x1692b:      mov     rdi, r13
0x1692e:      call    0x10160
0x16933:      cmp     byte ptr [r13 + 0x6d], 0
0x16938:      jne     0x16968
0x1693a:      xor     eax, eax
0x1693c:      mov     rdi, r13
0x1693f:      call    0x10240
0x16944:      cmp     byte ptr [r13 + 0x6d], 0
0x16949:      je      0x168df
0x1694b:      add     rsp, 8
0x1694f:      mov     rdi, r13
0x16952:      xor     eax, eax
0x16954:      pop     rbx
0x16955:      pop     rbp
0x16956:      pop     r12
0x16958:      pop     r13
0x1695a:      pop     r14
0x1695c:      pop     r15
0x1695e:      jmp     0x10d70
0x10d70
0x16963:      nop     dword ptr [rax + rax]
0x16968:      mov     rdi, r13
0x1696b:      xor     eax, eax
0x1696d:      call    0xec50

```

```

0x16972:      jmp      0x1693a
0x1693a
0x16974:      nop      word ptr cs:[rax + rax]
0x1697e:      nop
0x16980:      push     r15
0x16982:      push     r14
0x16984:      push     r13
0x16986:      push     r12
0x16988:      push     rbp
0x16989:      push     rbx
0x1698a:      sub      rsp, 0x8c8
0x16991:      mov      rdi, qword ptr [rip + 0x14ca8]
0x16998:      mov      r13, qword ptr [rip + 0x13cc1]
0x1699f:      mov      rax, qword ptr fs:[0x28]
0x169a8:      mov      qword ptr [rsp + 0x8b8], rax
0x169b0:      xor      eax, eax
0x169b2:      lea      rax, [rip + 0x14c8f]
0x169b9:      mov      qword ptr [rip + 0x13ca0], rax
0x169c0:      test     rdi, rdi
0x169c3:      je       0x16a78
0x169c9:      lea      rsi, [rsp + 0x20]
0x169ce:      call     0x18f40
0x169d3:      cmp      eax, -1
0x169d6:      je       0x16a78
0x169dc:      mov      rax, qword ptr [rsp + 0x68]
0x169e1:      mov      rdi, qword ptr [rip + 0x14c58]
0x169e8:      lea      rsi, [rip + 0xb5da]
0x169ef:      mov      qword ptr [rsp + 0x10], rax
0x169f4:      mov      rax, qword ptr [rsp + 0x78]
0x169f9:      mov      qword ptr [rsp + 0x18], rax
0x169fe:      call     0x4da0
0x16a03:      mov      r14, rax
0x16a06:      test     rax, rax
0x16a09:      je       0x16a78
0x16a0b:      mov      dword ptr [rsp + 4], 0
0x16a13:      xor      ebp, ebp
0x16a15:      xor      ebx, ebx
0x16a17:      xor      r12d, r12d
0x16a1a:      lea      r15, [rsp + 0xb0]
0x16a22:      nop      word ptr [rax + rax]
0x16a28:      mov      rdx, r14
0x16a2b:      mov      esi, 0x800
0x16a30:      mov      rdi, r15
0x16a33:      call     0x4b00
0x16a38:      test     rax, rax
0x16a3b:      je       0x16ad0
0x16a41:      cmp      byte ptr [rsp + 0xb0], 0
0x16a49:      je       0x16aa8

```



```

0x16a4b:    cmp     dword ptr [r15], 0x6d6f7246
0x16a52:    je      0x16ab0
0x16a54:    test    ebp, ebp
0x16a56:    jne     0x16a28
0x16a58:    test    ebx, ebx
0x16a5a:    je      0x16a28
0x16a5c:    cmp     dword ptr [r15], 0x74617453
0x16a63:    je      0x16b60
0x16a69:    xor     ebp, ebp
0x16a6b:    mov     ebx, 1
0x16a70:    jmp     0x16a28
0x16a28:
0x16a72:    nop     word ptr [rax + rax]
0x16a78:    xor     eax, eax
0x16a7a:    mov     rdx, qword ptr [rsp + 0x8b8]
0x16a82:    sub     rdx, qword ptr fs:[0x28]
0x16a8b:    jne     0x16c02
0x16a91:    add     rsp, 0x8c8
0x16a98:    pop     rbx
0x16a99:    pop     rbp
0x16a9a:    pop     r12
0x16a9c:    pop     r13
0x16a9e:    pop     r14
0x16aa0:    pop     r15
0x16aa2:    ret
0x16aa3:    nop     dword ptr [rax + rax]
0x16aa8:    xor     ebx, ebx
0x16aaa:    jmp     0x16a28
0x16a28:
0x16aaf:    nop
0x16ab0:    cmp     byte ptr [r15 + 4], 0x20
0x16ab5:    jne     0x16a54
0x16ab7:    add     r12d, 1
0x16abb:    xor     ebp, ebp
0x16abd:    mov     ebx, 1
0x16ac2:    jmp     0x16a28
0x16a28:
0x16ac7:    nop     word ptr [rax + rax]
0x16ad0:    mov     rdi, r14
0x16ad3:    call    0x4940
0x16ad8:    mov     rdi, qword ptr [rip + 0x14b61]
0x16adf:    lea     rsi, [rsp + 0x10]
0x16ae4:    call    0x47f0
0x16ae9:    mov     rbp, qword ptr [rip + 0x13b70]
0x16af0:    test    r12d, r12d
0x16af3:    je      0x16b0a
0x16af5:    cmp     r12d, dword ptr [rsp + 4]
0x16afa:    je      0x16b48

```

```

0x16afc:    lea     rbp, [rip + 0x13b1d]
0x16b03:    mov     qword ptr [rip + 0x13b56], rbp
0x16b0a:    mov     rdi, rbp
0x16b0d:    call    0x49b0
0x16b12:    mov     rsi, r13
0x16b15:    mov     rdi, rbp
0x16b18:    mov     r8, rax
0x16b1b:    mov     eax, dword ptr [rip + 0x14b2f]
0x16b21:    add     eax, dword ptr [rip + 0x14b2d]
0x16b27:    lea     eax, [rax + r8 + 0x15]
0x16b2c:    mov     dword ptr [rip + 0x14b1a], eax
0x16b32:    call    0x4b30
0x16b37:    test    eax, eax
0x16b39:    setne   al
0x16b3c:    movzx   eax, al
0x16b3f:    jmp     0x16a7a
0x16a7a:
0x16b44:    nop     dword ptr [rax]
0x16b48:    lea     rbp, [rip + 0x13adc]
0x16b4f:    mov     qword ptr [rip + 0x13b0a], rbp
0x16b56:    jmp     0x16b0a
0x16b0a:
0x16b58:    nop     dword ptr [rax + rax]
0x16b60:    cmp     word ptr [r15 + 4], 0x7375
0x16b67:    jne     0x16a69
0x16b6d:    cmp     byte ptr [r15 + 6], 0x3a
0x16b72:    jne     0x16a69
0x16b78:    mov     rdi, r15
0x16b7b:    mov     esi, 0x3a
0x16b80:    call    0x4a00
0x16b85:    movsx   rbx, byte ptr [rax + 1]
0x16b8a:    lea     rdi, [rax + 1]
0x16b8e:    test    bl, bl
0x16b90:    je      0x16bf8
0x16b92:    mov     qword ptr [rsp + 8], rdi
0x16b97:    call    0x4f00
0x16b9c:    mov     rdi, qword ptr [rsp + 8]
0x16ba1:    mov     rax, qword ptr [rax]
0x16ba4:    jmp     0x16bbd
0x16bbd:
0x16ba6:    nop     word ptr cs:[rax + rax]
0x16bb0:    movsx   rbx, byte ptr [rdi + 1]
0x16bb5:    add     rdi, 1
0x16bb9:    test    bl, bl
0x16bbb:    je      0x16bf8
0x16bbd:    test    byte ptr [rax + rbx*2 + 1], 0x20
0x16bc2:    jne     0x16bb0
0x16bc4:    mov     esi, 0x52

```

```

0x16bc9:      mov     ebp, 1
0x16bce:      mov     ebx, 1
0x16bd3:      call    0x4a00
0x16bd8:      test    rax, rax
0x16bdb:      je      0x16a28
0x16be1:      add     dword ptr [rsp + 4], 1
0x16be6:      mov     ebp, 1
0x16beb:      mov     ebx, 1
0x16bf0:      jmp     0x16a28
0x16a28
0x16bf5:      nop     dword ptr [rax]
0x16bf8:      mov     ebx, 1
0x16bfd:      jmp     0x16a28
0x16a28
0x16c02:      call    0x49d0
0x16c07:      nop     word ptr [rax + rax]
0x16c10:      endbr64
0x16c14:      sub     rsp, 0xa8
0x16c1b:      mov     edi, 0x1d
0x16c20:      mov     rax, qword ptr fs:[0x28]
0x16c29:      mov     qword ptr [rsp + 0x98], rax
0x16c31:      xor     eax, eax
0x16c33:      call    0x1d670
0x16c38:      lea     r9, [rip + 0x9950]
0x16c3f:      lea     r8, [rip + 0x991f]
0x16c46:      mov     rdx, -1
0x16c4d:      lea     rcx, [rip + 0xb219]
0x16c54:      mov     esi, 1
0x16c59:      mov     rdi, rax
0x16c5c:      mov     qword ptr [rip + 0x149fd], rax
0x16c63:      xor     eax, eax
0x16c65:      call    0x4f10
0x16c6a:      mov     rdi, qword ptr [rip + 0x149ef]
0x16c71:      call    0x49b0
0x16c76:      mov     rdi, qword ptr [rip + 0x14e1b]
0x16c7d:      mov     dword ptr [rip + 0x149d5], eax
0x16c83:      call    0x49b0
0x16c88:      mov     rdi, qword ptr [rip + 0x13d01]
0x16c8f:      mov     dword ptr [rip + 0x149bf], eax
0x16c95:      call    0x49b0
0x16c9a:      lea     rdi, [rip + 0x9999]
0x16ca1:      mov     dword ptr [rip + 0x149a9], eax
0x16ca7:      call    0x47e0
0x16cac:      mov     qword ptr [rip + 0x1498d], rax
0x16cb3:      test    rax, rax
0x16cb6:      je      0x16cc3
0x16cb8:      mov     rdi, rax
0x16cbb:      mov     rsi, rsp

```

```

0x16cbe:      call    0x18f40
0x16cc3:      xor     eax, eax
0x16cc5:      call    0x16980
0x16cca:      mov     rdi, qword ptr [rip + 0x1398f]
0x16cd1:      call    0x49b0
0x16cd6:      mov     r8, rax
0x16cd9:      mov     eax, dword ptr [rip + 0x14971]
0x16cdf:      add     eax, dword ptr [rip + 0x1496f]
0x16ce5:      lea     eax, [rax + r8 + 0x15]
0x16cea:      mov     dword ptr [rip + 0x1495c], eax
0x16cf0:      xor     eax, eax
0x16cf2:      call    0xc590
0x16cf7:      mov     qword ptr [rip + 0x1493a], rax
0x16cfe:      mov     rax, qword ptr [rsp + 0x98]
0x16d06:      sub     rax, qword ptr fs:[0x28]
0x16d0f:      jne     0x16d19
0x16d11:      add     rsp, 0xa8
0x16d18:      ret
0x16d19:      call    0x49d0
0x16d1e:      nop
0x16d20:      endbr64
0x16d24:      mov     rdi, qword ptr [rip + 0x1490d]
0x16d2b:      jmp     0xc5c0
0xc5c0
0x16d30:      endbr64
0x16d34:      mov     r8d, edi
0x16d37:      mov     rdi, qword ptr [rip + 0x148fa]
0x16d3e:      mov     edx, esi
0x16d40:      mov     ecx, 1
0x16d45:      xor     esi, esi
0x16d47:      jmp     0xc5e0
0xc5e0
0x16d4c:      nop     dword ptr [rax]
0x16d50:      endbr64
0x16d54:      push    r14
0x16d56:      push    r13
0x16d58:      push    r12
0x16d5a:      push    rbp
0x16d5b:      push    rbx
0x16d5c:      sub     rsp, 0x10
0x16d60:      mov     rax, qword ptr fs:[0x28]
0x16d69:      mov     qword ptr [rsp + 8], rax
0x16d6e:      xor     eax, eax
0x16d70:      lea     rbp, [rsp + 7]
0x16d75:      mov     rdi, rbp
0x16d78:      call    0xa6a0
0x16d7d:      mov     edx, dword ptr [rip + 0x138c1]
0x16d83:      mov     esi, dword ptr [rip + 0x138b7]

```

```

0x16d89:    mov     edi, dword ptr [rip + 0x138b9]
0x16d8f:    call   0xa570
0x16d94:    mov     rdi, qword ptr [rip + 0x1489d]
0x16d9b:    xor     edx, edx
0x16d9d:    xor     esi, esi
0x16d9f:    call   0xc680
0x16da4:    mov     edx, dword ptr [rip + 0x148ae]
0x16daa:    mov     rsi, qword ptr [rip + 0x148af]
0x16db1:    mov     rdi, qword ptr [rip + 0x14880]
0x16db8:    call   0xc600
0x16dbd:    mov     rax, qword ptr [rip + 0x14874]
0x16dc4:    mov     edi, dword ptr [rax + 0xc]
0x16dc7:    add     edi, 1
0x16dca:    movsxd  rdi, edi
0x16dcd:    call   0x1d670
0x16dd2:    mov     r13, rax
0x16dd5:    xor     eax, eax
0x16dd7:    call   0x16980
0x16ddc:    mov     r14, qword ptr [rip + 0x14855]
0x16de3:    mov     eax, dword ptr [rip + 0x1486f]
0x16de9:    mov     ecx, dword ptr [rip + 0x1485d]
0x16def:    mov     edx, dword ptr [r14 + 0xc]
0x16df3:    lea     esi, [rax + rcx + 2]
0x16df7:    cmp     esi, edx
0x16df9:    jl      0x16e58
0x16dfb:    cmp     eax, edx
0x16dfd:    jl      0x16e30
0x16dff:    mov     rdi, r13
0x16e02:    call   0x18d10
0x16e07:    mov     rdi, rbp
0x16e0a:    call   0xa6b0
0x16e0f:    mov     rax, qword ptr [rsp + 8]
0x16e14:    sub     rax, qword ptr fs:[0x28]
0x16e1d:    jne     0x16fc7
0x16e23:    add     rsp, 0x10
0x16e27:    pop     rbx
0x16e28:    pop     rbp
0x16e29:    pop     r12
0x16e2b:    pop     r13
0x16e2d:    pop     r14
0x16e2f:    ret
0x16e30:    sub     edx, eax
0x16e32:    mov     rdi, r13
0x16e35:    mov     esi, 0x20
0x16e3a:    mov     r12d, edx
0x16e3d:    movsxd  rdx, edx
0x16e40:    call   0x4a50
0x16e45:    mov     edx, r12d

```

```

0x16e48:      mov     rsi, r13
0x16e4b:      mov     rdi, r14
0x16e4e:      call    0xc600
0x16e53:      jmp     0x16dff
0x16dff:
0x16e55:      nop     dword ptr [rax]
0x16e58:      sub     edx, eax
0x16e5a:      mov     rdi, r13
0x16e5d:      mov     esi, 0x20
0x16e62:      mov     ebx, edx
0x16e64:      sub     ebx, ecx
0x16e66:      movsxd  rdx, ebx
0x16e69:      call    0x4a50
0x16e6e:      mov     edx, ebx
0x16e70:      mov     rsi, r13
0x16e73:      mov     rdi, r14
0x16e76:      call    0xc600
0x16e7b:      add     ebx, dword ptr [rip + 0x147d7]
0x16e81:      mov     rdi, qword ptr [rip + 0x147b0]
0x16e88:      xor     esi, esi
0x16e8a:      mov     edx, ebx
0x16e8c:      call    0xc680
0x16e91:      mov     rdi, qword ptr [rip + 0x147a0]
0x16e98:      mov     edx, 5
0x16e9d:      lea     rsi, [rip + 0x13793]
0x16ea4:      call    0xc600
0x16ea9:      mov     rdi, qword ptr [rip + 0x14788]
0x16eb0:      mov     esi, 0x20
0x16eb5:      call    0xc650
0x16eba:      mov     edi, dword ptr [rip + 0x1378c]
0x16ec0:      call    0xa4f0
0x16ec5:      mov     edx, dword ptr [rip + 0x14789]
0x16ecb:      mov     rsi, qword ptr [rip + 0x14bc6]
0x16ed2:      mov     rdi, qword ptr [rip + 0x1475f]
0x16ed9:      call    0xc600
0x16ede:      mov     rdi, qword ptr [rip + 0x14753]
0x16ee5:      mov     esi, 0x20
0x16eea:      call    0xc650
0x16eef:      mov     r12, qword ptr [rip + 0x1376a]
0x16ef6:      mov     rdi, r12
0x16ef9:      call    0x49b0
0x16efe:      mov     rdi, qword ptr [rip + 0x14733]
0x16f05:      mov     rsi, r12
0x16f08:      mov     edx, eax
0x16f0a:      call    0xc600
0x16f0f:      mov     rdi, qword ptr [rip + 0x14722]
0x16f16:      mov     esi, 0x20
0x16f1b:      call    0xc650

```

```

0x16f20:    mov     edi, dword ptr [rip + 0x1371a]
0x16f26:    call    0xa4f0
0x16f2b:    mov     rdi, qword ptr [rip + 0x14706]
0x16f32:    mov     edx, 4
0x16f37:    lea     rsi, [rip + 0x136f4]
0x16f3e:    call    0xc600
0x16f43:    mov     rdi, qword ptr [rip + 0x146ee]
0x16f4a:    mov     esi, 0x20
0x16f4f:    call    0xc650
0x16f54:    mov     edi, dword ptr [rip + 0x136f6]
0x16f5a:    call    0xa4f0
0x16f5f:    mov     edx, dword ptr [rip + 0x146eb]
0x16f65:    mov     rsi, qword ptr [rip + 0x13a24]
0x16f6c:    mov     rdi, qword ptr [rip + 0x146c5]
0x16f73:    call    0xc600
0x16f78:    mov     edi, dword ptr [rip + 0x136c2]
0x16f7e:    call    0xa4f0
0x16f83:    mov     rdi, qword ptr [rip + 0x146ae]
0x16f8a:    mov     esi, 0x20
0x16f8f:    call    0xc650
0x16f94:    xor     edi, edi
0x16f96:    xor     eax, eax
0x16f98:    call    0x16fd0
0x16f9d:    mov     rdi, qword ptr [rip + 0x14694]
0x16fa4:    xor     esi, esi
0x16fa6:    mov     eax, dword ptr [rdi + 0xc]
0x16fa9:    lea     edx, [rax - 1]
0x16fac:    call    0xc680
0x16fb1:    mov     rdi, qword ptr [rip + 0x14680]
0x16fb8:    mov     esi, 0x20
0x16fbd:    call    0xc650
0x16fc2:    jmp     0x16dff
0x16dff:
0x16fc7:    call    0x49d0
0x16fcc:    nop     dword ptr [rax]
0x16fd0:    endbr64
0x16fd4:    push    r12
0x16fd6:    push    rbp
0x16fd7:    push    rbx
0x16fd8:    mov     ebx, edi
0x16fda:    sub     rsp, 0x30
0x16fde:    mov     rax, qword ptr fs:[0x28]
0x16fe7:    mov     qword ptr [rsp + 0x28], rax
0x16fec:    xor     eax, eax
0x16fee:    call    0x7870
0x16ff3:    test    eax, eax
0x16ff5:    jne     0x17114
0x16ffb:    mov     eax, dword ptr [rip + 0x1464b]

```

```

0x17001:    mov     edx, dword ptr [rip + 0x14651]
0x17007:    lea     edx, [rdx + rax + 2]
0x1700b:    mov     rax, qword ptr [rip + 0x14626]
0x17012:    cmp     edx, dword ptr [rax + 0xc]
0x17015:    jge     0x17114
0x1701b:    test    ebx, ebx
0x1701d:    jne     0x17140
0x17023:    lea     r12, [rsp + 7]
0x17028:    xor     eax, eax
0x1702a:    call    0x18330
0x1702f:    mov     rdi, r12
0x17032:    mov     rbp, rax
0x17035:    call    0xa6a0
0x1703a:    lea     rsi, [rsp + 0xc]
0x1703f:    lea     rdi, [rsp + 8]
0x17044:    call    0xa4d0
0x17049:    xor     edi, edi
0x1704b:    call    0xa660
0x17050:    movsxd  r8, dword ptr [rbp + 8]
0x17054:    mov     r9d, dword ptr [rbp + 4]
0x17058:    lea     rbp, [rsp + 0x10]
0x1705d:    lea     rcx, [rip + 0xa0d4]
0x17064:    mov     esi, 1
0x17069:    mov     rdi, rbp
0x1706c:    mov     rax, r8
0x1706f:    imul    r8, r8, 0x2aaaaaab
0x17076:    cdq
0x17077:    sar     r8, 0x21
0x1707b:    sub     r8d, edx
0x1707e:    lea     edx, [r8 + r8*2]
0x17082:    mov     r8d, eax
0x17085:    shl     edx, 2
0x17088:    sub     r8d, edx
0x1708b:    mov     edx, 0xc
0x17090:    cmov    r8d, edx
0x17094:    cmp     eax, 0xc
0x17097:    mov     edx, 0x70
0x1709c:    mov     eax, 0x61
0x170a1:    cmovge  eax, edx
0x170a4:    sub     rsp, 8
0x170a8:    mov     edx, 0x10
0x170ad:    push    rax
0x170ae:    xor     eax, eax
0x170b0:    call    0x4f10
0x170b5:    mov     rdi, qword ptr [rip + 0x1457c]
0x170bc:    xor     esi, esi
0x170be:    mov     eax, dword ptr [rdi + 0xc]
0x170c1:    lea     edx, [rax - 7]

```



```

0x170c4:      call    0xc680
0x170c9:      mov     edx, dword ptr [rip + 0x13589]
0x170cf:      mov     esi, dword ptr [rip + 0x1357f]
0x170d5:      mov     edi, dword ptr [rip + 0x13581]
0x170db:      call    0xa570
0x170e0:      mov     rdi, rbp
0x170e3:      call    0x49b0
0x170e8:      mov     rdi, qword ptr [rip + 0x14549]
0x170ef:      mov     rsi, rbp
0x170f2:      mov     edx, eax
0x170f4:      call    0xc600
0x170f9:      mov     esi, dword ptr [rsp + 0x1c]
0x170fd:      mov     edi, dword ptr [rsp + 0x18]
0x17101:      call    0xa4b0
0x17106:      mov     rdi, r12
0x17109:      call    0xa6b0
0x1710e:      pop     rax
0x1710f:      pop     rdx
0x17110:      test    ebx, ebx
0x17112:      jne     0x17130
0x17114:      mov     rax, qword ptr [rsp + 0x28]
0x17119:      sub     rax, qword ptr fs:[0x28]
0x17122:      jne     0x1715b
0x17124:      add     rsp, 0x30
0x17128:      pop     rbx
0x17129:      pop     rbp
0x1712a:      pop     r12
0x1712c:      ret
0x1712d:      nop     dword ptr [rax]
0x17130:      xor     eax, eax
0x17132:      call    0xa820
0x17137:      jmp     0x17114
0x17114:
0x17139:      nop     dword ptr [rax]
0x17140:      xor     eax, eax
0x17142:      call    0x16980
0x17147:      test    eax, eax
0x17149:      je      0x17023
0x1714f:      xor     eax, eax
0x17151:      call    0x16d50
0x17156:      jmp     0x17023
0x17023:
0x1715b:      call    0x49d0
0x17160:      push    rbp
0x17161:      mov     esi, 0x400
0x17166:      push    rbx
0x17167:      lea     rbx, [rip + 0x14512]
0x1716e:      mov     rdi, rbx

```

```

0x17171:    sub     rsp, 8
0x17175:    mov     rdx, qword ptr [rip + 0x1490c]
0x1717c:    call    0x4b00
0x17181:    test    rax, rax
0x17184:    je      0x171d8
0x17186:    mov     rdi, rbx
0x17189:    call    0x49b0
0x1718e:    mov     rbp, rax
0x17191:    cmp     rax, 0x3ff
0x17197:    je      0x171e8
0x17199:    mov     esi, 0x23
0x1719e:    mov     rdi, rbx
0x171a1:    call    0x4a00
0x171a6:    test    rax, rax
0x171a9:    je      0x171c0
0x171ab:    mov     byte ptr [rax], 0
0x171ae:    mov     eax, 1
0x171b3:    add     rsp, 8
0x171b7:    pop     rbx
0x171b8:    pop     rbp
0x171b9:    ret
0x171ba:    nop     word ptr [rax + rax]
0x171c0:    cmp     byte ptr [rbx + rbp - 1], 0xa
0x171c5:    mov     eax, 1
0x171ca:    jne     0x171b3
0x171cc:    mov     byte ptr [rbx + rbp - 1], 0
0x171d1:    jmp     0x171b3
0x171b3:
0x171d3:    nop     dword ptr [rax + rax]
0x171d8:    add     rsp, 8
0x171dc:    xor     eax, eax
0x171de:    pop     rbx
0x171df:    pop     rbp
0x171e0:    ret
0x171e1:    nop     dword ptr [rax]
0x171e8:    mov     rcx, qword ptr [rip + 0x136f9]
0x171ef:    mov     rdi, qword ptr [rip + 0x1368a]
0x171f6:    lea     rdx, [rip + 0xaceb]
0x171fd:    xor     eax, eax
0x171ff:    mov     esi, 1
0x17204:    call    0x4e60
0x17209:    nop     dword ptr [rax]
0x17210:    mov     rdi, qword ptr [rip + 0x14871]
0x17217:    call    0x4a90
0x1721c:    cmp     eax, 0xa
0x1721f:    je      0x17199
0x17225:    cmp     eax, -1
0x17228:    jne     0x17210

```

```

0x1722a:      jmp      0x17199
0x17199
0x1722f:      nop
0x17230:      endbr64
0x17234:      test     rdi, rdi
0x17237:      je       0x17270
0x17239:      sub      rsp, 8
0x1723d:      lea      rsi, [rip + 0xad85]
0x17244:      call     0x4da0
0x17249:      mov      qword ptr [rip + 0x14838], rax
0x17250:      test     rax, rax
0x17253:      je       0x17269
0x17255:      mov      dword ptr [rip + 0x14821], 0xffffffff
0x1725f:      mov      eax, 1
0x17264:      add      rsp, 8
0x17268:      ret
0x17269:      xor      eax, eax
0x1726b:      add      rsp, 8
0x1726f:      ret
0x17270:      xor      eax, eax
0x17272:      ret
0x17273:      nop      word ptr cs:[rax + rax]
0x1727e:      nop
0x17280:      endbr64
0x17284:      mov      rdi, qword ptr [rip + 0x147fd]
0x1728b:      test     rdi, rdi
0x1728e:      je       0x17298
0x17290:      jmp      0x4940
0x4940
0x17295:      nop      dword ptr [rax]
0x17298:      ret
0x17299:      nop      dword ptr [rax]
0x172a0:      endbr64
0x172a4:      push     rbp
0x172a5:      xor      edx, edx
0x172a7:      xor      esi, esi
0x172a9:      lea      rbp, [rip + 0x143d0]
0x172b0:      push     rbx
0x172b1:      mov      rbx, rdi
0x172b4:      sub      rsp, 8
0x172b8:      mov      rdi, qword ptr [rip + 0x147c9]
0x172bf:      call     0x4c80
0x172c4:      jmp      0x172df
0x172df
0x172c6:      nop      word ptr cs:[rax + rax]
0x172d0:      mov      rsi, rbp
0x172d3:      mov      rdi, rbx
0x172d6:      call     0x4b30

```

```

0x172db:    test    eax, eax
0x172dd:    je      0x17300
0x172df:    xor     eax, eax
0x172e1:    call   0x17160
0x172e6:    test    eax, eax
0x172e8:    jne     0x172d0
0x172ea:    mov     eax, 0xffffffff
0x172ef:    mov     dword ptr [rip + 0x1478b], eax
0x172f5:    add     rsp, 8
0x172f9:    pop     rbx
0x172fa:    pop     rbp
0x172fb:    ret
0x172fc:    nop     dword ptr [rax]
0x17300:    mov     rdi, qword ptr [rip + 0x14781]
0x17307:    call   0x4b80
0x1730c:    mov     dword ptr [rip + 0x1476e], eax
0x17312:    add     rsp, 8
0x17316:    pop     rbx
0x17317:    pop     rbp
0x17318:    ret
0x17319:    nop     dword ptr [rax]
0x17320:    endbr64
0x17324:    push    r15
0x17326:    push    r14
0x17328:    push    r13
0x1732a:    mov     r13, rdi
0x1732d:    push    r12
0x1732f:    push    rbp
0x17330:    push    rbx
0x17331:    mov     ebx, ecx
0x17333:    sub     rsp, 0x438
0x1733a:    mov     qword ptr [rsp + 0x10], rsi
0x1733f:    mov     dword ptr [rsp + 0xc], edx
0x17343:    mov     rax, qword ptr fs:[0x28]
0x1734c:    mov     qword ptr [rsp + 0x428], rax
0x17354:    xor     eax, eax
0x17356:    test    ecx, ecx
0x17358:    jne     0x17618
0x1735e:    cmp     dword ptr [rsp + 0xc], 1
0x17363:    jne     0x175e0
0x17369:    mov     rax, qword ptr [rsp + 0x10]
0x1736e:    mov     qword ptr [rax], 0
0x17375:    lea     rbp, [rip + 0x14304]
0x1737c:    lea     r12, [rsp + 0x20]
0x17381:    lea     r14, [rip + 0xa092]
0x17388:    nop     dword ptr [rax + rax]
0x17390:    xor     eax, eax
0x17392:    call   0x17160

```

```

0x17397:    test    eax, eax
0x17399:    je      0x17600
0x1739f:    cmp     byte ptr [rbp], 0
0x173a3:    je      0x17600
0x173a9:    mov     rdx, r12
0x173ac:    mov     rsi, r14
0x173af:    mov     rdi, rbp
0x173b2:    xor     eax, eax
0x173b4:    mov     byte ptr [rsp + 0x20], 0
0x173b9:    call    0x4c60
0x173be:    call    0x4f00
0x173c3:    mov     r15, rax
0x173c6:    test    ebx, ebx
0x173c8:    jne     0x17410
0x173ca:    movsx   rcx, byte ptr [rsp + 0x20]
0x173d0:    mov     rax, qword ptr [rax]
0x173d3:    mov     byte ptr [rsp + 0x40], 0
0x173d8:    test    byte ptr [rax + rcx*2 + 1], 0x40
0x173dd:    jne     0x17433
0x173df:    mov     rax, qword ptr [rsp + 0x428]
0x173e7:    sub     rax, qword ptr fs:[0x28]
0x173f0:    jne     0x17657
0x173f6:    add     rsp, 0x438
0x173fd:    pop     rbx
0x173fe:    pop     rbp
0x173ff:    pop     r12
0x17401:    pop     r13
0x17403:    pop     r14
0x17405:    pop     r15
0x17407:    ret
0x17408:    nop     dword ptr [rax + rax]
0x17410:    movsx   rcx, byte ptr [rsp + 0x20]
0x17416:    mov     rax, qword ptr [rax]
0x17419:    test    byte ptr [rax + rcx*2 + 1], 0x40
0x1741e:    je      0x173df
0x17420:    mov     rsi, r12
0x17423:    mov     rdi, r13
0x17426:    call    0x4b30
0x1742b:    test    eax, eax
0x1742d:    jne     0x17390
0x17433:    mov     esi, 0x3d
0x17438:    mov     rdi, rbp
0x1743b:    call    0x4a00
0x17440:    test    rax, rax
0x17443:    je      0x174e3
0x17449:    cmp     byte ptr [rax + 1], 0
0x1744d:    je      0x174e3
0x17453:    mov     rdx, qword ptr [rsp + 0x10]

```

```

0x17458:    lea     rdi, [rax + 1]
0x1745c:    mov     qword ptr [rdx], rdi
0x1745f:    movzx   ecx, byte ptr [rax + 1]
0x17463:    test    cl, cl
0x17465:    je      0x1763f
0x1746b:    cmp     dword ptr [rsp + 0xc], 1
0x17470:    jle     0x1763f
0x17476:    mov     r8d, 1
0x1747c:    jmp     0x1748e
0x1748e:
0x1747e:    nop
0x17480:    mov     ecx, eax
0x17482:    cmp     r8d, dword ptr [rsp + 0xc]
0x17487:    jge     0x174d1
0x17489:    test    sil, sil
0x1748c:    je      0x174d1
0x1748e:    movzx   eax, byte ptr [rdi + 1]
0x17492:    test    al, al
0x17494:    setne   sil
0x17498:    add     rdi, 1
0x1749c:    cmp     cl, 0x3b
0x1749f:    jne     0x17480
0x174a1:    mov     byte ptr [rdi - 1], 0
0x174a5:    cmp     al, 0x3b
0x174a7:    je      0x17638
0x174ad:    test    sil, sil
0x174b0:    je      0x17638
0x174b6:    mov     rdx, qword ptr [rsp + 0x10]
0x174bb:    movsxd  rax, r8d
0x174be:    mov     qword ptr [rdx + rax*8], rdi
0x174c2:    movzx   ecx, byte ptr [rdi]
0x174c5:    test    cl, cl
0x174c7:    setne   sil
0x174cb:    add     r8d, 1
0x174cf:    jmp     0x17482
0x17482:
0x174d1:    mov     esi, 0x3b
0x174d6:    call    0x4a00
0x174db:    test    rax, rax
0x174de:    je      0x174ef
0x174e0:    mov     byte ptr [rax], 0
0x174e3:    mov     eax, dword ptr [rsp + 0xc]
0x174e7:    test    eax, eax
0x174e9:    jle     0x175c0
0x174ef:    mov     rsi, qword ptr [rsp + 0x10]
0x174f4:    xor     r8d, r8d
0x174f7:    jmp     0x17513
0x17513:

```

```

0x174f9:      nop      dword ptr [rax]
0x17500:      add      r8d, 1
0x17504:      add      rsi, 8
0x17508:      cmp      dword ptr [rsp + 0xc], r8d
0x1750d:      jle      0x175c0
0x17513:      mov      rbp, qword ptr [rsi]
0x17516:      test     rbp, rbp
0x17519:      je       0x17500
0x1751b:      mov      r14, qword ptr [r15]
0x1751e:      movsx    rax, byte ptr [rbp]
0x17523:      test     byte ptr [r14 + rax*2 + 1], 0x20
0x17529:      je       0x17547
0x1752b:      lea      rax, [rbp + 1]
0x1752f:      nop
0x17530:      mov      rbp, rax
0x17533:      mov      qword ptr [rsi], rax
0x17536:      add      rax, 1
0x1753a:      movsx    rcx, byte ptr [rbp]
0x1753f:      test     byte ptr [r14 + rcx*2 + 1], 0x20
0x17545:      jne      0x17530
0x17547:      mov      rdi, rbp
0x1754a:      mov      qword ptr [rsp + 0x18], rsi
0x1754f:      mov      dword ptr [rsp + 0x10], r8d
0x17554:      call     0x49b0
0x17559:      mov      r8d, dword ptr [rsp + 0x10]
0x1755e:      mov      rsi, qword ptr [rsp + 0x18]
0x17563:      lea      rax, [rbp + rax - 1]
0x17568:      movsx    rcx, byte ptr [rax]
0x1756c:      test     byte ptr [r14 + rcx*2 + 1], 0x20
0x17572:      jne      0x17591
0x17574:      jmp      0x17596
0x17596:
0x17576:      nop      word ptr cs:[rax + rax]
0x17580:      movsx    rcx, byte ptr [rax - 1]
0x17585:      sub      rax, 1
0x17589:      test     byte ptr [r14 + rcx*2 + 1], 0x20
0x1758f:      je       0x17596
0x17591:      cmp      rax, rbp
0x17594:      jae      0x17580
0x17596:      mov      byte ptr [rax + 1], 0
0x1759a:      mov      rax, qword ptr [rsi]
0x1759d:      cmp      byte ptr [rax], 0
0x175a0:      jne      0x17500
0x175a6:      mov      qword ptr [rsi], 0
0x175ad:      add      r8d, 1
0x175b1:      add      rsi, 8
0x175b5:      cmp      dword ptr [rsp + 0xc], r8d
0x175ba:      jg       0x17513

```

```

0x175c0:      test     ebx, ebx
0x175c2:      jne      0x173df
0x175c8:      mov      rsi, r12
0x175cb:      mov      rdi, r13
0x175ce:      call     0x48b0
0x175d3:      jmp      0x173df
0x173df
0x175d8:      nop      dword ptr [rax + rax]
0x175e0:      movsxd   rdx, dword ptr [rsp + 0xc]
0x175e5:      mov      rdi, qword ptr [rsp + 0x10]
0x175ea:      xor      esi, esi
0x175ec:      shl      rdx, 3
0x175f0:      call     0x4a50
0x175f5:      jmp      0x17375
0x17375
0x175fa:      nop      word ptr [rax + rax]
0x17600:      test     ebx, ebx
0x17602:      jne      0x173df
0x17608:      mov      byte ptr [r13], 0
0x1760d:      jmp      0x173df
0x173df
0x17612:      nop      word ptr [rax + rax]
0x17618:      movsxd   rsi, dword ptr [rip + 0x14461]
0x1761f:      mov      rdi, qword ptr [rip + 0x14462]
0x17626:      xor      edx, edx
0x17628:      call     0x4c80
0x1762d:      jmp      0x1735e
0x1735e
0x17632:      nop      word ptr [rax + rax]
0x17638:      mov      ecx, eax
0x1763a:      jmp      0x174cb
0x174cb
0x1763f:      mov      esi, 0x3b
0x17644:      call     0x4a00
0x17649:      test     rax, rax
0x1764c:      jne      0x174e0
0x17652:      jmp      0x174e3
0x174e3
0x17657:      call     0x49d0
0x1765c:      nop      dword ptr [rax]
0x17660:      endbr64
0x17664:      push     r12
0x17666:      mov      edi, 0xf
0x1766b:      lea      r12, [rip + 0xa8b4]
0x17672:      call     0x1d670
0x17677:      mov      rcx, r12
0x1767a:      lea      r9, [rip + 0xa895]
0x17681:      lea      r8, [rip + 0xa893]

```



```

0x17688:      mov     rdi, rax
0x1768b:      mov     rdx, -1
0x17692:      mov     esi, 1
0x17697:      mov     qword ptr [rip + 0x14402], rax
0x1769e:      xor     eax, eax
0x176a0:      call    0x4f10
0x176a5:      mov     edi, 0x17
0x176aa:      call    0x1d670
0x176af:      mov     rcx, r12
0x176b2:      pop     r12
0x176b4:      lea     r9, [rip + 0xa870]
0x176bb:      mov     qword ptr [rip + 0x143e6], rax
0x176c2:      mov     rdi, rax
0x176c5:      xor     eax, eax
0x176c7:      lea     r8, [rip + 0xa84d]
0x176ce:      mov     rdx, -1
0x176d5:      mov     esi, 1
0x176da:      jmp     0x4f10
0x4f10:
0x176df:      nop
0x176e0:      endbr64
0x176e4:      push    rbp
0x176e5:      mov     rbp, rdi
0x176e8:      xor     edi, edi
0x176ea:      push    rbx
0x176eb:      sub     rsp, 0x18
0x176ef:      mov     rax, qword ptr fs:[0x28]
0x176f8:      mov     qword ptr [rsp + 8], rax
0x176fd:      xor     eax, eax
0x176ff:      call    0x4be0
0x17704:      mov     rdi, rsp
0x17707:      mov     qword ptr [rsp], rax
0x1770b:      call    0x4840
0x17710:      mov     rbx, rax
0x17713:      call    0x4920
0x17718:      movsxd  rcx, dword ptr [rbx + 0x10]
0x1771c:      sub     rsp, 8
0x17720:      lea     rdx, [rip + 0x12fd9]
0x17727:      mov     r9d, dword ptr [rbx + 8]
0x1772b:      mov     r8d, dword ptr [rbx + 0xc]
0x1772f:      mov     esi, 1
0x17734:      mov     rcx, qword ptr [rdx + rcx*8]
0x17738:      push    rbp
0x17739:      lea     rdx, [rip + 0xa8d0]
0x17740:      push    rax
0x17741:      mov     rdi, qword ptr [rip + 0x13138]
0x17748:      push    qword ptr [rip + 0x1319a]
0x1774e:      mov     eax, dword ptr [rbx]

```

```

0x17750:    push    rax
0x17751:    mov     eax, dword ptr [rbx + 4]
0x17754:    push    rax
0x17755:    xor     eax, eax
0x17757:    call    0x4e60
0x1775c:    add     rsp, 0x30
0x17760:    mov     rax, qword ptr [rsp + 8]
0x17765:    sub     rax, qword ptr fs:[0x28]
0x1776e:    jne     0x17777
0x17770:    add     rsp, 0x18
0x17774:    pop     rbx
0x17775:    pop     rbp
0x17776:    ret
0x17777:    call    0x49d0
0x1777c:    nop     dword ptr [rax]
0x17780:    endbr64
0x17784:    push    rbp
0x17785:    xor     eax, eax
0x17787:    mov     ebp, edi
0x17789:    call    0x8ad0
0x1778e:    cmp     ebp, 0xb
0x17791:    je      0x177d3
0x17793:    jg      0x177c0
0x17795:    lea     eax, [rbp - 1]
0x17798:    cmp     eax, 1
0x1779b:    ja      0x17802
0x1779d:    cmp     ebp, 1
0x177a0:    lea     rdi, [rip + 0xa791]
0x177a7:    lea     rax, [rip + 0xa78e]
0x177ae:    cmovne  rdi, rax
0x177b2:    call    0x176e0
0x177b7:    mov     edi, ebp
0x177b9:    call    0x4e20
0x177be:    nop
0x177c0:    cmp     ebp, 0xf
0x177c3:    jne     0x17802
0x177c5:    lea     rdi, [rip + 0x8dbe]
0x177cc:    call    0x176e0
0x177d1:    jmp     0x177b7
0x177b7:
0x177d3:    lea     rdi, [rip + 0xa766]
0x177da:    call    0x176e0
0x177df:    mov     rcx, qword ptr [rip + 0x13102]
0x177e6:    mov     rdi, qword ptr [rip + 0x13093]
0x177ed:    lea     rdx, [rip + 0xa88c]
0x177f4:    xor     eax, eax
0x177f6:    mov     esi, 1
0x177fb:    call    0x4e60

```

```

0x17800:      jmp      0x177b7
0x177b7
0x17802:      mov     rcx, qword ptr [rip + 0x130df]
0x17809:      mov     rdi, qword ptr [rip + 0x13070]
0x17810:      mov     r8d, ebp
0x17813:      xor     eax, eax
0x17815:      lea     rdx, [rip + 0xa82c]
0x1781c:      mov     esi, 1
0x17821:      call    0x4e60
0x17826:      jmp     0x177df
0x177df
0x17828:      nop     dword ptr [rax + rax]
0x17830:      endbr64
0x17834:      push    rbx
0x17835:      mov     rcx, qword ptr [rip + 0x130ac]
0x1783c:      mov     rbx, rdi
0x1783f:      mov     esi, 1
0x17844:      mov     rdi, qword ptr [rip + 0x13035]
0x1784b:      lea     rdx, [rip + 0xa6f3]
0x17852:      xor     eax, eax
0x17854:      call    0x4e60
0x17859:      mov     r8, rbx
0x1785c:      mov     rcx, qword ptr [rip + 0x13085]
0x17863:      pop     rbx
0x17864:      mov     rdi, qword ptr [rip + 0x13015]
0x1786b:      lea     rdx, [rip + 0xa836]
0x17872:      mov     esi, 1
0x17877:      xor     eax, eax
0x17879:      jmp     0x4e60
0x4e60
0x1787e:      nop
0x17880:      endbr64
0x17884:      sub     rsp, 8
0x17888:      mov     r8, rdi
0x1788b:      mov     rcx, qword ptr [rip + 0x13056]
0x17892:      xor     eax, eax
0x17894:      mov     rdi, qword ptr [rip + 0x12fe5]
0x1789b:      lea     rdx, [rip + 0xa836]
0x178a2:      mov     esi, 1
0x178a7:      call    0x4e60
0x178ac:      mov     rcx, qword ptr [rip + 0x13035]
0x178b3:      mov     esi, 1
0x178b8:      xor     eax, eax
0x178ba:      mov     rdi, qword ptr [rip + 0x12fbf]
0x178c1:      lea     rdx, [rip + 0xa840]
0x178c8:      call    0x4e60
0x178cd:      mov     rcx, qword ptr [rip + 0x13014]
0x178d4:      mov     esi, 1

```

```

0x178d9:    xor     eax, eax
0x178db:    mov     rdi, qword ptr [rip + 0x12f9e]
0x178e2:    lea     rdx, [rip + 0xa84f]
0x178e9:    call    0x4e60
0x178ee:    mov     rcx, qword ptr [rip + 0x12ff3]
0x178f5:    mov     esi, 1
0x178fa:    xor     eax, eax
0x178fc:    mov     rdi, qword ptr [rip + 0x12f7d]
0x17903:    lea     rdx, [rip + 0xa86e]
0x1790a:    call    0x4e60
0x1790f:    mov     rcx, qword ptr [rip + 0x12fd2]
0x17916:    mov     esi, 1
0x1791b:    xor     eax, eax
0x1791d:    mov     rdi, qword ptr [rip + 0x12f5c]
0x17924:    lea     rdx, [rip + 0xa885]
0x1792b:    call    0x4e60
0x17930:    mov     rcx, qword ptr [rip + 0x12fb1]
0x17937:    mov     esi, 1
0x1793c:    xor     eax, eax
0x1793e:    mov     rdi, qword ptr [rip + 0x12f3b]
0x17945:    lea     rdx, [rip + 0xa8a4]
0x1794c:    call    0x4e60
0x17951:    mov     rcx, qword ptr [rip + 0x12f90]
0x17958:    mov     esi, 1
0x1795d:    xor     eax, eax
0x1795f:    mov     rdi, qword ptr [rip + 0x12f1a]
0x17966:    lea     rdx, [rip + 0xa8c3]
0x1796d:    call    0x4e60
0x17972:    mov     rcx, qword ptr [rip + 0x12f6f]
0x17979:    mov     esi, 1
0x1797e:    xor     eax, eax
0x17980:    mov     rdi, qword ptr [rip + 0x12ef9]
0x17987:    lea     rdx, [rip + 0xa8ea]
0x1798e:    add     rsp, 8
0x17992:    jmp     0x4e60
0x4e60:
0x17997:    nop     word ptr [rax + rax]
0x179a0:    endbr64
0x179a4:    push    r12
0x179a6:    lea     r12, [rip + 0x12d33]
0x179ad:    push    rbp
0x179ae:    sub     rsp, 8
0x179b2:    mov     rdi, qword ptr [rip + 0x140ef]
0x179b9:    call    0x49b0
0x179be:    mov     rdi, r12
0x179c1:    mov     rbp, rax
0x179c4:    call    0x49b0
0x179c9:    lea     rdi, [rbp + rax + 9]

```

```

0x179ce:      call    0x1d670
0x179d3:      mov     rsi, qword ptr [rip + 0x140ce]
0x179da:      mov     rdi, rax
0x179dd:      mov     rbp, rax
0x179e0:      call    0x4980
0x179e5:      mov     rsi, r12
0x179e8:      mov     rdi, rax
0x179eb:      call    0x4980
0x179f0:      mov     edx, 0x6e6f
0x179f5:      mov     rdi, rbp
0x179f8:      mov     dword ptr [rax], 0x6d6d6f63
0x179fe:      mov     word ptr [rax + 4], dx
0x17a02:      mov     byte ptr [rax + 6], 0
0x17a06:      call    0x17230
0x17a0b:      test    eax, eax
0x17a0d:      je      0x17a17
0x17a0f:      add     rsp, 8
0x17a13:      pop     rbp
0x17a14:      pop     r12
0x17a16:      ret
0x17a17:      mov     rdi, rbp
0x17a1a:      call    0x17830
0x17a1f:      mov     edi, 1
0x17a24:      call    0x4e20
0x17a29:      nop     dword ptr [rax]
0x17a30:      endbr64
0x17a34:      push    r13
0x17a36:      push    r12
0x17a38:      lea     r12, [rip + 0x12c91]
0x17a3f:      push    rbp
0x17a40:      push    rbx
0x17a41:      sub     rsp, 8
0x17a45:      mov     rdi, qword ptr [rip + 0x12ea4]
0x17a4c:      call    0x49b0
0x17a51:      mov     rdi, r12
0x17a54:      mov     rbx, rax
0x17a57:      call    0x49b0
0x17a5c:      mov     rdi, qword ptr [rip + 0x12eed]
0x17a63:      mov     rbp, rax
0x17a66:      call    0x49b0
0x17a6b:      lea     rdi, [rbx + rbp + 2]
0x17a70:      add     rdi, rax
0x17a73:      call    0x1d670
0x17a78:      mov     rsi, qword ptr [rip + 0x12e71]
0x17a7f:      mov     rdi, rax
0x17a82:      mov     rbp, rax
0x17a85:      call    0x4980
0x17a8a:      mov     rsi, r12

```

```

0x17a8d:    mov     rdi, rax
0x17a90:    call   0x4980
0x17a95:    mov     rsi, qword ptr [rip + 0x12eb4]
0x17a9c:    mov     rdi, rax
0x17a9f:    call   0x48b0
0x17aa4:    mov     rdi, rbp
0x17aa7:    call   0x17230
0x17aac:    test    eax, eax
0x17aae:    jne     0x17b2a
0x17ab0:    mov     rdi, rbp
0x17ab3:    lea     r12, [rip + 0x12c06]
0x17aba:    call   0x18d10
0x17abf:    mov     rdi, qword ptr [rip + 0x12e2a]
0x17ac6:    call   0x49b0
0x17acb:    mov     rdi, r12
0x17ace:    mov     rbp, rax
0x17ad1:    call   0x49b0
0x17ad6:    mov     rdi, qword ptr [rip + 0x12e73]
0x17add:    mov     rbx, rax
0x17ae0:    call   0x49b0
0x17ae5:    lea     rdi, [rbp + rbx + 2]
0x17aea:    add     rdi, rax
0x17aed:    call   0x1d670
0x17af2:    mov     rsi, qword ptr [rip + 0x12df7]
0x17af9:    mov     rdi, rax
0x17afc:    mov     rbp, rax
0x17aff:    call   0x4980
0x17b04:    mov     rsi, r12
0x17b07:    mov     rdi, rax
0x17b0a:    call   0x4980
0x17b0f:    mov     rsi, qword ptr [rip + 0x12e3a]
0x17b16:    mov     rdi, rax
0x17b19:    call   0x48b0
0x17b1e:    mov     rdi, rbp
0x17b21:    call   0x17230
0x17b26:    test    eax, eax
0x17b28:    je      0x17b50
0x17b2a:    mov     rdi, rbp
0x17b2d:    mov     r12d, 1
0x17b33:    call   0x18d10
0x17b38:    add     rsp, 8
0x17b3c:    mov     eax, r12d
0x17b3f:    pop     rbx
0x17b40:    pop     rbp
0x17b41:    pop     r12
0x17b43:    pop     r13
0x17b45:    ret
0x17b46:    nop     word ptr cs:[rax + rax]

```

```

0x17b50:    mov     rdi, rbp
0x17b53:    lea     r13, [rip + 0x12b86]
0x17b5a:    call    0x18d10
0x17b5f:    mov     rdi, qword ptr [rip + 0x13f42]
0x17b66:    call    0x49b0
0x17b6b:    mov     rdi, r13
0x17b6e:    mov     rbp, rax
0x17b71:    call    0x49b0
0x17b76:    mov     rdi, qword ptr [rip + 0x12dd3]
0x17b7d:    mov     rbx, rax
0x17b80:    call    0x49b0
0x17b85:    lea     rdi, [rbp + rbx + 2]
0x17b8a:    add     rdi, rax
0x17b8d:    call    0x1d670
0x17b92:    mov     rsi, qword ptr [rip + 0x13f0f]
0x17b99:    mov     rdi, rax
0x17b9c:    mov     rbp, rax
0x17b9f:    call    0x4980
0x17ba4:    mov     rsi, r13
0x17ba7:    mov     rdi, rax
0x17baa:    call    0x4980
0x17baf:    mov     rsi, qword ptr [rip + 0x12d9a]
0x17bb6:    mov     rdi, rax
0x17bb9:    call    0x48b0
0x17bbe:    mov     rdi, rbp
0x17bc1:    call    0x17230
0x17bc6:    mov     r12d, eax
0x17bc9:    test    eax, eax
0x17bcb:    jne     0x17b2a
0x17bd1:    xor     eax, eax
0x17bd3:    mov     rdi, rbp
0x17bd6:    call    0x17880
0x17bdb:    mov     rdi, rbp
0x17bde:    call    0x18d10
0x17be3:    mov     rdi, qword ptr [rip + 0x13ebe]
0x17bea:    call    0x49b0
0x17bef:    mov     rdi, r13
0x17bf2:    mov     rbx, rax
0x17bf5:    call    0x49b0
0x17bfa:    lea     rdi, [rbx + rax + 0xa]
0x17bff:    call    0x1d670
0x17c04:    mov     rsi, qword ptr [rip + 0x13e9d]
0x17c0b:    mov     rdi, rax
0x17c0e:    mov     rbp, rax
0x17c11:    call    0x4980
0x17c16:    mov     rsi, r13
0x17c19:    mov     rdi, rax
0x17c1c:    call    0x4980

```

```

0x17c21:      mov     rdi, rbp
0x17c24:      movabs  rdx, 0x636972656e6567
0x17c2e:      mov     qword ptr [rax], rdx
0x17c31:      call    0x17230
0x17c36:      test    eax, eax
0x17c38:      jne     0x17b38
0x17c3e:      mov     rdi, rbp
0x17c41:      call    0x17830
0x17c46:      mov     edi, 1
0x17c4b:      call    0x4e20
0x17c50:      endbr64
0x17c54:      push    rbp
0x17c55:      push    rbx
0x17c56:      mov     rbx, rdi
0x17c59:      sub     rsp, 8
0x17c5d:      call    0x172a0
0x17c62:      cmp     eax, -1
0x17c65:      je      0x17c70
0x17c67:      add     rsp, 8
0x17c6b:      pop     rbx
0x17c6c:      pop     rbp
0x17c6d:      ret
0x17c6e:      nop
0x17c70:      mov     edx, 7
0x17c75:      lea     rsi, [rip + 0xa2e5]
0x17c7c:      mov     rdi, rbx
0x17c7f:      call    0x4890
0x17c84:      test    eax, eax
0x17c86:      jne     0x17cd4
0x17c88:      lea     rdi, [rbx + 5]
0x17c8c:      call    0x49b0
0x17c91:      lea     rdi, [rax + 6]
0x17c95:      call    0x1d670
0x17c9a:      lea     rsi, [rbx + 7]
0x17c9e:      mov     dword ptr [rax], 0x5449475b
0x17ca4:      mov     rbp, rax
0x17ca7:      lea     rdi, [rax + 5]
0x17cab:      mov     byte ptr [rax + 4], 0x2d
0x17caf:      call    0x48b0
0x17cb4:      mov     rdi, rbp
0x17cb7:      call    0x172a0
0x17cbc:      mov     rdi, rbp
0x17cbf:      cmp     eax, -1
0x17cc2:      je      0x17ccf
0x17cc4:      add     rsp, 8
0x17cc8:      pop     rbx
0x17cc9:      pop     rbp
0x17cca:      jmp     0x18d10

```



```

0x18d10
0x17ccf:      call    0x18d10
0x17cd4:      mov     rdi, qword ptr [rip + 0x12ba5]
0x17cdb:      mov     rcx, qword ptr [rip + 0x12c06]
0x17ce2:      mov     r8, rbx
0x17ce5:      xor     eax, eax
0x17ce7:      lea     rdx, [rip + 0xa5ba]
0x17cee:      mov     esi, 1
0x17cf3:      call    0x4e60
0x17cf8:      mov     edi, 1
0x17cfd:      call    0x4e20
0x17d02:      nop     word ptr cs:[rax + rax]
0x17d0d:      nop     dword ptr [rax]
0x17d10:      endbr64
0x17d14:      push    rbx
0x17d15:      mov     ecx, 1
0x17d1a:      mov     ebx, esi
0x17d1c:      mov     edx, 1
0x17d21:      sub     rsp, 0x10
0x17d25:      mov     rax, qword ptr fs:[0x28]
0x17d2e:      mov     qword ptr [rsp + 8], rax
0x17d33:      xor     eax, eax
0x17d35:      mov     rsi, rsp
0x17d38:      call    0x17320
0x17d3d:      mov     rdi, qword ptr [rsp]
0x17d41:      mov     eax, ebx
0x17d43:      test    rdi, rdi
0x17d46:      je      0x17d54
0x17d48:      mov     edx, 0xa
0x17d4d:      xor     esi, esi
0x17d4f:      call    0x4ba0
0x17d54:      mov     rdx, qword ptr [rsp + 8]
0x17d59:      sub     rdx, qword ptr fs:[0x28]
0x17d62:      jne     0x17d6a
0x17d64:      add     rsp, 0x10
0x17d68:      pop     rbx
0x17d69:      ret
0x17d6a:      call    0x49d0
0x17d6f:      nop
0x17d70:      endbr64
0x17d74:      push    r15
0x17d76:      push    r14
0x17d78:      mov     r14, rdi
0x17d7b:      push    r13
0x17d7d:      mov     r13d, ecx
0x17d80:      mov     ecx, 1
0x17d85:      push    r12
0x17d87:      mov     r12d, r13d

```

```

0x17d8a:    push    rbp
0x17d8b:    mov     ebp, edx
0x17d8d:    mov     edx, 1
0x17d92:    push    rbx
0x17d93:    mov     rbx, rsi
0x17d96:    sub     rsp, 0x18
0x17d9a:    mov     rax, qword ptr fs:[0x28]
0x17da3:    mov     qword ptr [rsp + 8], rax
0x17da8:    xor     eax, eax
0x17daa:    mov     rsi, rsp
0x17dad:    call    0x17320
0x17db2:    mov     r15, qword ptr [rsp]
0x17db6:    test    r15, r15
0x17db9:    je      0x17de4
0x17dbb:    test    ebp, ebp
0x17dbd:    jle     0x17e3c
0x17dbf:    xor     r12d, r12d
0x17dc2:    jmp     0x17dd5
0x17dd5:
0x17dc4:    nop     dword ptr [rax]
0x17dc8:    add     r12d, 1
0x17dcc:    add     rbx, 8
0x17dd0:    cmp     ebp, r12d
0x17dd3:    je      0x17e10
0x17dd5:    mov     rsi, qword ptr [rbx]
0x17dd8:    mov     rdi, r15
0x17ddb:    call    0x4b30
0x17de0:    test    eax, eax
0x17de2:    jne     0x17dc8
0x17de4:    mov     rax, qword ptr [rsp + 8]
0x17de9:    sub     rax, qword ptr fs:[0x28]
0x17df2:    jne     0x17e43
0x17df4:    add     rsp, 0x18
0x17df8:    mov     eax, r12d
0x17dfb:    pop     rbx
0x17dfc:    pop     rbp
0x17dfd:    pop     r12
0x17dff:    pop     r13
0x17e01:    pop     r14
0x17e03:    pop     r15
0x17e05:    ret
0x17e06:    nop     word ptr cs:[rax + rax]
0x17e10:    mov     rcx, qword ptr [rip + 0x12ad1]
0x17e17:    mov     r9, r15
0x17e1a:    mov     r8, r14
0x17e1d:    xor     eax, eax
0x17e1f:    mov     rdi, qword ptr [rip + 0x12a5a]
0x17e26:    lea     rdx, [rip + 0xa13c]

```

```

0x17e2d:      mov     esi, 1
0x17e32:      mov     r12d, r13d
0x17e35:      call    0x4e60
0x17e3a:      jmp     0x17de4
0x17de4:
0x17e3c:      je      0x17e10
0x17e3e:      xor     r12d, r12d
0x17e41:      jmp     0x17de4
0x17de4:
0x17e43:      call    0x49d0
0x17e48:      nop     dword ptr [rax + rax]
0x17e50:      endbr64
0x17e54:      push    r12
0x17e56:      mov     ecx, 1
0x17e5b:      mov     edx, 1
0x17e60:      mov     r12, rdi
0x17e63:      push    rbp
0x17e64:      push    rbx
0x17e65:      mov     ebx, esi
0x17e67:      sub     rsp, 0x10
0x17e6b:      mov     rax, qword ptr fs:[0x28]
0x17e74:      mov     qword ptr [rsp + 8], rax
0x17e79:      xor     eax, eax
0x17e7b:      mov     rsi, rsp
0x17e7e:      call    0x17320
0x17e83:      mov     rbp, qword ptr [rsp]
0x17e87:      mov     eax, ebx
0x17e89:      test    rbp, rbp
0x17e8c:      je      0x17ea7
0x17e8e:      cmp     byte ptr [rbp], 0x4f
0x17e92:      je      0x17ec0
0x17e94:      lea     rsi, [rip + 0x8e4b]
0x17e9b:      mov     rdi, rbp
0x17e9e:      call    0x4b30
0x17ea3:      test    eax, eax
0x17ea5:      jne     0x17ed8
0x17ea7:      mov     rdx, qword ptr [rsp + 8]
0x17eac:      sub     rdx, qword ptr fs:[0x28]
0x17eb5:      jne     0x17f03
0x17eb7:      add     rsp, 0x10
0x17ebb:      pop     rbx
0x17ebc:      pop     rbp
0x17ebd:      pop     r12
0x17ebf:      ret
0x17ec0:      cmp     byte ptr [rbp + 1], 0x4e
0x17ec4:      jne     0x17e94
0x17ec6:      cmp     byte ptr [rbp + 2], 0
0x17eca:      mov     eax, 1

```

```

0x17ecf:      je      0x17ea7
0x17ed1:      jmp     0x17e94
0x17e94:
0x17ed3:      nop     dword ptr [rax + rax]
0x17ed8:      mov     rcx, qword ptr [rip + 0x12a09]
0x17edf:      mov     r9, rbp
0x17ee2:      mov     r8, r12
0x17ee5:      xor     eax, eax
0x17ee7:      mov     rdi, qword ptr [rip + 0x12992]
0x17eee:      lea     rdx, [rip + 0xa074]
0x17ef5:      mov     esi, 1
0x17efa:      call    0x4e60
0x17eff:      mov     eax, ebx
0x17f01:      jmp     0x17ea7
0x17ea7:
0x17f03:      call    0x49d0
0x17f08:      nop     dword ptr [rax + rax]
0x17f10:      endbr64
0x17f14:      push    r12
0x17f16:      mov     ecx, 1
0x17f1b:      mov     r12, rsi
0x17f1e:      mov     edx, 1
0x17f23:      sub     rsp, 0x10
0x17f27:      mov     rax, qword ptr fs:[0x28]
0x17f30:      mov     qword ptr [rsp + 8], rax
0x17f35:      xor     eax, eax
0x17f37:      mov     rsi, rsp
0x17f3a:      call    0x17320
0x17f3f:      mov     rdi, qword ptr [rsp]
0x17f43:      test    rdi, rdi
0x17f46:      je      0x17f50
0x17f48:      call    0x1d7a0
0x17f4d:      mov     r12, rax
0x17f50:      mov     rax, qword ptr [rsp + 8]
0x17f55:      sub     rax, qword ptr fs:[0x28]
0x17f5e:      jne     0x17f6a
0x17f60:      add     rsp, 0x10
0x17f64:      mov     rax, r12
0x17f67:      pop     r12
0x17f69:      ret
0x17f6a:      call    0x49d0
0x17f6f:      nop
0x17f70:      endbr64
0x17f74:      push    r14
0x17f76:      push    r13
0x17f78:      push    r12
0x17f7a:      push    rbp
0x17f7b:      push    rbx

```

```

0x17f7c:      sub     rsp, 0x10
0x17f80:      mov     rax, qword ptr fs:[0x28]
0x17f89:      mov     qword ptr [rsp + 8], rax
0x17f8e:      xor     eax, eax
0x17f90:      test    edx, edx
0x17f92:      jle     0x17fe8
0x17f94:      lea     eax, [rdx - 1]
0x17f97:      mov     rbx, rsi
0x17f9a:      mov     rbp, rdi
0x17f9d:      mov     r12, rsp
0x17fa0:      lea     r13, [rsi + rax*8 + 8]
0x17fa5:      lea     r14, [rip + 0x9fbd]
0x17fac:      nop     dword ptr [rax]
0x17fb0:      mov     rdi, qword ptr [rbx]
0x17fb3:      mov     ecx, 1
0x17fb8:      mov     edx, 1
0x17fbd:      mov     rsi, r12
0x17fc0:      call    0x17320
0x17fc5:      mov     rdi, qword ptr [rsp]
0x17fc9:      test    rdi, rdi
0x17fcc:      je      0x17fdb
0x17fce:      call    0xbd00
0x17fd3:      cmp     eax, -1
0x17fd6:      je      0x18008
0x17fd8:      mov     dword ptr [rbp], eax
0x17fdb:      add     rbx, 8
0x17fdf:      add     rbp, 4
0x17fe3:      cmp     rbx, r13
0x17fe6:      jne     0x17fb0
0x17fe8:      mov     rax, qword ptr [rsp + 8]
0x17fed:      sub     rax, qword ptr fs:[0x28]
0x17ff6:      jne     0x1802e
0x17ff8:      add     rsp, 0x10
0x17ffc:      pop     rbx
0x17ffd:      pop     rbp
0x17ffe:      pop     r12
0x18000:      pop     r13
0x18002:      pop     r14
0x18004:      ret
0x18005:      nop     dword ptr [rax]
0x18008:      mov     r9, qword ptr [rsp]
0x1800c:      mov     r8, qword ptr [rbx]
0x1800f:      mov     rdx, r14
0x18012:      mov     esi, 1
0x18017:      mov     rcx, qword ptr [rip + 0x128ca]
0x1801e:      mov     rdi, qword ptr [rip + 0x1285b]
0x18025:      xor     eax, eax
0x18027:      call    0x4e60

```

```

0x1802c:      jmp      0x17fdb
0x17fdb
0x1802e:      call     0x49d0
0x18033:      nop      word ptr cs:[rax + rax]
0x1803e:      nop
0x18040:      endbr64
0x18044:      push     r12
0x18046:      cmp      byte ptr [rdi], 0x2f
0x18049:      mov      r12, rdi
0x1804c:      lea      rax, [rdi + 1]
0x18050:      je       0x18061
0x18052:      lea      rdi, [rip + 0x9f26]
0x18059:      call     0x8b00
0x1805e:      mov      rax, r12
0x18061:      movzx    edx, byte ptr [rax]
0x18064:      mov      rsi, rax
0x18067:      test     dl, dl
0x18069:      jne      0x1808c
0x1806b:      jmp      0x180f0
0x180f0
0x18070:      cmp      dl, 0x2e
0x18073:      jne      0x18118
0x18079:      movzx    ecx, byte ptr [rax + 1]
0x1807d:      cmp      cl, 0x2f
0x18080:      je       0x18086
0x18082:      test     cl, cl
0x18084:      jne      0x180a0
0x18086:      add      rax, 1
0x1808a:      mov      edx, ecx
0x1808c:      cmp      dl, 0x2f
0x1808f:      jne      0x18070
0x18091:      movzx    ecx, byte ptr [rax + 1]
0x18095:      jmp      0x18086
0x18086
0x18097:      nop      word ptr [rax + rax]
0x180a0:      cmp      cl, 0x2e
0x180a3:      jne      0x180c0
0x180a5:      movzx    edx, byte ptr [rax + 2]
0x180a9:      cmp      dl, 0x2f
0x180ac:      je       0x18120
0x180ae:      test     dl, dl
0x180b0:      je       0x18120
0x180b2:      mov      edx, 0x2e
0x180b7:      nop      word ptr [rax + rax]
0x180c0:      mov      rdi, rax
0x180c3:      mov      rcx, rsi
0x180c6:      add      rax, 1
0x180ca:      add      rsi, 1

```

```

0x180ce:    mov     byte ptr [rsi - 1], dl
0x180d1:    movzx   edx, byte ptr [rax]
0x180d4:    test    dl, dl
0x180d6:    je      0x180f0
0x180d8:    cmp     dl, 0x2f
0x180db:    jne     0x180c0
0x180dd:    test    dl, dl
0x180df:    je      0x180f0
0x180e1:    mov     byte ptr [rsi], 0x2f
0x180e4:    movzx   edx, byte ptr [rdi + 1]
0x180e8:    lea     rsi, [rcx + 2]
0x180ec:    test    dl, dl
0x180ee:    jne     0x1808c
0x180f0:    cmp     byte ptr [rsi - 1], 0x2f
0x180f4:    jne     0x18109
0x180f6:    mov     rax, rsi
0x180f9:    sub     rax, r12
0x180fc:    cmp     rax, 1
0x18100:    setg    al
0x18103:    movzx   eax, al
0x18106:    sub     rsi, rax
0x18109:    mov     rax, r12
0x1810c:    mov     byte ptr [rsi], 0
0x1810f:    pop     r12
0x18111:    ret
0x18112:    nop     word ptr [rax + rax]
0x18118:    test    dl, dl
0x1811a:    jne     0x180c0
0x1811c:    jmp     0x180f0
0x180f0
0x1811e:    nop
0x18120:    lea     rdi, [rsi - 1]
0x18124:    cmp     rdi, r12
0x18127:    jne     0x1812f
0x18129:    add     rax, 2
0x1812d:    jmp     0x180ec
0x180ec
0x1812f:    cmp     byte ptr [rsi - 2], 0x2f
0x18133:    lea     rcx, [rsi - 2]
0x18137:    je      0x1814e
0x18139:    nop     dword ptr [rax]
0x18140:    mov     rsi, rcx
0x18143:    sub     rcx, 1
0x18147:    cmp     byte ptr [rcx], 0x2f
0x1814a:    jne     0x18140
0x1814c:    jmp     0x18129
0x18129
0x1814e:    mov     rsi, rdi

```

```

0x18151:      jmp      0x18129
0x18129
0x18153:      nop      word ptr cs:[rax + rax]
0x1815e:      nop
0x18160:      endbr64
0x18164:      push     rbx
0x18165:      call     0x4a60
0x1816a:      mov      edi, eax
0x1816c:      mov      ebx, eax
0x1816e:      call     0x4970
0x18173:      test     rax, rax
0x18176:      je       0x18189
0x18178:      mov      rdi, qword ptr [rax]
0x1817b:      call     0x1d7a0
0x18180:      pop      rbx
0x18181:      mov      qword ptr [rip + 0x13910], rax
0x18188:      ret
0x18189:      mov      rcx, qword ptr [rip + 0x12758]
0x18190:      mov      r8d, ebx
0x18193:      mov      rdi, qword ptr [rip + 0x126e6]
0x1819a:      lea      rdx, [rip + 0xa13f]
0x181a1:      mov      esi, 1
0x181a6:      call     0x4e60
0x181ab:      mov      rcx, qword ptr [rip + 0x12736]
0x181b2:      xor      eax, eax
0x181b4:      mov      rdi, qword ptr [rip + 0x126c5]
0x181bb:      lea      r8, [rip + 0x9dd7]
0x181c2:      lea      rdx, [rip + 0xa14f]
0x181c9:      mov      esi, 1
0x181ce:      call     0x4e60
0x181d3:      mov      rdi, qword ptr [rip + 0x126a6]
0x181da:      mov      esi, 1
0x181df:      xor      eax, eax
0x181e1:      mov      rcx, qword ptr [rip + 0x12700]
0x181e8:      lea      rdx, [rip + 0xa161]
0x181ef:      call     0x4e60
0x181f4:      mov      edi, 1
0x181f9:      call     0x4e20
0x181fe:      nop
0x18200:      endbr64
0x18204:      push     r12
0x18206:      push     rbp
0x18207:      mov      rbp, rsi
0x1820a:      push     rbx
0x1820b:      movsxd   rbx, edx
0x1820e:      cmp      ebx, 2
0x18211:      je       0x18250
0x18213:      mov      r12, rdi

```



```

0x18216:      jg      0x18230
0x18218:      test   ebx, ebx
0x1821a:      je      0x18224
0x1821c:      cmp     ebx, 1
0x1821f:      jne     0x18260
0x18221:      mov     byte ptr [rsi], 0x20
0x18224:      pop     rbx
0x18225:      pop     rbp
0x18226:      pop     r12
0x18228:      ret
0x18229:      nop     dword ptr [rax]
0x18230:      cmp     ebx, 3
0x18233:      jne     0x18260
0x18235:      mov     edx, 0x2020
0x1823a:      mov     byte ptr [rsi + 2], 0x20
0x1823e:      mov     word ptr [rsi], dx
0x18241:      pop     rbx
0x18242:      pop     rbp
0x18243:      pop     r12
0x18245:      ret
0x18246:      nop     word ptr cs:[rax + rax]
0x18250:      mov     ecx, 0x2020
0x18255:      mov     word ptr [rsi], cx
0x18258:      pop     rbx
0x18259:      pop     rbp
0x1825a:      pop     r12
0x1825c:      ret
0x1825d:      nop     dword ptr [rax]
0x18260:      mov     rdi, r12
0x18263:      call    0x49b0
0x18268:      movsxd  rdx, eax
0x1826b:      cmp     ebx, eax
0x1826d:      jge     0x182a0
0x1826f:      mov     eax, 0x2e2e
0x18274:      lea     r8d, [rbx - 3]
0x18278:      mov     byte ptr [rbp + 2], 0x2e
0x1827c:      sub     rdx, rbx
0x1827f:      mov     word ptr [rbp], ax
0x18283:      movsxd  r8, r8d
0x18286:      lea     rsi, [r12 + rdx + 3]
0x1828b:      pop     rbx
0x1828c:      lea     rdi, [rbp + 3]
0x18290:      mov     rdx, r8
0x18293:      pop     rbp
0x18294:      pop     r12
0x18296:      jmp     0x4bb0
0x4bb0
0x1829b:      nop     dword ptr [rax + rax]

```

```

0x182a0:      pop     rbx
0x182a1:      mov     rsi, r12
0x182a4:      mov     rdi, rbp
0x182a7:      pop     rbp
0x182a8:      pop     r12
0x182aa:      jmp     0x4bb0
0x4bb0:
0x182af:      nop
0x182b0:      endbr64
0x182b4:      push    r12
0x182b6:      mov     r12, rsi
0x182b9:      push    rbp
0x182ba:      mov     rbp, rdi
0x182bd:      push    rbx
0x182be:      lea     ebx, [rdx - 1]
0x182c1:      mov     edx, ebx
0x182c3:      call    0x18200
0x182c8:      mov     rdi, rbp
0x182cb:      call    0x49b0
0x182d0:      movsxd  rcx, ebx
0x182d3:      cmp     ebx, eax
0x182d5:      movsxd  rdx, eax
0x182d8:      mov     rax, rcx
0x182db:      pop     rbx
0x182dc:      cmovg   rax, rdx
0x182e0:      pop     rbp
0x182e1:      mov     byte ptr [r12 + rax], 0
0x182e6:      mov     rax, r12
0x182e9:      pop     r12
0x182eb:      ret
0x182ec:      nop     dword ptr [rax]
0x182f0:      endbr64
0x182f4:      push    r13
0x182f6:      mov     edx, 1
0x182fb:      xor     esi, esi
0x182fd:      push    r12
0x182ff:      push    rbp
0x18300:      mov     ebp, edi
0x18302:      call    0x4e50
0x18307:      mov     edi, ebp
0x18309:      mov     edx, 2
0x1830e:      xor     esi, esi
0x18310:      mov     r13, rax
0x18313:      call    0x4e50
0x18318:      mov     rsi, r13
0x1831b:      mov     edi, ebp
0x1831d:      xor     edx, edx
0x1831f:      mov     r12, rax

```

```

0x18322:    call    0x4e50
0x18327:    mov     rax, r12
0x1832a:    pop     rbp
0x1832b:    pop     r12
0x1832d:    pop     r13
0x1832f:    ret
0x18330:    endbr64
0x18334:    sub     rsp, 0x18
0x18338:    xor     edi, edi
0x1833a:    mov     rax, qword ptr fs:[0x28]
0x18343:    mov     qword ptr [rsp + 8], rax
0x18348:    xor     eax, eax
0x1834a:    call    0x4be0
0x1834f:    mov     rdi, rsp
0x18352:    mov     qword ptr [rsp], rax
0x18356:    call    0x4840
0x1835b:    mov     rdx, qword ptr [rsp + 8]
0x18360:    sub     rdx, qword ptr fs:[0x28]
0x18369:    jne     0x18370
0x1836b:    add     rsp, 0x18
0x1836f:    ret
0x18370:    call    0x49d0
0x18375:    nop     word ptr cs:[rax + rax]
0x18380:    endbr64
0x18384:    push    r12
0x18386:    push    rbp
0x18387:    mov     rbp, rsi
0x1838a:    push    rbx
0x1838b:    mov     rbx, rdi
0x1838e:    call    0x49b0
0x18393:    mov     rdi, rbp
0x18396:    mov     r12, rax
0x18399:    call    0x49b0
0x1839e:    lea     rdi, [r12 + rax + 2]
0x183a3:    call    0x1d670
0x183a8:    mov     r9, rbp
0x183ab:    mov     r8, rbx
0x183ae:    mov     esi, 1
0x183b3:    mov     r12, rax
0x183b6:    mov     rdi, rax
0x183b9:    xor     eax, eax
0x183bb:    mov     rdx, -1
0x183c2:    lea     rcx, [rip + 0x9be5]
0x183c9:    call    0x4f10
0x183ce:    mov     rdi, r12
0x183d1:    call    0x4b20
0x183d6:    mov     r12d, eax
0x183d9:    cmp     eax, -1

```

```

0x183dc:    je      0x183f0
0x183de:    mov     eax, r12d
0x183e1:    pop     rbx
0x183e2:    pop     rbp
0x183e3:    pop     r12
0x183e5:    ret
0x183e6:    nop     word ptr cs:[rax + rax]
0x183f0:    mov     rcx, qword ptr [rip + 0x124f1]
0x183f7:    mov     rdi, qword ptr [rip + 0x12482]
0x183fe:    mov     r8, rbx
0x18401:    xor     eax, eax
0x18403:    lea     rdx, [rip + 0x9f6e]
0x1840a:    mov     esi, 1
0x1840f:    call    0x4e60
0x18414:    mov     eax, r12d
0x18417:    pop     rbx
0x18418:    pop     rbp
0x18419:    pop     r12
0x1841b:    ret
0x1841c:    nop     dword ptr [rax]
0x18420:    endbr64
0x18424:    push    r13
0x18426:    push    r12
0x18428:    lea     r12, [rip + 0x9b85]
0x1842f:    push    rbp
0x18430:    mov     rdi, r12
0x18433:    push    rbx
0x18434:    sub     rsp, 8
0x18438:    call    0x47e0
0x1843d:    mov     r13, qword ptr [rip + 0x1365c]
0x18444:    test    rax, rax
0x18447:    je      0x184b0
0x18449:    mov     rdi, rax
0x1844c:    mov     rbx, rax
0x1844f:    call    0x49b0
0x18454:    mov     rdi, r13
0x18457:    mov     rbp, rax
0x1845a:    call    0x49b0
0x1845f:    lea     rdi, [rbp + rax + 2]
0x18464:    call    0x1d670
0x18469:    mov     r9, qword ptr [rip + 0x13630]
0x18470:    mov     r8, rbx
0x18473:    lea     rcx, [rip + 0x9b3f]
0x1847a:    mov     rbp, rax
0x1847d:    mov     rdi, rax
0x18480:    mov     esi, 1
0x18485:    xor     eax, eax
0x18487:    mov     rdx, -1

```

```

0x1848e:      call    0x4f10
0x18493:      mov     rdi, r12
0x18496:      mov     rsi, rbp
0x18499:      call    0x18380
0x1849e:      add     rsp, 8
0x184a2:      mov     rdi, rbp
0x184a5:      pop     rbx
0x184a6:      pop     rbp
0x184a7:      pop     r12
0x184a9:      pop     r13
0x184ab:      jmp     0x18d10
0x18d10
0x184b0:      add     rsp, 8
0x184b4:      mov     rsi, r13
0x184b7:      mov     rdi, r12
0x184ba:      pop     rbx
0x184bb:      pop     rbp
0x184bc:      pop     r12
0x184be:      pop     r13
0x184c0:      jmp     0x18380
0x18380
0x184c5:      nop     word ptr cs:[rax + rax]
0x184d0:      endbr64
0x184d4:      test    rsi, rsi
0x184d7:      je      0x18520
0x184d9:      push    rbp
0x184da:      mov     rbp, rsi
0x184dd:      push    rbx
0x184de:      mov     rbx, rdi
0x184e1:      sub     rsp, 8
0x184e5:      call    0x4f00
0x184ea:      mov     rdi, rbx
0x184ed:      lea     rsi, [rbx + rbp]
0x184f1:      nop     dword ptr [rax]
0x184f8:      movsx   rcx, byte ptr [rdi]
0x184fc:      mov     rdx, qword ptr [rax]
0x184ff:      test    byte ptr [rdx + rcx*2 + 1], 0x40
0x18504:      jne     0x18509
0x18506:      mov     byte ptr [rdi], 0x3f
0x18509:      add     rdi, 1
0x1850d:      cmp     rsi, rdi
0x18510:      jne     0x184f8
0x18512:      add     rsp, 8
0x18516:      pop     rbx
0x18517:      pop     rbp
0x18518:      ret
0x18519:      nop     dword ptr [rax]
0x18520:      ret

```

```

0x18521:      nop      word ptr cs:[rax + rax]
0x1852c:      nop      dword ptr [rax]
0x18530:      endbr64
0x18534:      test     rsi, rsi
0x18537:      je       0x185c2
0x1853d:      push     rbp
0x1853e:      mov      rbp, rsi
0x18541:      push     rbx
0x18542:      mov      rbx, rdi
0x18545:      sub      rsp, 8
0x18549:      call     0x4f00
0x1854e:      mov      rdi, rbx
0x18551:      lea      rsi, [rbx + rbp]
0x18555:      movabs   r9, 0x68000000c0000001
0x1855f:      movabs   r8, 0x2400f82000000000
0x18569:      mov      rcx, qword ptr [rax]
0x1856c:      nop      dword ptr [rax]
0x18570:      movsx    rdx, byte ptr [rdi]
0x18574:      mov      rax, rdx
0x18577:      test     byte ptr [rcx + rdx*2], 8
0x1857b:      jne      0x185b0
0x1857d:      cmp      dl, 0x3d
0x18580:      jg       0x185a0
0x18582:      cmp      dl, 0x24
0x18585:      jle      0x1858d
0x18587:      bt       r8, rdx
0x1858b:      jb       0x185b0
0x1858d:      add      rsp, 8
0x18591:      mov      eax, 1
0x18596:      pop      rbx
0x18597:      pop      rbp
0x18598:      ret
0x18599:      nop      dword ptr [rax]
0x185a0:      sub      eax, 0x40
0x185a3:      cmp      al, 0x3e
0x185a5:      ja       0x1858d
0x185a7:      bt       r9, rax
0x185ab:      jae      0x1858d
0x185ad:      nop      dword ptr [rax]
0x185b0:      add      rdi, 1
0x185b4:      cmp      rsi, rdi
0x185b7:      jne      0x18570
0x185b9:      add      rsp, 8
0x185bd:      xor      eax, eax
0x185bf:      pop      rbx
0x185c0:      pop      rbp
0x185c1:      ret
0x185c2:      xor      eax, eax

```

```

0x185c4:      ret
0x185c5:      nop      word ptr cs:[rax + rax]
0x185d0:      endbr64
0x185d4:      push     rbx
0x185d5:      mov      rbx, rdi
0x185d8:      call     0x49b0
0x185dd:      sub      eax, 1
0x185e0:      js       0x18618
0x185e2:      cdqe
0x185e4:      jmp      0x18602
0x18602
0x185e6:      nop      word ptr cs:[rax + rax]
0x185f0:      cmp      dl, 0x20
0x185f3:      je       0x185fa
0x185f5:      cmp      dl, 9
0x185f8:      jne      0x18618
0x185fa:      sub      rax, 1
0x185fe:      test     eax, eax
0x18600:      js       0x18618
0x18602:      movzx    edx, byte ptr [rbx + rax]
0x18606:      cmp      dl, 0x26
0x18609:      jne      0x185f0
0x1860b:      mov      eax, 1
0x18610:      pop      rbx
0x18611:      ret
0x18612:      nop      word ptr [rax + rax]
0x18618:      xor      eax, eax
0x1861a:      pop      rbx
0x1861b:      ret
0x1861c:      nop      dword ptr [rax]
0x18620:      endbr64
0x18624:      jmp      0x1863c
0x1863c
0x18626:      nop      word ptr cs:[rax + rax]
0x18630:      cmp      al, 0x20
0x18632:      je       0x18638
0x18634:      cmp      al, 0x3b
0x18636:      jne      0x18650
0x18638:      add      rdi, 1
0x1863c:      movzx    eax, byte ptr [rdi]
0x1863f:      test     al, al
0x18641:      jne      0x18630
0x18643:      mov      eax, 1
0x18648:      ret
0x18649:      nop      dword ptr [rax]
0x18650:      xor      eax, eax
0x18652:      ret
0x18653:      nop      word ptr cs:[rax + rax]

```

```

0x1865e:      nop
0x18660:      endbr64
0x18664:      push    r15
0x18666:      push    r14
0x18668:      push    r13
0x1866a:      push    r12
0x1866c:      push    rbp
0x1866d:      push    rbx
0x1866e:      sub     rsp, 0x98
0x18675:      mov     rax, qword ptr fs:[0x28]
0x1867e:      mov     qword ptr [rsp + 0x88], rax
0x18686:      xor     eax, eax
0x18688:      lea     r15, [rsp + 0x10]
0x1868d:      lea     r13, [rsp + 0x30]
0x18692:      mov     qword ptr [rsp + 8], 0
0x1869b:      mov     qword ptr [rsp], 0
0x186a3:      jmp     0x186d3
0x186d3:
0x186a5:      nop     dword ptr [rax]
0x186a8:      mov     rax, qword ptr [rsp]
0x186ac:      mov     qword ptr [rax + 0x18], r14
0x186b0:      mov     rdi, r13
0x186b3:      call    0x1d7a0
0x186b8:      mov     dword ptr [r14 + 8], r12d
0x186bc:      mov     qword ptr [r14], rax
0x186bf:      mov     dword ptr [r14 + 0xc], ebp
0x186c3:      mov     dword ptr [r14 + 0x10], ebx
0x186c7:      mov     qword ptr [r14 + 0x18], 0
0x186cf:      mov     qword ptr [rsp], r14
0x186d3:      xor     ecx, ecx
0x186d5:      mov     edx, 3
0x186da:      mov     rsi, r15
0x186dd:      mov     rdi, r13
0x186e0:      call    0x17320
0x186e5:      cmp     byte ptr [rsp + 0x30], 0
0x186ea:      je      0x18758
0x186ec:      mov     rdi, qword ptr [rsp + 0x10]
0x186f1:      mov     r12d, 0xffffffff
0x186f7:      test    rdi, rdi
0x186fa:      je      0x18704
0x186fc:      call    0xbd00
0x18701:      mov     r12d, eax
0x18704:      mov     rdi, qword ptr [rsp + 0x18]
0x18709:      mov     ebp, 0xffffffff
0x1870e:      test    rdi, rdi
0x18711:      je      0x1871a
0x18713:      call    0xbd00
0x18718:      mov     ebp, eax

```



```

0x1871a:    mov     rdi, qword ptr [rsp + 0x20]
0x1871f:    mov     ebx, 0xffffffff
0x18724:    test    rdi, rdi
0x18727:    je      0x18730
0x18729:    call    0xbd00
0x1872e:    mov     ebx, eax
0x18730:    mov     edi, 0x20
0x18735:    call    0x1d670
0x1873a:    cmp     qword ptr [rsp + 8], 0
0x18740:    mov     r14, rax
0x18743:    jne     0x186a8
0x18749:    mov     qword ptr [rsp + 8], rax
0x1874e:    jmp     0x186b0
0x186b0:
0x18753:    nop     dword ptr [rax + rax]
0x18758:    mov     rdx, qword ptr [rsp + 8]
0x1875d:    test    rdx, rdx
0x18760:    je      0x1877d
0x18762:    mov     rax, qword ptr [rip + 0x13327]
0x18769:    test    rax, rax
0x1876c:    je      0x187a8
0x1876e:    mov     rcx, qword ptr [rsp]
0x18772:    mov     qword ptr [rip + 0x13317], rdx
0x18779:    mov     qword ptr [rcx + 0x18], rax
0x1877d:    mov     rax, qword ptr [rsp + 0x88]
0x18785:    sub     rax, qword ptr fs:[0x28]
0x1878e:    jne     0x187b6
0x18790:    add     rsp, 0x98
0x18797:    pop     rbx
0x18798:    pop     rbp
0x18799:    pop     r12
0x1879b:    pop     r13
0x1879d:    pop     r14
0x1879f:    pop     r15
0x187a1:    ret
0x187a2:    nop     word ptr [rax + rax]
0x187a8:    mov     rax, qword ptr [rsp + 8]
0x187ad:    mov     qword ptr [rip + 0x132dc], rax
0x187b4:    jmp     0x1877d
0x1877d:
0x187b6:    call    0x49d0
0x187bb:    nop     dword ptr [rax + rax]
0x187c0:    endbr64
0x187c4:    sub     rsp, 8
0x187c8:    lea     rsi, [rip - 0xf]
0x187cf:    call    0x4b50
0x187d4:    mov     dword ptr [rip + 0x132d2], 1
0x187de:    add     rsp, 8

```

```

0x187e2:      ret
0x187e3:      nop      word ptr cs:[rax + rax]
0x187ee:      nop
0x187f0:      cmp      edi, 0x14
0x187f3:      je       0x18850
0x187f5:      jg       0x18818
0x187f7:      cmp      edi, 0xe
0x187fa:      je       0x18838
0x187fc:      cmp      edi, 0x12
0x187ff:      jne      0x18861
0x18801:      lea      rsi, [rip + 0x148]
0x18808:      mov      edi, 0x12
0x1880d:      jmp      0x4b50
0x4b50
0x18812:      nop      word ptr [rax + rax]
0x18818:      cmp      edi, 0x1c
0x1881b:      jne      0x18830
0x1881d:      lea      rsi, [rip + 0x1cc]
0x18824:      mov      edi, 0x1c
0x18829:      jmp      0x4b50
0x4b50
0x1882e:      nop
0x18830:      ret
0x18831:      nop      dword ptr [rax]
0x18838:      lea      rsi, [rip + 0x31]
0x1883f:      mov      edi, 0xe
0x18844:      jmp      0x4b50
0x4b50
0x18849:      nop      dword ptr [rax]
0x18850:      lea      rsi, [rip + 0x89]
0x18857:      mov      edi, 0x14
0x1885c:      jmp      0x4b50
0x4b50
0x18861:      ret
0x18862:      nop      word ptr cs:[rax + rax]
0x1886d:      nop      dword ptr [rax]
0x18870:      endbr64
0x18874:      mov      edx, dword ptr [rip + 0x1323a]
0x1887a:      push     rbp
0x1887b:      mov      eax, 1
0x18880:      mov      ebp, edi
0x18882:      test     edx, edx
0x18884:      jne      0x188b0
0x18886:      mov      edi, ebp
0x18888:      mov      dword ptr [rip + 0x1322e], eax
0x1888e:      call     0x187f0
0x18893:      xor      eax, eax
0x18895:      call     0x18330

```

```

0x1889a:      mov     edi, 0x3c
0x1889f:      pop     rbp
0x188a0:      sub     edi, dword ptr [rax]
0x188a2:      jmp     0x4aa0
0x4aa0
0x188a7:      nop     word ptr [rax + rax]
0x188b0:      xor     eax, eax
0x188b2:      call    0x18330
0x188b7:      mov     eax, dword ptr [rax]
0x188b9:      test    eax, eax
0x188bb:      je      0x188d0
0x188bd:      xor     eax, eax
0x188bf:      call    0xb4e0
0x188c4:      xor     eax, eax
0x188c6:      jmp     0x18886
0x18886
0x188c8:      nop     dword ptr [rax + rax]
0x188d0:      mov     edi, ebp
0x188d2:      call    0x16fd0
0x188d7:      jmp     0x188bd
0x188bd
0x188d9:      nop     dword ptr [rax]
0x188e0:      endbr64
0x188e4:      mov     eax, dword ptr [rip + 0x131ca]
0x188ea:      push    rbp
0x188eb:      mov     ebp, edi
0x188ed:      test    eax, eax
0x188ef:      jne     0x18908
0x188f1:      mov     eax, 1
0x188f6:      mov     edi, ebp
0x188f8:      pop     rbp
0x188f9:      mov     dword ptr [rip + 0x131c5], eax
0x188ff:      jmp     0x187f0
0x187f0
0x18904:      nop     dword ptr [rax]
0x18908:      xor     eax, eax
0x1890a:      call    0x9c90
0x1890f:      cmp     eax, 1
0x18912:      sete    al
0x18915:      movzx   eax, al
0x18918:      mov     dword ptr [rip + 0x1319a], eax
0x1891e:      xor     eax, eax
0x18920:      call    0x9610
0x18925:      call    0x4920
0x1892a:      mov     esi, 0x13
0x1892f:      mov     edi, eax
0x18931:      call    0x4bd0
0x18936:      xor     eax, eax

```

```

0x18938:      mov     edi, ebp
0x1893a:      pop     rbp
0x1893b:      mov     dword ptr [rip + 0x13183], eax
0x18941:      jmp     0x187f0
0x187f0
0x18946:      nop     word ptr cs:[rax + rax]
0x18950:      endbr64
0x18954:      mov     eax, dword ptr [rip + 0x1315e]
0x1895a:      push    rbx
0x1895b:      mov     ebx, edi
0x1895d:      test    eax, eax
0x1895f:      jne     0x189a0
0x18961:      cmp     ebx, 0x14
0x18964:      je      0x189b4
0x18966:      jg      0x18988
0x18968:      cmp     ebx, 0xe
0x1896b:      je      0x189d8
0x1896d:      cmp     ebx, 0x12
0x18970:      jne     0x189d0
0x18972:      lea     rsi, [rip - 0x29]
0x18979:      mov     edi, 0x12
0x1897e:      pop     rbx
0x1897f:      jmp     0x4b50
0x4b50
0x18984:      nop     dword ptr [rax]
0x18988:      cmp     ebx, 0x1c
0x1898b:      jne     0x189d0
0x1898d:      lea     rsi, [rip + 0x5c]
0x18994:      mov     edi, 0x1c
0x18999:      pop     rbx
0x1899a:      jmp     0x4b50
0x4b50
0x1899f:      nop
0x189a0:      call    0x7bd0
0x189a5:      mov     dword ptr [rip + 0x13111], 0
0x189af:      cmp     ebx, 0x14
0x189b2:      jne     0x18966
0x189b4:      lea     rsi, [rip - 0xdb]
0x189bb:      mov     edi, 0x14
0x189c0:      pop     rbx
0x189c1:      jmp     0x4b50
0x4b50
0x189c6:      nop     word ptr cs:[rax + rax]
0x189d0:      pop     rbx
0x189d1:      ret
0x189d2:      nop     word ptr [rax + rax]
0x189d8:      lea     rsi, [rip - 0x16f]
0x189df:      mov     edi, 0xe

```

```

0x189e4:      pop     rbx
0x189e5:      jmp     0x4b50
0x4b50
0x189ea:      nop     word ptr [rax + rax]
0x189f0:      endbr64
0x189f4:      mov     eax, dword ptr [rip + 0x130ba]
0x189fa:      push    rbx
0x189fb:      mov     ebx, edi
0x189fd:      test    eax, eax
0x189ff:      jne     0x18a50
0x18a01:      mov     eax, 1
0x18a06:      mov     dword ptr [rip + 0x130b4], eax
0x18a0c:      cmp     ebx, 0x14
0x18a0f:      je      0x18a72
0x18a11:      jg      0x18a30
0x18a13:      cmp     ebx, 0xe
0x18a16:      je      0x18a90
0x18a18:      cmp     ebx, 0x12
0x18a1b:      jne     0x18a88
0x18a1d:      lea     rsi, [rip - 0xd4]
0x18a24:      mov     edi, 0x12
0x18a29:      pop     rbx
0x18a2a:      jmp     0x4b50
0x4b50
0x18a2f:      nop
0x18a30:      cmp     ebx, 0x1c
0x18a33:      jne     0x18a88
0x18a35:      lea     rsi, [rip - 0x4c]
0x18a3c:      mov     edi, 0x1c
0x18a41:      pop     rbx
0x18a42:      jmp     0x4b50
0x4b50
0x18a47:      nop     word ptr [rax + rax]
0x18a50:      xor     eax, eax
0x18a52:      call    0xa6c0
0x18a57:      xor     eax, eax
0x18a59:      call    0x9d80
0x18a5e:      mov     edi, ebx
0x18a60:      call    0x7bd0
0x18a65:      xor     eax, eax
0x18a67:      mov     dword ptr [rip + 0x13053], eax
0x18a6d:      cmp     ebx, 0x14
0x18a70:      jne     0x18a11
0x18a72:      lea     rsi, [rip - 0x199]
0x18a79:      mov     edi, 0x14
0x18a7e:      pop     rbx
0x18a7f:      jmp     0x4b50
0x4b50

```

```

0x18a84:      nop      dword ptr [rax]
0x18a88:      pop      rbx
0x18a89:      ret
0x18a8a:      nop      word ptr [rax + rax]
0x18a90:      lea      rsi, [rip - 0x227]
0x18a97:      mov      edi, 0xe
0x18a9c:      pop      rbx
0x18a9d:      jmp      0x4b50
0x4b50
0x18aa2:      nop      word ptr cs:[rax + rax]
0x18aad:      nop      dword ptr [rax]
0x18ab0:      endbr64
0x18ab4:      sub      rsp, 8
0x18ab8:      mov      ecx, dword ptr [rip + 0x13006]
0x18abe:      test     ecx, ecx
0x18ac0:      jne      0x18af0
0x18ac2:      mov      edx, dword ptr [rip + 0x12ff8]
0x18ac8:      test     edx, edx
0x18aca:      jne      0x18b30
0x18acc:      mov      eax, dword ptr [rip + 0x12fea]
0x18ad2:      test     eax, eax
0x18ad4:      jne      0x18ae0
0x18ad6:      add      rsp, 8
0x18ada:      ret
0x18adb:      nop      dword ptr [rax + rax]
0x18ae0:      mov      dword ptr [rip + 0x12fd2], 0
0x18aea:      add      rsp, 8
0x18aee:      ret
0x18aef:      nop
0x18af0:      xor      eax, eax
0x18af2:      call     0x9610
0x18af7:      call     0x4920
0x18afc:      mov      esi, 0x13
0x18b01:      mov      edi, eax
0x18b03:      call     0x4bd0
0x18b08:      mov      dword ptr [rip + 0x12fb2], 0
0x18b12:      mov      dword ptr [rip + 0x12fa4], 0
0x18b1c:      mov      dword ptr [rip + 0x12f96], 0
0x18b26:      add      rsp, 8
0x18b2a:      ret
0x18b2b:      nop      dword ptr [rax + rax]
0x18b30:      xor      eax, eax
0x18b32:      call     0xa6c0
0x18b37:      xor      eax, eax
0x18b39:      call     0x9d80
0x18b3e:      mov      edi, 0x1c
0x18b43:      call     0x7bd0
0x18b48:      mov      dword ptr [rip + 0x12f6e], 0

```

```

0x18b52:    mov     dword ptr [rip + 0x12f60], 0
0x18b5c:    add     rsp, 8
0x18b60:    ret
0x18b61:    nop     word ptr cs:[rax + rax]
0x18b6c:    nop     dword ptr [rax]
0x18b70:    endbr64
0x18b74:    mov     dword ptr [rip + 0x12f3a], edi
0x18b7a:    test    edi, edi
0x18b7c:    jne     0x18b80
0x18b7e:    ret
0x18b7f:    nop
0x18b80:    xor     eax, eax
0x18b82:    jmp     0x18ab0
0x18ab0:
0x18b87:    nop     word ptr [rax + rax]
0x18b90:    endbr64
0x18b94:    sub     rsp, 8
0x18b98:    mov     eax, dword ptr [rip + 0x11bfa]
0x18b9e:    cmp     edi, 1
0x18ba1:    je      0x18bf0
0x18ba3:    test    eax, eax
0x18ba5:    jne     0x18bd0
0x18ba7:    mov     esi, 1
0x18bac:    mov     edi, 0x1c
0x18bb1:    call    0x4b50
0x18bb6:    mov     esi, 1
0x18bbb:    mov     edi, 0xe
0x18bc0:    add     rsp, 8
0x18bc4:    jmp     0x4b50
0x4b50:
0x18bc9:    nop     dword ptr [rax]
0x18bd0:    mov     esi, 1
0x18bd5:    mov     edi, 0x14
0x18bda:    call    0x4b50
0x18bdf:    mov     esi, 1
0x18be4:    mov     edi, 0x12
0x18be9:    call    0x4b50
0x18bee:    jmp     0x18ba7
0x18ba7:
0x18bf0:    test    eax, eax
0x18bf2:    jne     0x18c20
0x18bf4:    lea     rsi, [rip - 0x20b]
0x18bfb:    mov     edi, 0x1c
0x18c00:    call    0x4b50
0x18c05:    lea     rsi, [rip - 0x39c]
0x18c0c:    mov     edi, 0xe
0x18c11:    add     rsp, 8
0x18c15:    jmp     0x4b50

```

```

0x4b50
0x18c1a:      nop      word ptr [rax + rax]
0x18c20:      lea      rsi, [rip - 0x347]
0x18c27:      mov      edi, 0x14
0x18c2c:      call     0x4b50
0x18c31:      lea      rsi, [rip - 0x2e8]
0x18c38:      mov      edi, 0x12
0x18c3d:      call     0x4b50
0x18c42:      jmp      0x18bf4
0x18bf4
0x18c44:      nop      word ptr cs:[rax + rax]
0x18c4f:      nop
0x18c50:      endbr64
0x18c54:      push     rbp
0x18c55:      mov      esi, 1
0x18c5a:      mov      edi, 0x14
0x18c5f:      call     0x4b50
0x18c64:      mov      esi, 1
0x18c69:      mov      edi, 0x12
0x18c6e:      cmp      rax, 1
0x18c72:      setne    al
0x18c75:      movzx    eax, al
0x18c78:      mov      dword ptr [rip + 0x11b1a], eax
0x18c7e:      call     0x4b50
0x18c83:      mov      esi, 1
0x18c88:      mov      edi, 0x1c
0x18c8d:      call     0x4b50
0x18c92:      lea      rbp, [rip - 0x1519]
0x18c99:      mov      edi, 0xb
0x18c9e:      mov      rsi, rbp
0x18ca1:      call     0x4b50
0x18ca6:      mov      rsi, rbp
0x18ca9:      mov      edi, 1
0x18cae:      call     0x4b50
0x18cb3:      mov      rsi, rbp
0x18cb6:      mov      edi, 0xf
0x18cbb:      call     0x4b50
0x18cc0:      mov      esi, 1
0x18cc5:      mov      edi, 3
0x18cca:      call     0x4b50
0x18ccf:      lea      rsi, [rip - 0x516]
0x18cd6:      mov      edi, 2
0x18cdb:      pop      rbp
0x18cdc:      jmp      0x4b50
0x4b50
0x18ce1:      nop      word ptr cs:[rax + rax]
0x18ceb:      nop      dword ptr [rax + rax]
0x18cf0:      endbr64

```



```

0x18cf4:    push    rax
0x18cf5:    pop     rax
0x18cf6:    lea     rdi, [rip + 0x96a8]
0x18cfd:    sub     rsp, 8
0x18d01:    call    0x8b00
0x18d06:    call    0x4850
0x18d0b:    nop     dword ptr [rax + rax]
0x18d10:    endbr64
0x18d14:    test    rdi, rdi
0x18d17:    je      0x18d20
0x18d19:    jmp     0x4800
0x4800
0x18d1e:    nop
0x18d20:    lea     rdi, [rip + 0x9697]
0x18d27:    jmp     0x8b00
0x8b00
0x18d2c:    nop     dword ptr [rax]
0x18d30:    endbr64
0x18d34:    xor     r8d, r8d
0x18d37:    test    rdx, rdx
0x18d3a:    je      0x18d90
0x18d3c:    push    r14
0x18d3e:    push    r13
0x18d40:    push    r12
0x18d42:    mov     r12, rsi
0x18d45:    push    rbp
0x18d46:    mov     ebp, edi
0x18d48:    push    rbx
0x18d49:    mov     rbx, rdx
0x18d4c:    call    0x4860
0x18d51:    mov     r13d, dword ptr [rax]
0x18d54:    mov     r14, rax
0x18d57:    jmp     0x18d66
0x18d66
0x18d59:    nop     dword ptr [rax]
0x18d60:    cmp     dword ptr [r14], 4
0x18d64:    jne     0x18d7d
0x18d66:    mov     dword ptr [r14], r13d
0x18d69:    mov     rdx, rbx
0x18d6c:    mov     rsi, r12
0x18d6f:    mov     edi, ebp
0x18d71:    call    0x47a0
0x18d76:    mov     r8d, eax
0x18d79:    test    eax, eax
0x18d7b:    js      0x18d60
0x18d7d:    pop     rbx
0x18d7e:    mov     eax, r8d
0x18d81:    pop     rbp

```

```

0x18d82:    pop     r12
0x18d84:    pop     r13
0x18d86:    pop     r14
0x18d88:    ret
0x18d89:    nop     dword ptr [rax]
0x18d90:    mov     eax, r8d
0x18d93:    ret
0x18d94:    nop     word ptr cs:[rax + rax]
0x18d9f:    nop
0x18da0:    endbr64
0x18da4:    xor     r8d, r8d
0x18da7:    test    rdx, rdx
0x18daa:    je      0x18e00
0x18dac:    push    r14
0x18dae:    push    r13
0x18db0:    push    r12
0x18db2:    mov     r12, rsi
0x18db5:    push    rbp
0x18db6:    mov     ebp, edi
0x18db8:    push    rbx
0x18db9:    mov     rbx, rdx
0x18dbc:    call    0x4860
0x18dc1:    mov     r13d, dword ptr [rax]
0x18dc4:    mov     r14, rax
0x18dc7:    jmp     0x18dd6
0x18dd6:
0x18dc9:    nop     dword ptr [rax]
0x18dd0:    cmp     dword ptr [r14], 4
0x18dd4:    jne     0x18ded
0x18dd6:    mov     dword ptr [r14], r13d
0x18dd9:    mov     rdx, rbx
0x18ddc:    mov     rsi, r12
0x18ddf:    mov     edi, ebp
0x18de1:    call    0x4790
0x18de6:    mov     r8d, eax
0x18de9:    test    eax, eax
0x18deb:    js      0x18dd0
0x18ded:    pop     rbx
0x18dee:    mov     eax, r8d
0x18df1:    pop     rbp
0x18df2:    pop     r12
0x18df4:    pop     r13
0x18df6:    pop     r14
0x18df8:    ret
0x18df9:    nop     dword ptr [rax]
0x18e00:    mov     eax, r8d
0x18e03:    ret
0x18e04:    nop     word ptr cs:[rax + rax]

```

```

0x18e0f:      nop
0x18e10:      endbr64
0x18e14:      xor     r8d, r8d
0x18e17:      test    rdx, rdx
0x18e1a:      je      0x18e70
0x18e1c:      push     r14
0x18e1e:      push     r13
0x18e20:      push     r12
0x18e22:      mov      r12, rsi
0x18e25:      push     rbp
0x18e26:      mov      rbp, rdi
0x18e29:      push     rbx
0x18e2a:      mov      rbx, rdx
0x18e2d:      call     0x4860
0x18e32:      mov      r13d, dword ptr [rax]
0x18e35:      mov      r14, rax
0x18e38:      jmp      0x18e46
0x18e46:
0x18e3a:      nop      word ptr [rax + rax]
0x18e40:      cmp      dword ptr [r14], 4
0x18e44:      jne      0x18e5e
0x18e46:      mov      dword ptr [r14], r13d
0x18e49:      mov      rdx, rbx
0x18e4c:      mov      rsi, r12
0x18e4f:      mov      rdi, rbp
0x18e52:      call     0x4910
0x18e57:      mov      r8d, eax
0x18e5a:      test     eax, eax
0x18e5c:      js       0x18e40
0x18e5e:      pop      rbx
0x18e5f:      mov      eax, r8d
0x18e62:      pop      rbp
0x18e63:      pop      r12
0x18e65:      pop      r13
0x18e67:      pop      r14
0x18e69:      ret
0x18e6a:      nop      word ptr [rax + rax]
0x18e70:      mov      eax, r8d
0x18e73:      ret
0x18e74:      nop      word ptr cs:[rax + rax]
0x18e7f:      nop
0x18e80:      endbr64
0x18e84:      push     r13
0x18e86:      mov      r13, rdi
0x18e89:      push     r12
0x18e8b:      push     rbp
0x18e8c:      push     rbx
0x18e8d:      mov      ebx, 0x64

```

```

0x18e92:      sub     rsp, 0x18
0x18e96:      jmp     0x18eaa
0x18eaa:
0x18e98:      nop     dword ptr [rax + rax]
0x18ea0:      mov     rdi, rbp
0x18ea3:      add     ebx, ebx
0x18ea5:      call    0x18d10
0x18eaa:      movsxd  r12, ebx
0x18ead:      mov     rdi, r12
0x18eb0:      call    0x1d670
0x18eb5:      mov     rdx, r12
0x18eb8:      mov     rdi, r13
0x18ebb:      mov     rbp, rax
0x18ebe:      mov     rsi, rax
0x18ec1:      xor     eax, eax
0x18ec3:      call    0x18e10
0x18ec8:      cmp     ebx, eax
0x18eca:      jle     0x18ea0
0x18ecc:      mov     rdi, rbp
0x18ecf:      mov     dword ptr [rsp + 0xc], eax
0x18ed3:      call    0x18d10
0x18ed8:      mov     eax, dword ptr [rsp + 0xc]
0x18edc:      add     rsp, 0x18
0x18ee0:      pop     rbx
0x18ee1:      pop     rbp
0x18ee2:      pop     r12
0x18ee4:      pop     r13
0x18ee6:      ret
0x18ee7:      nop     word ptr [rax + rax]
0x18ef0:      endbr64
0x18ef4:      push    r13
0x18ef6:      push    r12
0x18ef8:      mov     r12d, edi
0x18efb:      push    rbp
0x18efc:      mov     rbp, rsi
0x18eff:      push    rbx
0x18f00:      sub     rsp, 8
0x18f04:      call    0x4860
0x18f09:      mov     r13d, dword ptr [rax]
0x18f0c:      mov     rbx, rax
0x18f0f:      jmp     0x18f1d
0x18f1d:
0x18f11:      nop     dword ptr [rax]
0x18f18:      cmp     dword ptr [rbx], 4
0x18f1b:      jne     0x18f2f
0x18f1d:      mov     dword ptr [rbx], r13d
0x18f20:      mov     rsi, rbp
0x18f23:      mov     edi, r12d

```

```

0x18f26:      call    0x4eb0
0x18f2b:      test    eax, eax
0x18f2d:      js      0x18f18
0x18f2f:      add     rsp, 8
0x18f33:      pop     rbx
0x18f34:      pop     rbp
0x18f35:      pop     r12
0x18f37:      pop     r13
0x18f39:      ret
0x18f3a:      nop     word ptr [rax + rax]
0x18f40:      endbr64
0x18f44:      push    r13
0x18f46:      push    r12
0x18f48:      mov     r12, rdi
0x18f4b:      push    rbp
0x18f4c:      mov     rbp, rsi
0x18f4f:      push    rbx
0x18f50:      sub     rsp, 8
0x18f54:      call    0x4860
0x18f59:      mov     r13d, dword ptr [rax]
0x18f5c:      mov     rbx, rax
0x18f5f:      jmp     0x18f6d
0x18f6d:
0x18f61:      nop     dword ptr [rax]
0x18f68:      cmp     dword ptr [rbx], 4
0x18f6b:      jne     0x18f7f
0x18f6d:      mov     dword ptr [rbx], r13d
0x18f70:      mov     rsi, rbp
0x18f73:      mov     rdi, r12
0x18f76:      call    0x4b90
0x18f7b:      test    eax, eax
0x18f7d:      js      0x18f68
0x18f7f:      add     rsp, 8
0x18f83:      pop     rbx
0x18f84:      pop     rbp
0x18f85:      pop     r12
0x18f87:      pop     r13
0x18f89:      ret
0x18f8a:      nop     word ptr [rax + rax]
0x18f90:      endbr64
0x18f94:      push    r13
0x18f96:      push    r12
0x18f98:      mov     r12, rdi
0x18f9b:      push    rbp
0x18f9c:      mov     rbp, rsi
0x18f9f:      push    rbx
0x18fa0:      sub     rsp, 8
0x18fa4:      call    0x4860

```

```

0x18fa9:      mov     r13d, dword ptr [rax]
0x18fac:      mov     rbx, rax
0x18faf:      jmp     0x18fbd
0x18fbd:
0x18fb1:      nop     dword ptr [rax]
0x18fb8:      cmp     dword ptr [rbx], 4
0x18fbb:      jne     0x18fcf
0x18fbd:      mov     dword ptr [rbx], r13d
0x18fc0:      mov     rsi, rbp
0x18fc3:      mov     rdi, r12
0x18fc6:      call    0x4ad0
0x18fcb:      test    eax, eax
0x18fcd:      js      0x18fb8
0x18fcf:      add     rsp, 8
0x18fd3:      pop     rbx
0x18fd4:      pop     rbp
0x18fd5:      pop     r12
0x18fd7:      pop     r13
0x18fd9:      ret
0x18fda:      nop     word ptr [rax + rax]
0x18fe0:      endbr64
0x18fe4:      push    r12
0x18fe6:      push    rbp
0x18fe7:      push    rbx
0x18fe8:      mov     ebx, 0x40
0x18fed:      call    0x4860
0x18ff2:      mov     edi, 0x40
0x18ff7:      mov     dword ptr [rax], 0
0x18ffd:      mov     r12, rax
0x19000:      call    0x1d670
0x19005:      mov     rbp, rax
0x19008:      jmp     0x19031
0x19031:
0x1900a:      nop     word ptr [rax + rax]
0x19010:      cmp     dword ptr [r12], 0x22
0x19015:      jne     0x19050
0x19017:      add     rbx, 0x40
0x1901b:      mov     rdi, rbp
0x1901e:      mov     rsi, rbx
0x19021:      call    0x1d690
0x19026:      mov     dword ptr [r12], 0
0x1902e:      mov     rbp, rax
0x19031:      mov     rsi, rbx
0x19034:      mov     rdi, rbp
0x19037:      call    0x4a80
0x1903c:      test    rax, rax
0x1903f:      je      0x19010
0x19041:      mov     rax, rbp

```

```

0x19044:    pop     rbx
0x19045:    pop     rbp
0x19046:    pop     r12
0x19048:    ret
0x19049:    nop     dword ptr [rax]
0x19050:    mov     rdi, rbp
0x19053:    xor     ebp, ebp
0x19055:    call    0x18d10
0x1905a:    mov     rax, rbp
0x1905d:    pop     rbx
0x1905e:    pop     rbp
0x1905f:    pop     r12
0x19061:    ret
0x19062:    nop     word ptr cs:[rax + rax]
0x1906d:    nop     dword ptr [rax]
0x19070:    endbr64
0x19074:    push    rbp
0x19075:    mov     esi, 0x2f
0x1907a:    mov     rbp, rdi
0x1907d:    push    rbx
0x1907e:    sub     rsp, 8
0x19082:    call    0x4a20
0x19087:    test    rax, rax
0x1908a:    je      0x190a8
0x1908c:    mov     rbx, rax
0x1908f:    mov     rdi, rbp
0x19092:    call    0x1d7a0
0x19097:    sub     rbx, rbp
0x1909a:    mov     byte ptr [rax + rbx], 0x30
0x1909e:    add     rsp, 8
0x190a2:    pop     rbx
0x190a3:    pop     rbp
0x190a4:    ret
0x190a5:    nop     dword ptr [rax]
0x190a8:    add     rsp, 8
0x190ac:    lea     rdi, [rip + 0x862b]
0x190b3:    pop     rbx
0x190b4:    pop     rbp
0x190b5:    jmp     0x1d7a0
0x1d7a0
0x190ba:    nop     word ptr [rax + rax]
0x190c0:    endbr64
0x190c4:    push    rbx
0x190c5:    mov     rbx, rdi
0x190c8:    call    0x49b0
0x190cd:    lea     rax, [rbx + rax - 1]
0x190d2:    cmp     byte ptr [rax], 0x2f
0x190d5:    jne     0x190da

```

```

0x190d7:    mov     byte ptr [rax], 0
0x190da:    mov     rdi, rbx
0x190dd:    mov     esi, 0x2f
0x190e2:    call    0x4a20
0x190e7:    lea     rdx, [rax + 1]
0x190eb:    test    rax, rax
0x190ee:    mov     rax, rbx
0x190f1:    pop     rbx
0x190f2:    cmovne  rax, rdx
0x190f6:    ret
0x190f7:    nop     word ptr [rax + rax]
0x19100:    endbr64
0x19104:    push    rbx
0x19105:    mov     esi, 0x10
0x1910a:    mov     ebx, edi
0x1910c:    mov     edi, 1
0x19111:    call    0x1d740
0x19116:    mov     dword ptr [rax + 8], ebx
0x19119:    pop     rbx
0x1911a:    ret
0x1911b:    nop     dword ptr [rax + rax]
0x19120:    endbr64
0x19124:    test    rdi, rdi
0x19127:    je      0x19148
0x19129:    push    rbp
0x1912a:    mov     rbp, rdi
0x1912d:    mov     rdi, qword ptr [rdi]
0x19130:    test    rdi, rdi
0x19133:    je      0x1913a
0x19135:    call    0x18d10
0x1913a:    mov     rdi, rbp
0x1913d:    pop     rbp
0x1913e:    jmp     0x18d10
0x18d10:
0x19143:    nop     dword ptr [rax + rax]
0x19148:    ret
0x19149:    nop     dword ptr [rax]
0x19150:    endbr64
0x19154:    push    rbp
0x19155:    mov     rbp, rsi
0x19158:    push    rbx
0x19159:    mov     rbx, rdi
0x1915c:    sub     rsp, 8
0x19160:    mov     eax, dword ptr [rdi + 0xc]
0x19163:    lea     esi, [rax + 1]
0x19166:    mov     dword ptr [rdi + 0xc], esi
0x19169:    imul    esi, dword ptr [rdi + 8]
0x1916d:    mov     rdi, qword ptr [rdi]

```



```

0x19170:      movsxd  rsi, esi
0x19173:      call   0x1d690
0x19178:      mov     ecx, dword ptr [rbx + 0xc]
0x1917b:      movsxd  rdx, dword ptr [rbx + 8]
0x1917f:      mov     rsi, rbp
0x19182:      mov     qword ptr [rbx], rax
0x19185:      add     rsp, 8
0x19189:      sub     ecx, 1
0x1918c:      pop     rbx
0x1918d:      pop     rbp
0x1918e:      imul    ecx, edx
0x19191:      movsxd  rcx, ecx
0x19194:      lea     rdi, [rax + rcx]
0x19198:      jmp     0x4bb0
0x4bb0
0x1919d:      nop     dword ptr [rax]
0x191a0:      endbr64
0x191a4:      mov     eax, dword ptr [rdi + 0xc]
0x191a7:      test    eax, eax
0x191a9:      je      0x191f8
0x191ab:      push    rbp
0x191ac:      sub     eax, 1
0x191af:      mov     rbp, rsi
0x191b2:      push    rbx
0x191b3:      mov     rbx, rdi
0x191b6:      sub     rsp, 8
0x191ba:      movsxd  rdx, dword ptr [rdi + 8]
0x191be:      mov     dword ptr [rdi + 0xc], eax
0x191c1:      mov     rsi, rdx
0x191c4:      imul    esi, eax
0x191c7:      movsxd  rsi, esi
0x191ca:      add     rsi, qword ptr [rdi]
0x191cd:      mov     rdi, rbp
0x191d0:      call   0x4bb0
0x191d5:      mov     esi, dword ptr [rbx + 0xc]
0x191d8:      imul    esi, dword ptr [rbx + 8]
0x191dc:      mov     rdi, qword ptr [rbx]
0x191df:      movsxd  rsi, esi
0x191e2:      call   0x1d690
0x191e7:      mov     qword ptr [rbx], rax
0x191ea:      add     rsp, 8
0x191ee:      mov     rax, rbp
0x191f1:      pop     rbx
0x191f2:      pop     rbp
0x191f3:      ret
0x191f4:      nop     dword ptr [rax]
0x191f8:      xor     eax, eax
0x191fa:      ret

```

```

0x191fb:      nop      dword ptr [rax + rax]
0x19200:      endbr64
0x19204:      mov      ecx, dword ptr [rdi + 0xc]
0x19207:      cmp      ecx, edx
0x19209:      jl       0x19240
0x1920b:      test     ecx, ecx
0x1920d:      je       0x19240
0x1920f:      sub      rsp, 8
0x19213:      movsxd   r9, dword ptr [rdi + 8]
0x19217:      sub      ecx, edx
0x19219:      mov      r8, rsi
0x1921c:      imul     ecx, r9d
0x19220:      mov      rdx, r9
0x19223:      movsxd   rsi, ecx
0x19226:      add      rsi, qword ptr [rdi]
0x19229:      mov      rdi, r8
0x1922c:      call     0x4bb0
0x19231:      add      rsp, 8
0x19235:      ret
0x19236:      nop      word ptr cs:[rax + rax]
0x19240:      xor      eax, eax
0x19242:      ret
0x19243:      nop      word ptr cs:[rax + rax]
0x1924e:      nop
0x19250:      endbr64
0x19254:      push     rbx
0x19255:      mov      rbx, rdi
0x19258:      mov      dword ptr [rdi + 0xc], esi
0x1925b:      imul     esi, dword ptr [rdi + 8]
0x1925f:      mov      rdi, qword ptr [rdi]
0x19262:      movsxd   rsi, esi
0x19265:      call     0x1d690
0x1926a:      mov      qword ptr [rbx], rax
0x1926d:      pop      rbx
0x1926e:      ret
0x1926f:      nop
0x19270:      endbr64
0x19274:      mov      eax, dword ptr [rdi + 0xc]
0x19277:      ret
0x19278:      nop      dword ptr [rax + rax]
0x19280:      push     r12
0x19282:      mov      r12, rdi
0x19285:      mov      edi, 2
0x1928a:      push     rbp
0x1928b:      push     rbx
0x1928c:      mov      ebx, esi
0x1928e:      xor      esi, esi
0x19290:      call     0x4b50

```

```

0x19295:      xor     esi, esi
0x19297:      mov     edi, 3
0x1929c:      call    0x4b50
0x192a1:      test    ebx, ebx
0x192a3:      jne     0x192e0
0x192a5:      mov     r9, qword ptr [rip + 0x115b4]
0x192ac:      xor     r8d, r8d
0x192af:      mov     rcx, r12
0x192b2:      xor     eax, eax
0x192b4:      lea     rdi, [rip + 0x7363]
0x192bb:      lea     rdx, [rip + 0x911e]
0x192c2:      lea     rsi, [rip + 0x735a]
0x192c9:      call    0x4e30
0x192ce:      mov     edi, 0x7f
0x192d3:      call    0x48a0
0x192d8:      nop     dword ptr [rax + rax]
0x192e0:      mov     edi, 1
0x192e5:      lea     rbp, [rip + 0x90ed]
0x192ec:      call    0x4ab0
0x192f1:      mov     edi, 2
0x192f6:      call    0x4ab0
0x192fb:      mov     rdi, qword ptr [rip + 0x127ce]
0x19302:      mov     rsi, rbp
0x19305:      call    0x4da0
0x1930a:      mov     rdi, qword ptr [rip + 0x127b7]
0x19311:      mov     rsi, rbp
0x19314:      call    0x4da0
0x19319:      mov     edx, 4
0x1931e:      lea     rsi, [rip + 0x90b6]
0x19325:      mov     rdi, r12
0x19328:      call    0x4890
0x1932d:      test    eax, eax
0x1932f:      je      0x192a5
0x19335:      xor     edi, edi
0x19337:      call    0x4ab0
0x1933c:      jmp     0x192a5
0x192a5
0x19341:      nop     word ptr cs:[rax + rax]
0x1934c:      nop     dword ptr [rax]
0x19350:      endbr64
0x19354:      push    r12
0x19356:      push    rbp
0x19357:      push    rbx
0x19358:      sub     rsp, 0x10
0x1935c:      mov     rax, qword ptr fs:[0x28]
0x19365:      mov     qword ptr [rsp + 8], rax
0x1936a:      xor     eax, eax
0x1936c:      test    rdi, rdi

```

```

0x1936f:      je      0x193d0
0x19371:      mov     rbp, rdi
0x19374:      mov     r12d, esi
0x19377:      call    0x4ed0
0x1937c:      mov     ebx, eax
0x1937e:      test    eax, eax
0x19380:      js      0x193c0
0x19382:      je      0x193dc
0x19384:      lea     rbp, [rsp + 4]
0x19389:      nop     dword ptr [rax]
0x19390:      mov     rdi, rbp
0x19393:      call    0x4ec0
0x19398:      cmp     eax, ebx
0x1939a:      jne     0x19390
0x1939c:      mov     eax, dword ptr [rsp + 4]
0x193a0:      mov     rdx, qword ptr [rsp + 8]
0x193a5:      sub     rdx, qword ptr fs:[0x28]
0x193ae:      jne     0x193d7
0x193b0:      add     rsp, 0x10
0x193b4:      pop     rbx
0x193b5:      pop     rbp
0x193b6:      pop     r12
0x193b8:      ret
0x193b9:      nop     dword ptr [rax]
0x193c0:      mov     eax, 0xffffffff
0x193c5:      jmp     0x193a0
0x193a0
0x193c7:      nop     word ptr [rax + rax]
0x193d0:      mov     eax, 1
0x193d5:      jmp     0x193a0
0x193a0
0x193d7:      call    0x49d0
0x193dc:      mov     esi, r12d
0x193df:      mov     rdi, rbp
0x193e2:      xor     eax, eax
0x193e4:      call    0x19280
0x193e9:      nop     dword ptr [rax]
0x193f0:      endbr64
0x193f4:      push    r13
0x193f6:      mov     r13, rdi
0x193f9:      push    r12
0x193fb:      push    rbp
0x193fc:      push    rbx
0x193fd:      sub     rsp, 0x18
0x19401:      mov     rax, qword ptr fs:[0x28]
0x1940a:      mov     qword ptr [rsp + 8], rax
0x1940f:      xor     eax, eax
0x19411:      test    esi, esi

```

```

0x19413:      je      0x19488
0x19415:      mov     edi, 1
0x1941a:      mov     ebp, esi
0x1941c:      call    0x18b70
0x19421:      test    r13, r13
0x19424:      je      0x19520
0x1942a:      call    0x4ed0
0x1942f:      mov     r12d, eax
0x19432:      test    eax, eax
0x19434:      js      0x19550
0x1943a:      je      0x19560
0x19440:      lea     rbx, [rsp + 4]
0x19445:      nop     dword ptr [rax]
0x19448:      mov     rdi, rbx
0x1944b:      call    0x4ec0
0x19450:      cmp     r12d, eax
0x19453:      jne     0x19448
0x19455:      mov     r12d, dword ptr [rsp + 4]
0x1945a:      xor     edi, edi
0x1945c:      call    0x18b70
0x19461:      mov     rax, qword ptr [rsp + 8]
0x19466:      sub     rax, qword ptr fs:[0x28]
0x1946f:      jne     0x1955b
0x19475:      add     rsp, 0x18
0x19479:      mov     eax, r12d
0x1947c:      pop     rbx
0x1947d:      pop     rbp
0x1947e:      pop     r12
0x19480:      pop     r13
0x19482:      ret
0x19483:      nop     dword ptr [rax + rax]
0x19488:      xor     edi, edi
0x1948a:      call    0x9b00
0x1948f:      xor     eax, eax
0x19491:      call    0xa6c0
0x19496:      mov     rdi, qword ptr [rip + 0x1142b]
0x1949d:      call    0xbb40
0x194a2:      xor     edi, edi
0x194a4:      call    0x18b90
0x194a9:      test    r13, r13
0x194ac:      je      0x19530
0x194b2:      call    0x4ed0
0x194b7:      mov     ebp, eax
0x194b9:      test    eax, eax
0x194bb:      js      0x19540
0x194c1:      je      0x1956c
0x194c7:      lea     rbx, [rsp + 4]
0x194cc:      nop     dword ptr [rax]

```

```

0x194d0:    mov     rdi, rbx
0x194d3:    call   0x4ec0
0x194d8:    cmp     ebp, eax
0x194da:    jne     0x194d0
0x194dc:    mov     r12d, dword ptr [rsp + 4]
0x194e1:    mov     edi, 1
0x194e6:    call   0x18b90
0x194eb:    mov     edx, 2
0x194f0:    mov     edi, 1
0x194f5:    lea     rsi, [rip + 0x8ee7]
0x194fc:    call   0x18da0
0x19501:    mov     edi, 1
0x19506:    call   0x9b00
0x1950b:    xor     eax, eax
0x1950d:    call   0xa6c0
0x19512:    xor     edi, edi
0x19514:    call   0x7890
0x19519:    jmp     0x19461
0x19461:
0x1951e:    nop
0x19520:    mov     r12d, 1
0x19526:    jmp     0x1945a
0x1945a:
0x1952b:    nop     dword ptr [rax + rax]
0x19530:    mov     r12d, 1
0x19536:    jmp     0x194e1
0x194e1:
0x19538:    nop     dword ptr [rax + rax]
0x19540:    mov     r12d, 0xffffffff
0x19546:    jmp     0x194e1
0x194e1:
0x19548:    nop     dword ptr [rax + rax]
0x19550:    mov     r12d, 0xffffffff
0x19556:    jmp     0x1945a
0x1945a:
0x1955b:    call   0x49d0
0x19560:    mov     esi, ebp
0x19562:    mov     rdi, r13
0x19565:    xor     eax, eax
0x19567:    call   0x19280
0x1956c:    xor     esi, esi
0x1956e:    mov     rdi, r13
0x19571:    xor     eax, eax
0x19573:    call   0x19280
0x19578:    nop     dword ptr [rax + rax]
0x19580:    endbr64
0x19584:    mov     rdi, qword ptr [rip + 0x12545]
0x1958b:    test    rdi, rdi

```

```

0x1958e:    je      0x195b8
0x19590:    sub     rsp, 8
0x19594:    call    0x4870
0x19599:    mov     rdi, qword ptr [rip + 0x12528]
0x195a0:    test    rdi, rdi
0x195a3:    je      0x195b0
0x195a5:    add     rsp, 8
0x195a9:    jmp     0x4870
0x4870
0x195ae:    nop
0x195b0:    add     rsp, 8
0x195b4:    ret
0x195b5:    nop     dword ptr [rax]
0x195b8:    mov     rdi, qword ptr [rip + 0x12509]
0x195bf:    test    rdi, rdi
0x195c2:    je      0x195d0
0x195c4:    jmp     0x4870
0x4870
0x195c9:    nop     dword ptr [rax]
0x195d0:    ret
0x195d1:    nop     word ptr cs:[rax + rax]
0x195dc:    nop     dword ptr [rax]
0x195e0:    endbr64
0x195e4:    push    r12
0x195e6:    lea     rsi, [rip + 0x89dc]
0x195ed:    push    rbp
0x195ee:    mov     rbp, rdi
0x195f1:    sub     rsp, 8
0x195f5:    mov     rdi, qword ptr [rip + 0x124cc]
0x195fc:    call    0x4da0
0x19601:    test    rax, rax
0x19604:    je      0x19676
0x19606:    mov     edi, 0x801
0x1960b:    mov     r12, rax
0x1960e:    call    0x1d670
0x19613:    mov     rbp, rax
0x19616:    jmp     0x1964a
0x1964a
0x19618:    nop     dword ptr [rax + rax]
0x19620:    mov     rdi, rbp
0x19623:    call    0x49b0
0x19628:    cdqe
0x1962a:    lea     rax, [rbp + rax - 1]
0x1962f:    cmp     byte ptr [rax], 0xa
0x19632:    jne     0x19637
0x19634:    mov     byte ptr [rax], 0
0x19637:    xor     esi, esi
0x19639:    mov     edx, 0xa

```

```

0x1963e:    mov     rdi, rbp
0x19641:    call   0x8070
0x19646:    test   al, al
0x19648:    je     0x1965f
0x1964a:    mov     rdx, r12
0x1964d:    mov     esi, 0x801
0x19652:    mov     rdi, rbp
0x19655:    call   0x4b00
0x1965a:    test   rax, rax
0x1965d:    jne    0x19620
0x1965f:    mov     rdi, rbp
0x19662:    call   0x18d10
0x19667:    add     rsp, 8
0x1966b:    mov     rdi, r12
0x1966e:    pop     rbp
0x1966f:    pop     r12
0x19671:    jmp     0x4940
0x4940:
0x19676:    mov     rdi, rbp
0x19679:    call   0x49b0
0x1967e:    mov     rdi, qword ptr [rip + 0x12443]
0x19685:    mov     r12, rax
0x19688:    call   0x49b0
0x1968d:    lea     rdi, [r12 + rax + 0x20]
0x19692:    call   0x1d670
0x19697:    mov     r9, qword ptr [rip + 0x1242a]
0x1969e:    mov     r8, rbp
0x196a1:    or      rdx, 0xffffffffffffffff
0x196a5:    mov     r12, rax
0x196a8:    mov     rdi, rax
0x196ab:    mov     esi, 1
0x196b0:    xor     eax, eax
0x196b2:    lea     rcx, [rip + 0x8d2d]
0x196b9:    call   0x4f10
0x196be:    mov     rdi, r12
0x196c1:    mov     edx, 0x1e
0x196c6:    xor     esi, esi
0x196c8:    call   0x8070
0x196cd:    pop     rax
0x196ce:    mov     rdi, r12
0x196d1:    pop     rbp
0x196d2:    pop     r12
0x196d4:    jmp     0x18d10
0x18d10:
0x196d9:    nop     dword ptr [rax]
0x196e0:    endbr64
0x196e4:    push    r15
0x196e6:    push    r14

```



```

0x196e8:    push    r13
0x196ea:    push    r12
0x196ec:    push    rbp
0x196ed:    push    rbx
0x196ee:    sub     rsp, 8
0x196f2:    test    rdi, rdi
0x196f5:    je      0x19878
0x196fb:    mov     rbp, rdi
0x196fe:    call    0x49b0
0x19703:    lea     rdi, [rax + 1]
0x19707:    call    0x1d670
0x1970c:    mov     rsi, rbp
0x1970f:    mov     rdi, rax
0x19712:    call    0x48b0
0x19717:    cmp     byte ptr [rax], 0x7e
0x1971a:    mov     rbp, rax
0x1971d:    mov     r12, rax
0x19720:    jne     0x19800
0x19726:    movzx   eax, byte ptr [rax + 1]
0x1972a:    lea     r13, [rbp + 1]
0x1972e:    test    al, al
0x19730:    je      0x19818
0x19736:    cmp     al, 0x2f
0x19738:    je      0x19818
0x1973e:    mov     rdi, rbp
0x19741:    call    0x49b0
0x19746:    mov     rdi, rax
0x19749:    call    0x1d670
0x1974e:    movzx   ecx, byte ptr [rbp + 1]
0x19752:    mov     r14, rax
0x19755:    cmp     cl, 0x2f
0x19758:    je      0x19938
0x1975e:    test    cl, cl
0x19760:    je      0x19938
0x19766:    mov     edx, 2
0x1976b:    nop     dword ptr [rax + rax]
0x19770:    mov     byte ptr [r14 + rdx - 2], cl
0x19775:    movzx   ecx, byte ptr [rbp + rdx]
0x1977a:    lea     r13, [rbp + rdx]
0x1977f:    mov     rax, rdx
0x19782:    add     rdx, 1
0x19786:    test    cl, cl
0x19788:    je      0x1978f
0x1978a:    cmp     cl, 0x2f
0x1978d:    jne     0x19770
0x1978f:    lea     rax, [r14 + rax - 1]
0x19794:    mov     byte ptr [rax], 0
0x19797:    mov     rdi, r14

```

```

0x1979a:      call    0x4b60
0x1979f:      mov     rbx, rax
0x197a2:      test   rax, rax
0x197a5:      je      0x19880
0x197ab:      mov     rdi, qword ptr [rax + 0x20]
0x197af:      call    0x49b0
0x197b4:      mov     rdi, r13
0x197b7:      mov     r15, rax
0x197ba:      call    0x49b0
0x197bf:      lea     edi, [r15 + 1]
0x197c3:      movsxd  rdi, edi
0x197c6:      add     rdi, rax
0x197c9:      call    0x1d670
0x197ce:      mov     rsi, qword ptr [rbx + 0x20]
0x197d2:      mov     rdi, rax
0x197d5:      mov     r12, rax
0x197d8:      call    0x48b0
0x197dd:      movsxd  rdi, r15d
0x197e0:      mov     rsi, r13
0x197e3:      add     rdi, r12
0x197e6:      call    0x48b0
0x197eb:      mov     rdi, rbp
0x197ee:      call    0x18d10
0x197f3:      call    0x4c70
0x197f8:      mov     rdi, r14
0x197fb:      call    0x18d10
0x19800:      add     rsp, 8
0x19804:      mov     rax, r12
0x19807:      pop     rbx
0x19808:      pop     rbp
0x19809:      pop     r12
0x1980b:      pop     r13
0x1980d:      pop     r14
0x1980f:      pop     r15
0x19811:      ret
0x19812:      nop     word ptr [rax + rax]
0x19818:      lea     rdi, [rip + 0x6d41]
0x1981f:      call    0x47e0
0x19824:      mov     r14, rax
0x19827:      test   rax, rax
0x1982a:      je      0x198f8
0x19830:      mov     rdi, r14
0x19833:      call    0x49b0
0x19838:      mov     rdi, r13
0x1983b:      mov     rbx, rax
0x1983e:      call    0x49b0
0x19843:      movsxd  rbx, ebx
0x19846:      lea     rdi, [rbx + rax + 1]

```

```

0x1984b:      call    0x1d670
0x19850:      mov     rsi, r14
0x19853:      mov     r12, rax
0x19856:      mov     rdi, rax
0x19859:      call    0x48b0
0x1985e:      lea     rdi, [r12 + rbx]
0x19862:      mov     rsi, r13
0x19865:      call    0x48b0
0x1986a:      mov     rdi, rbp
0x1986d:      call    0x18d10
0x19872:      jmp     0x19800
0x19800
0x19874:      nop     dword ptr [rax]
0x19878:      xor     r12d, r12d
0x1987b:      jmp     0x19800
0x19800
0x1987d:      nop     dword ptr [rax]
0x19880:      mov     rdx, qword ptr [rip + 0x12251]
0x19887:      test    rdx, rdx
0x1988a:      je      0x197f3
0x19890:      mov     rdi, r14
0x19893:      xor     eax, eax
0x19895:      call    rdx
0x19897:      mov     r15, rax
0x1989a:      test    rax, rax
0x1989d:      je      0x197f3
0x198a3:      mov     rdi, rax
0x198a6:      call    0x49b0
0x198ab:      mov     rdi, r13
0x198ae:      mov     rbx, rax
0x198b1:      call    0x49b0
0x198b6:      lea     edi, [rbx + 1]
0x198b9:      movsxd  rdi, edi
0x198bc:      add     rdi, rax
0x198bf:      call    0x1d670
0x198c4:      mov     rsi, r15
0x198c7:      mov     rdi, rax
0x198ca:      mov     r12, rax
0x198cd:      call    0x48b0
0x198d2:      movsxd  rdi, ebx
0x198d5:      mov     rsi, r13
0x198d8:      add     rdi, r12
0x198db:      call    0x48b0
0x198e0:      mov     rdi, r15
0x198e3:      call    0x18d10
0x198e8:      mov     rdi, rbp
0x198eb:      call    0x18d10
0x198f0:      jmp     0x197f3

```

```

0x197f3
0x198f5:    nop     dword ptr [rax]
0x198f8:    call   0x49e0
0x198fd:    mov     edi, eax
0x198ff:    call   0x4970
0x19904:    test    rax, rax
0x19907:    je      0x19916
0x19909:    mov     r14, qword ptr [rax + 0x20]
0x1990d:    test    r14, r14
0x19910:    jne     0x19830
0x19916:    mov     rdi, r13
0x19919:    call   0x49b0
0x1991e:    lea     rdi, [rax + 1]
0x19922:    call   0x1d670
0x19927:    mov     r12, rax
0x1992a:    mov     rdi, rax
0x1992d:    jmp     0x19862
0x19862
0x19932:    nop     word ptr [rax + rax]
0x19938:    mov     rax, r14
0x1993b:    jmp     0x19794
0x19794
0x19940:    endbr64
0x19944:    push    r15
0x19946:    push    r14
0x19948:    mov     r14, rdi
0x1994b:    push    r13
0x1994d:    push    r12
0x1994f:    push    rbp
0x19950:    push    rbx
0x19951:    xor     ebx, ebx
0x19953:    sub     rsp, 0x48
0x19957:    mov     dword ptr [rsp + 0x1c], 0
0x1995f:    mov     qword ptr [rsp + 0x10], 0
0x19968:    nop     dword ptr [rax + rax]
0x19970:    mov     r15, qword ptr [rip + 0x10e31]
0x19977:    mov     rdi, r14
0x1997a:    mov     qword ptr [rsp + 8], r15
0x1997f:    call   0x49b0
0x19984:    mov     qword ptr [rsp + 0x20], rax
0x19989:    mov     rcx, rax
0x1998c:    movzx   eax, byte ptr [r14]
0x19990:    test    al, al
0x19992:    je      0x19bd0
0x19998:    cmp     al, 0x7e
0x1999a:    je      0x19bd0
0x199a0:    mov     dword ptr [rsp + 0x18], ecx
0x199a4:    mov     rax, rcx

```

```

0x199a7:    test    r15, r15
0x199aa:    je      0x19c50
0x199b0:    mov     edx, dword ptr [rsp + 0x20]
0x199b4:    test    edx, edx
0x199b6:    jle     0x19c4a
0x199bc:    mov     rax, qword ptr [rsp + 8]
0x199c1:    mov     dword ptr [rsp + 0x3c], ebx
0x199c5:    xor     r12d, r12d
0x199c8:    mov     rax, qword ptr [rax]
0x199cb:    mov     qword ptr [rsp + 0x28], rax
0x199d0:    mov     rax, qword ptr [rsp + 0x20]
0x199d5:    sub     eax, 1
0x199d8:    mov     qword ptr [rsp + 0x30], rax
0x199dd:    mov     rax, qword ptr [rsp + 8]
0x199e2:    mov     dword ptr [rsp + 0x38], r12d
0x199e7:    lea     rbp, [r14 + r12]
0x199eb:    lea     rbx, [rax + 8]
0x199ef:    mov     rax, qword ptr [rsp + 0x28]
0x199f4:    mov     r13, rax
0x199f7:    test    rax, rax
0x199fa:    jne     0x19a18
0x199fc:    jmp     0x19bb0
0x19bb0:
0x19a01:    nop     dword ptr [rax]
0x19a08:    mov     r13, qword ptr [rbx]
0x19a0b:    add     rbx, 8
0x19a0f:    test    r13, r13
0x19a12:    je      0x19bb0
0x19a18:    mov     rdi, r13
0x19a1b:    call    0x49b0
0x19a20:    mov     rsi, r13
0x19a23:    mov     rdi, rbp
0x19a26:    mov     rdx, rax
0x19a29:    mov     r15, rax
0x19a2c:    call    0x4890
0x19a31:    test    eax, eax
0x19a33:    jne     0x19a08
0x19a35:    mov     eax, dword ptr [rsp + 0x38]
0x19a39:    mov     ebx, dword ptr [rsp + 0x3c]
0x19a3d:    lea     eax, [rax + r15 - 1]
0x19a42:    lea     ecx, [rbx + rax]
0x19a45:    mov     dword ptr [rsp + 0x18], eax
0x19a49:    movsxd  r13, eax
0x19a4c:    mov     dword ptr [rsp + 8], ecx
0x19a50:    lea     r15, [r14 + r13]
0x19a54:    mov     ecx, dword ptr [rsp + 8]
0x19a58:    cmp     dword ptr [rsp + 0x1c], ecx
0x19a5c:    jle     0x19bf0

```

```

0x19a62:    movsxd   rbx, ebx
0x19a65:    add      rbx, qword ptr [rsp + 0x10]
0x19a6a:    mov      rdx, r13
0x19a6d:    mov      rsi, r14
0x19a70:    mov      rdi, rbx
0x19a73:    mov      qword ptr [rsp + 0x28], rbx
0x19a78:    call     0x4880
0x19a7d:    mov      rdi, r15
0x19a80:    mov      r13, qword ptr [rip + 0x10d19]
0x19a87:    call     0x49b0
0x19a8c:    mov      dword ptr [rsp + 0x20], eax
0x19a90:    test     eax, eax
0x19a92:    jle      0x19c64
0x19a98:    mov      rbx, r15
0x19a9b:    xor      ebp, ebp
0x19a9d:    movzx    eax, byte ptr [rbx]
0x19aa0:    cmp      al, 0x2f
0x19aa2:    je       0x19b80
0x19aa8:    test     al, al
0x19aaa:    je       0x19b80
0x19ab0:    test     r13, r13
0x19ab3:    je       0x19aea
0x19ab5:    mov      r14, r13
0x19ab8:    jmp      0x19ae2
0x19ae2:
0x19aba:    nop      word ptr [rax + rax]
0x19ac0:    mov      rdi, r12
0x19ac3:    add      r14, 8
0x19ac7:    call     0x49b0
0x19acc:    mov      rsi, r12
0x19acf:    mov      rdi, rbx
0x19ad2:    mov      rdx, rax
0x19ad5:    call     0x4890
0x19ada:    test     eax, eax
0x19adc:    je       0x19b80
0x19ae2:    mov      r12, qword ptr [r14]
0x19ae5:    test     r12, r12
0x19ae8:    jne      0x19ac0
0x19aea:    add      ebp, 1
0x19aed:    add      rbx, 1
0x19af1:    cmp      dword ptr [rsp + 0x20], ebp
0x19af5:    jne      0x19a9d
0x19af7:    nop      word ptr [rax + rax]
0x19b00:    lea      edi, [rbp + 1]
0x19b03:    movsxd   rbp, ebp
0x19b06:    movsxd   rdi, edi
0x19b09:    lea      r14, [r15 + rbp]
0x19b0d:    call     0x1d670

```

```

0x19b12:    mov     rdx, rbp
0x19b15:    mov     rsi, r15
0x19b18:    mov     r13, rax
0x19b1b:    mov     rdi, rax
0x19b1e:    call    0x4880
0x19b23:    mov     byte ptr [r13 + rbp], 0
0x19b29:    mov     rdi, r13
0x19b2c:    xor     eax, eax
0x19b2e:    call    0x196e0
0x19b33:    mov     rdi, r13
0x19b36:    mov     rbp, rax
0x19b39:    call    0x18d10
0x19b3e:    mov     rdi, rbp
0x19b41:    call    0x49b0
0x19b46:    mov     ecx, dword ptr [rsp + 8]
0x19b4a:    lea     ebx, [rcx + rax]
0x19b4d:    cmp     ebx, dword ptr [rsp + 0x1c]
0x19b51:    jge     0x19c20
0x19b57:    movsxd  rdi, dword ptr [rsp + 8]
0x19b5c:    mov     rsi, rbp
0x19b5f:    add     rdi, qword ptr [rsp + 0x10]
0x19b64:    call    0x48b0
0x19b69:    mov     rdi, rbp
0x19b6c:    call    0x18d10
0x19b71:    jmp     0x19970
0x19970:
0x19b76:    nop     word ptr cs:[rax + rax]
0x19b80:    or      dword ptr [rsp + 0x18], ebp
0x19b84:    mov     eax, dword ptr [rsp + 0x18]
0x19b88:    test    eax, eax
0x19b8a:    jne     0x19b00
0x19b90:    mov     rax, qword ptr [rsp + 0x28]
0x19b95:    mov     byte ptr [rax], 0
0x19b98:    mov     rax, qword ptr [rsp + 0x10]
0x19b9d:    add     rsp, 0x48
0x19ba1:    pop     rbx
0x19ba2:    pop     rbp
0x19ba3:    pop     r12
0x19ba5:    pop     r13
0x19ba7:    pop     r14
0x19ba9:    pop     r15
0x19bab:    ret
0x19bac:    nop     dword ptr [rax]
0x19bb0:    lea     rax, [r12 + 1]
0x19bb5:    cmp     qword ptr [rsp + 0x30], r12
0x19bba:    je      0x19c46
0x19bc0:    mov     r12, rax
0x19bc3:    jmp     0x199dd

```

```

0x199dd
0x19bc8:    nop        dword ptr [rax + rax]
0x19bd0:    mov        dword ptr [rsp + 8], ebx
0x19bd4:    mov        r15, r14
0x19bd7:    mov        ecx, dword ptr [rsp + 8]
0x19bdb:    xor        r13d, r13d
0x19bde:    mov        dword ptr [rsp + 0x18], 0
0x19be6:    cmp        dword ptr [rsp + 0x1c], ecx
0x19bea:    jg         0x19a62
0x19bf0:    mov        eax, dword ptr [rsp + 0x1c]
0x19bf4:    mov        ecx, dword ptr [rsp + 0x18]
0x19bf8:    mov        rdi, qword ptr [rsp + 0x10]
0x19bfd:    lea        eax, [rax + rcx + 0x14]
0x19c01:    lea        esi, [rax + 1]
0x19c04:    mov        dword ptr [rsp + 0x1c], eax
0x19c08:    movsxd    rsi, esi
0x19c0b:    call       0x1d690
0x19c10:    mov        qword ptr [rsp + 0x10], rax
0x19c15:    jmp        0x19a62
0x19a62
0x19c1a:    nop        word ptr [rax + rax]
0x19c20:    mov        ecx, dword ptr [rsp + 0x1c]
0x19c24:    mov        rdi, qword ptr [rsp + 0x10]
0x19c29:    lea        eax, [rcx + rax + 0x14]
0x19c2d:    lea        esi, [rax + 1]
0x19c30:    mov        dword ptr [rsp + 0x1c], eax
0x19c34:    movsxd    rsi, esi
0x19c37:    call       0x1d690
0x19c3c:    mov        qword ptr [rsp + 0x10], rax
0x19c41:    jmp        0x19b57
0x19b57
0x19c46:    mov        ebx, dword ptr [rsp + 0x3c]
0x19c4a:    mov        rax, qword ptr [rsp + 0x20]
0x19c4f:    nop
0x19c50:    mov        ecx, eax
0x19c52:    movsxd    r13, eax
0x19c55:    add        ecx, ebx
0x19c57:    lea        r15, [r14 + r13]
0x19c5b:    mov        dword ptr [rsp + 8], ecx
0x19c5f:    jmp        0x19a54
0x19a54
0x19c64:    xor        ebp, ebp
0x19c66:    jmp        0x19b84
0x19b84
0x19c6b:    nop        dword ptr [rax + rax]
0x19c70:    push       r15
0x19c72:    mov        r15, rsi
0x19c75:    push       r14

```



```

0x19c77:      mov     r14d, edx
0x19c7a:      push    r13
0x19c7c:      push    r12
0x19c7e:      mov     r12, rdi
0x19c81:      push    rbp
0x19c82:      push    rbx
0x19c83:      sub     rsp, 0x18
0x19c87:      movsxd   rsi, dword ptr [rsi]
0x19c8a:      lea     r13, [rdi + rsi]
0x19c8e:      mov     edi, ecx
0x19c90:      mov     ebx, esi
0x19c92:      mov     rbp, rsi
0x19c95:      movzx   eax, byte ptr [r13]
0x19c9a:      and     edi, 1
0x19c9d:      mov     rdx, rsi
0x19ca0:      test    al, al
0x19ca2:      jne     0x19cdc
0x19ca4:      jmp     0x19db9
0x19db9:
0x19ca9:      nop     dword ptr [rax]
0x19cb0:      cmp     al, 0x5c
0x19cb2:      jne     0x19cc5
0x19cb4:      movsx   eax, byte ptr [r12 + rdx + 1]
0x19cba:      cmp     eax, r14d
0x19cbd:      sete    al
0x19cc0:      movzx   eax, al
0x19cc3:      add     ebx, eax
0x19cc5:      add     ebx, 1
0x19cc8:      movsxd   rdx, ebx
0x19ccb:      lea     r13, [r12 + rdx]
0x19ccf:      movzx   eax, byte ptr [r13]
0x19cd4:      test    al, al
0x19cd6:      je      0x19da0
0x19cdc:      movsx   ecx, al
0x19cdf:      cmp     ecx, r14d
0x19ce2:      jne     0x19cb0
0x19ce4:      cmp     ebp, ebx
0x19ce6:      jl      0x19d10
0x19ce8:      test    dil, dil
0x19ceb:      jne     0x19d10
0x19ced:      movzx   eax, byte ptr [r13]
0x19cf2:      xor     r9d, r9d
0x19cf5:      cmp     al, 1
0x19cf7:      sbb     ebx, -1
0x19cfa:      mov     dword ptr [r15], ebx
0x19cfd:      add     rsp, 0x18
0x19d01:      mov     rax, r9
0x19d04:      pop     rbx

```

```

0x19d05:      pop     rbp
0x19d06:      pop     r12
0x19d08:      pop     r13
0x19d0a:      pop     r14
0x19d0c:      pop     r15
0x19d0e:      ret
0x19d0f:      nop
0x19d10:      mov     edi, ebx
0x19d12:      mov     qword ptr [rsp], r8
0x19d16:      sub     edi, ebp
0x19d18:      mov     qword ptr [rsp + 8], rsi
0x19d1d:      add     edi, 1
0x19d20:      movsxd  rdi, edi
0x19d23:      call    0x1d670
0x19d28:      cmp     ebp, ebx
0x19d2a:      mov     r8, qword ptr [rsp]
0x19d2e:      mov     r9, rax
0x19d31:      jge     0x19dd6
0x19d37:      mov     rsi, qword ptr [rsp + 8]
0x19d3c:      mov     eax, 1
0x19d41:      jmp     0x19d5d
0x19d5d
0x19d43:      nop     dword ptr [rax + rax]
0x19d48:      add     ebp, 1
0x19d4b:      mov     byte ptr [r9 + rax - 1], dl
0x19d50:      mov     edx, eax
0x19d52:      add     rax, 1
0x19d56:      cmp     ebp, ebx
0x19d58:      jge     0x19d80
0x19d5a:      movsxd  rsi, ebp
0x19d5d:      movzx   edx, byte ptr [r12 + rsi]
0x19d62:      cmp     dl, 0x5c
0x19d65:      jne     0x19d48
0x19d67:      movsx   esi, byte ptr [r12 + rsi + 1]
0x19d6d:      cmp     esi, r14d
0x19d70:      jne     0x19d48
0x19d72:      add     ebp, 1
0x19d75:      mov     edx, esi
0x19d77:      jmp     0x19d48
0x19d48
0x19d79:      nop     dword ptr [rax]
0x19d80:      movsxd  rax, edx
0x19d83:      add     rax, r9
0x19d86:      mov     byte ptr [rax], 0
0x19d89:      mov     dword ptr [r8], edx
0x19d8c:      movzx   eax, byte ptr [r13]
0x19d91:      jmp     0x19cf5
0x19cf5

```

```

0x19d96:      nop      word ptr cs:[rax + rax]
0x19da0:      cmp      ebp, ebx
0x19da2:      jl       0x19d10
0x19da8:      test     dil, dil
0x19dab:      jne      0x19d10
0x19db1:      xor      r9d, r9d
0x19db4:      jmp      0x19cfa
0x19cfa
0x19db9:      test     ecx, ecx
0x19dbb:      je       0x19db1
0x19dbd:      mov      edi, 1
0x19dc2:      mov      qword ptr [rsp], r8
0x19dc6:      call     0x1d670
0x19dcb:      mov      r8, qword ptr [rsp]
0x19dcf:      xor      edx, edx
0x19dd1:      mov      r9, rax
0x19dd4:      jmp      0x19d86
0x19d86
0x19dd6:      xor      edx, edx
0x19dd8:      jmp      0x19d86
0x19d86
0x19dda:      nop      word ptr [rax + rax]
0x19de0:      push     r15
0x19de2:      push     r14
0x19de4:      push     r13
0x19de6:      push     r12
0x19de8:      push     rbp
0x19de9:      push     rbx
0x19dea:      sub      rsp, 0x28
0x19dee:      mov      eax, dword ptr [rip + 0x11cf0]
0x19df4:      mov      ebx, dword ptr [rip + 0x11d02]
0x19dfa:      mov      dword ptr [rsp + 8], esi
0x19dfe:      mov      dword ptr [rsp + 0x18], edx
0x19e02:      mov      dword ptr [rsp + 0x1c], eax
0x19e06:      test     eax, eax
0x19e08:      je       0x19f9d
0x19e0e:      mov      rbp, rdi
0x19e11:      cmp      eax, ebx
0x19e13:      je       0x19fb0
0x19e19:      call     0x49b0
0x19e1e:      mov      ecx, dword ptr [rsp + 8]
0x19e22:      mov      dword ptr [rsp + 0xc], eax
0x19e26:      test     ecx, ecx
0x19e28:      js       0x19fbb
0x19e2e:      mov      rax, qword ptr [rip + 0x11ccb]
0x19e35:      movsxd   r14, dword ptr [rsp + 0xc]
0x19e3a:      mov      qword ptr [rsp + 0x10], rax
0x19e3f:      mov      edx, dword ptr [rsp + 8]

```

```

0x19e43:    test    edx, edx
0x19e45:    js      0x19f16
0x19e4b:    cmp     dword ptr [rsp + 0x1c], ebx
0x19e4f:    je      0x19f9d
0x19e55:    mov     rcx, qword ptr [rsp + 0x10]
0x19e5a:    movsxd  rax, ebx
0x19e5d:    mov     rax, qword ptr [rcx + rax*8]
0x19e61:    mov     r13, qword ptr [rax]
0x19e64:    mov     rdi, r13
0x19e67:    call    0x49b0
0x19e6c:    mov     r10, rax
0x19e6f:    cmp     dword ptr [rsp + 0xc], eax
0x19e73:    jg      0x19f07
0x19e79:    cmp     dword ptr [rsp + 0x18], 1
0x19e7e:    movzx   r12d, byte ptr [rbp]
0x19e83:    je      0x19ee0
0x19e85:    mov     r15d, dword ptr [rsp + 0x18]
0x19e8a:    sub     r10d, dword ptr [rsp + 0xc]
0x19e8f:    jmp     0x19ea5
0x19ea5:
0x19e91:    nop     dword ptr [rax]
0x19e98:    add     r15d, 1
0x19e9c:    add     r13, 1
0x19ea0:    cmp     r15d, r10d
0x19ea3:    jg      0x19f07
0x19ea5:    cmp     byte ptr [r13], r12b
0x19ea9:    jne     0x19e98
0x19eab:    mov     rdx, r14
0x19eae:    mov     rsi, r13
0x19eb1:    mov     rdi, rbp
0x19eb4:    mov     dword ptr [rsp + 4], r10d
0x19eb9:    call    0x4890
0x19ebe:    mov     r10d, dword ptr [rsp + 4]
0x19ec3:    test    eax, eax
0x19ec5:    jne     0x19e98
0x19ec7:    mov     dword ptr [rip + 0x11c2f], ebx
0x19ecd:    add     rsp, 0x28
0x19ed1:    mov     eax, r15d
0x19ed4:    pop     rbx
0x19ed5:    pop     rbp
0x19ed6:    pop     r12
0x19ed8:    pop     r13
0x19eda:    pop     r14
0x19edc:    pop     r15
0x19ede:    ret
0x19edf:    nop
0x19ee0:    cmp     byte ptr [r13], r12b
0x19ee4:    jne     0x19efb

```

```

0x19ee6:    mov     rdx, r14
0x19ee9:    mov     rsi, r13
0x19eec:    mov     rdi, rbp
0x19eef:    call    0x4890
0x19ef4:    mov     r15d, eax
0x19ef7:    test    eax, eax
0x19ef9:    je      0x19ec7
0x19efb:    mov     eax, dword ptr [rsp + 8]
0x19eff:    test    eax, eax
0x19f01:    js      0x19f90
0x19f07:    mov     edx, dword ptr [rsp + 8]
0x19f0b:    add     ebx, 1
0x19f0e:    test    edx, edx
0x19f10:    jns     0x19e4b
0x19f16:    mov     rcx, qword ptr [rsp + 0x10]
0x19f1b:    movsxd  rax, ebx
0x19f1e:    mov     rax, qword ptr [rcx + rax*8]
0x19f22:    mov     r13, qword ptr [rax]
0x19f25:    mov     rdi, r13
0x19f28:    call    0x49b0
0x19f2d:    mov     r8, rax
0x19f30:    cmp     dword ptr [rsp + 0xc], eax
0x19f34:    jg      0x19f90
0x19f36:    cmp     dword ptr [rsp + 0x18], 1
0x19f3b:    movzx   r12d, byte ptr [rbp]
0x19f40:    je      0x19ee0
0x19f42:    mov     r15d, r8d
0x19f45:    sub     r15d, dword ptr [rsp + 0xc]
0x19f4a:    movsxd  rcx, r15d
0x19f4d:    add     r13, rcx
0x19f50:    jmp     0x19f66
0x19f66:
0x19f52:    nop     word ptr [rax + rax]
0x19f58:    sub     r15d, 1
0x19f5c:    sub     r13, 1
0x19f60:    cmp     r15d, -1
0x19f64:    je      0x19f90
0x19f66:    cmp     byte ptr [r13], r12b
0x19f6a:    jne     0x19f58
0x19f6c:    mov     rdx, r14
0x19f6f:    mov     rsi, r13
0x19f72:    mov     rdi, rbp
0x19f75:    call    0x4890
0x19f7a:    test    eax, eax
0x19f7c:    je      0x19ec7
0x19f82:    sub     r15d, 1
0x19f86:    sub     r13, 1
0x19f8a:    cmp     r15d, -1

```

```

0x19f8e:    jne     0x19f66
0x19f90:    sub     ebx, 1
0x19f93:    test    dword ptr [rsp + 8], ebx
0x19f97:    jns     0x19e3f
0x19f9d:    mov     r15d, 0xffffffff
0x19fa3:    jmp     0x19ecd
0x19ecd:
0x19fa8:    nop     dword ptr [rax + rax]
0x19fb0:    test    esi, esi
0x19fb2:    jns     0x19f9d
0x19fb4:    mov     eax, dword ptr [rsp + 0x1c]
0x19fb8:    lea     ebx, [rax - 1]
0x19fbb:    mov     rdi, rbp
0x19fbe:    call    0x49b0
0x19fc3:    mov     dword ptr [rsp + 0xc], eax
0x19fc7:    test    dword ptr [rsp + 8], ebx
0x19fcb:    jns     0x19e2e
0x19fd1:    jmp     0x19f9d
0x19f9d:
0x19fd3:    nop     word ptr cs:[rax + rax]
0x19fde:    nop
0x19fe0:    push    r14
0x19fe2:    push    r13
0x19fe4:    movsxd  r13, esi
0x19fe7:    sub     edx, r13d
0x19fea:    push    r12
0x19fec:    push    rbp
0x19fed:    mov     rbp, rdi
0x19ff0:    push    rbx
0x19ff1:    movsxd  rbx, edx
0x19ff4:    cmp     ecx, 2
0x19ff7:    je      0x1a070
0x19ff9:    mov     eax, 0x1d
0x19ffe:    lea     r14, [rip + 0x843a]
0x1a005:    cmp     ecx, 3
0x1a008:    je      0x1a028
0x1a00a:    xor     eax, eax
0x1a00c:    lea     r14, [rip + 0x83f5]
0x1a013:    lea     rdx, [rip + 0x83fe]
0x1a01a:    cmp     ecx, 1
0x1a01d:    sete    al
0x1a020:    cmove   r14, rdx
0x1a024:    lea     eax, [rax + rax*2 + 0xf]
0x1a028:    lea     edi, [rax + rbx + 3]
0x1a02c:    movsxd  rdi, edi
0x1a02f:    call    0x1d670
0x1a034:    lea     rsi, [rbp + r13]
0x1a039:    mov     rdx, rbx

```

```

0x1a03c:    mov     r12, rax
0x1a03f:    mov     rdi, rax
0x1a042:    call    0x4880
0x1a047:    mov     byte ptr [r12 + rbx], 0x3a
0x1a04c:    lea     rdi, [r12 + rbx + 2]
0x1a051:    mov     rsi, r14
0x1a054:    mov     byte ptr [r12 + rbx + 1], 0x20
0x1a05a:    call    0x48b0
0x1a05f:    pop     rbx
0x1a060:    mov     rax, r12
0x1a063:    pop     rbp
0x1a064:    pop     r12
0x1a066:    pop     r13
0x1a068:    pop     r14
0x1a06a:    ret
0x1a06b:    nop     dword ptr [rax + rax]
0x1a070:    mov     eax, 0x13
0x1a075:    lea     r14, [rip + 0x83af]
0x1a07c:    jmp     0x1a028
0x1a028:
0x1a07e:    nop
0x1a080:    push    r12
0x1a082:    push    rbp
0x1a083:    push    rbx
0x1a084:    test    rdi, rdi
0x1a087:    je      0x1a0b0
0x1a089:    mov     rbp, rdi
0x1a08c:    call    0x49b0
0x1a091:    lea     rdi, [rax + 1]
0x1a095:    call    0x1d670
0x1a09a:    pop     rbx
0x1a09b:    mov     rsi, rbp
0x1a09e:    pop     rbp
0x1a09f:    mov     rdi, rax
0x1a0a2:    pop     r12
0x1a0a4:    jmp     0x48b0
0x48b0:
0x1a0a9:    nop     dword ptr [rax]
0x1a0b0:    lea     rdi, [rip + 0x64a9]
0x1a0b7:    call    0x47e0
0x1a0bc:    mov     r12, rax
0x1a0bf:    test    rax, rax
0x1a0c2:    je      0x1a110
0x1a0c4:    mov     rdi, rax
0x1a0c7:    call    0x49b0
0x1a0cc:    lea     edi, [rax + 0xa]
0x1a0cf:    movsxd  rbp, eax
0x1a0d2:    movsxd  rdi, edi

```

```

0x1a0d5:    lea     rbx, [rbp + 1]
0x1a0d9:    call   0x1d670
0x1a0de:    mov     rsi, r12
0x1a0e1:    mov     rdi, rax
0x1a0e4:    call   0x48b0
0x1a0e9:    movabs  rdx, 0x79726f747369682e
0x1a0f3:    mov     byte ptr [rax + rbp], 0x2f
0x1a0f7:    mov     r8, rax
0x1a0fa:    lea     rax, [rax + rbx]
0x1a0fe:    mov     qword ptr [rax], rdx
0x1a101:    mov     byte ptr [rax + 8], 0
0x1a105:    mov     rax, r8
0x1a108:    pop     rbx
0x1a109:    pop     rbp
0x1a10a:    pop     r12
0x1a10c:    ret
0x1a10d:    nop     dword ptr [rax]
0x1a110:    mov     ebx, 2
0x1a115:    mov     ebp, 1
0x1a11a:    mov     edi, 0xb
0x1a11f:    lea     r12, [rip + 0x75b8]
0x1a126:    jmp     0x1a0d9
0x1a0d9
0x1a128:    nop     dword ptr [rax + rax]
0x1a130:    push    r15
0x1a132:    xor     eax, eax
0x1a134:    push    r14
0x1a136:    mov     r14d, esi
0x1a139:    push    r13
0x1a13b:    push    r12
0x1a13d:    push    rbp
0x1a13e:    push    rbx
0x1a13f:    mov     ebx, edx
0x1a141:    sub     rsp, 0x18
0x1a145:    call   0x1a080
0x1a14a:    cmp     ebx, 1
0x1a14d:    mov     edx, 0x1b6
0x1a152:    sbb     esi, esi
0x1a154:    mov     r13, rax
0x1a157:    mov     rdi, rax
0x1a15a:    xor     eax, eax
0x1a15c:    and     esi, 0x1c0
0x1a162:    add     esi, 0x241
0x1a168:    call   0x4d80
0x1a16d:    cmp     eax, -1
0x1a170:    je      0x1a270
0x1a176:    mov     ebp, eax
0x1a178:    mov     eax, dword ptr [rip + 0x11966]

```



```

0x1a17e:      cmp     eax, r14d
0x1a181:      mov     edx, eax
0x1a183:      cmovle  r14d, eax
0x1a187:      sub     edx, r14d
0x1a18a:      cmp     eax, edx
0x1a18c:      jle     0x1a290
0x1a192:      sub     eax, 1
0x1a195:      mov     rcx, qword ptr [rip + 0x11964]
0x1a19c:      movsxd  rsi, edx
0x1a19f:      xor     r12d, r12d
0x1a1a2:      sub     eax, edx
0x1a1a4:      add     rax, rsi
0x1a1a7:      lea     rbx, [rcx + rsi*8]
0x1a1ab:      lea     r15, [rcx + rax*8 + 8]
0x1a1b0:      mov     rax, qword ptr [rbx]
0x1a1b3:      add     rbx, 8
0x1a1b7:      mov     rdi, qword ptr [rax]
0x1a1ba:      call    0x49b0
0x1a1bf:      lea     r12d, [r12 + rax + 1]
0x1a1c4:      cmp     r15, rbx
0x1a1c7:      jne     0x1a1b0
0x1a1c9:      movsxd  rax, r12d
0x1a1cc:      mov     qword ptr [rsp + 8], rax
0x1a1d1:      mov     rdi, qword ptr [rsp + 8]
0x1a1d6:      call    0x1d670
0x1a1db:      mov     r15, rax
0x1a1de:      mov     eax, dword ptr [rip + 0x11900]
0x1a1e4:      mov     ebx, eax
0x1a1e6:      sub     ebx, r14d
0x1a1e9:      cmp     eax, ebx
0x1a1eb:      jle     0x1a238
0x1a1ed:      mov     r12, qword ptr [rip + 0x1190c]
0x1a1f4:      movsxd  rbx, ebx
0x1a1f7:      xor     r14d, r14d
0x1a1fa:      nop     word ptr [rax + rax]
0x1a200:      mov     rax, qword ptr [r12 + rbx*8]
0x1a204:      movsxd  rdi, r14d
0x1a207:      add     rdi, r15
0x1a20a:      mov     rsi, qword ptr [rax]
0x1a20d:      call    0x48b0
0x1a212:      mov     rax, qword ptr [r12 + rbx*8]
0x1a216:      add     rbx, 1
0x1a21a:      mov     rdi, qword ptr [rax]
0x1a21d:      call    0x49b0
0x1a222:      add     eax, r14d
0x1a225:      lea     r14d, [rax + 1]
0x1a229:      cdqe
0x1a22b:      mov     byte ptr [r15 + rax], 0xa

```

```

0x1a230:    cmp     dword ptr [rip + 0x118ae], ebx
0x1a236:    jg      0x1a200
0x1a238:    mov     rdx, qword ptr [rsp + 8]
0x1a23d:    mov     rsi, r15
0x1a240:    mov     edi, ebp
0x1a242:    call    0x4790
0x1a247:    mov     rdi, r15
0x1a24a:    call    0x18d10
0x1a24f:    mov     edi, ebp
0x1a251:    call    0x4ab0
0x1a256:    mov     rdi, r13
0x1a259:    call    0x18d10
0x1a25e:    add     rsp, 0x18
0x1a262:    xor     eax, eax
0x1a264:    pop     rbx
0x1a265:    pop     rbp
0x1a266:    pop     r12
0x1a268:    pop     r13
0x1a26a:    pop     r14
0x1a26c:    pop     r15
0x1a26e:    ret
0x1a26f:    nop
0x1a270:    mov     rdi, r13
0x1a273:    call    0x18d10
0x1a278:    call    0x4860
0x1a27d:    mov     eax, dword ptr [rax]
0x1a27f:    add     rsp, 0x18
0x1a283:    pop     rbx
0x1a284:    pop     rbp
0x1a285:    pop     r12
0x1a287:    pop     r13
0x1a289:    pop     r14
0x1a28b:    pop     r15
0x1a28d:    ret
0x1a28e:    nop
0x1a290:    mov     qword ptr [rsp + 8], 0
0x1a299:    jmp     0x1a1d1
0x1a1d1:
0x1a29e:    nop
0x1a2a0:    endbr64
0x1a2a4:    sub     rsp, 8
0x1a2a8:    mov     edi, 0x18
0x1a2ad:    call    0x1d670
0x1a2b2:    mov     rdx, qword ptr [rip + 0x11847]
0x1a2b9:    mov     qword ptr [rax], rdx
0x1a2bc:    mov     edx, dword ptr [rip + 0x1183a]
0x1a2c2:    mov     dword ptr [rax + 8], edx
0x1a2c5:    mov     edx, dword ptr [rip + 0x11819]

```

```

0x1a2cb:    mov     dword ptr [rax + 0xc], edx
0x1a2ce:    mov     edx, dword ptr [rip + 0x11824]
0x1a2d4:    mov     dword ptr [rax + 0x10], edx
0x1a2d7:    add     rsp, 8
0x1a2db:    ret
0x1a2dc:    nop     dword ptr [rax]
0x1a2e0:    endbr64
0x1a2e4:    mov     rax, qword ptr [rdi]
0x1a2e7:    mov     qword ptr [rip + 0x11812], rax
0x1a2ee:    mov     eax, dword ptr [rdi + 8]
0x1a2f1:    mov     dword ptr [rip + 0x11805], eax
0x1a2f7:    mov     eax, dword ptr [rdi + 0xc]
0x1a2fa:    mov     dword ptr [rip + 0x117e4], eax
0x1a300:    mov     eax, dword ptr [rdi + 0x10]
0x1a303:    mov     dword ptr [rip + 0x117ef], eax
0x1a309:    ret
0x1a30a:    nop     word ptr [rax + rax]
0x1a310:    endbr64
0x1a314:    mov     eax, dword ptr [rip + 0x117ca]
0x1a31a:    mov     dword ptr [rip + 0x117dc], eax
0x1a320:    ret
0x1a321:    nop     word ptr cs:[rax + rax]
0x1a32c:    nop     dword ptr [rax]
0x1a330:    endbr64
0x1a334:    push    r12
0x1a336:    xor     r12d, r12d
0x1a339:    push    rbx
0x1a33a:    sub     rsp, 8
0x1a33e:    mov     rbx, qword ptr [rip + 0x117bb]
0x1a345:    test    rbx, rbx
0x1a348:    jne     0x1a35b
0x1a34a:    jmp     0x1a367
0x1a367:
0x1a34c:    nop     dword ptr [rax]
0x1a350:    mov     rdi, qword ptr [rax]
0x1a353:    call    0x49b0
0x1a358:    add     r12d, eax
0x1a35b:    mov     rax, qword ptr [rbx]
0x1a35e:    add     rbx, 8
0x1a362:    test    rax, rax
0x1a365:    jne     0x1a350
0x1a367:    add     rsp, 8
0x1a36b:    mov     eax, r12d
0x1a36e:    pop     rbx
0x1a36f:    pop     r12
0x1a371:    ret
0x1a372:    nop     word ptr cs:[rax + rax]
0x1a37d:    nop     dword ptr [rax]

```

```

0x1a380:      endbr64
0x1a384:      push    rbp
0x1a385:      mov     rbp, rdi
0x1a388:      push    rbx
0x1a389:      sub     rsp, 8
0x1a38d:      mov     eax, dword ptr [rip + 0x11759]
0x1a393:      test    eax, eax
0x1a395:      je      0x1a3aa
0x1a397:      movsxd  rdx, dword ptr [rip + 0x11746]
0x1a39e:      cmp     edx, dword ptr [rip + 0x11744]
0x1a3a4:      je      0x1a458
0x1a3aa:      mov     eax, dword ptr [rip + 0x11748]
0x1a3b0:      test    eax, eax
0x1a3b2:      je      0x1a428
0x1a3b4:      mov     edx, dword ptr [rip + 0x1172a]
0x1a3ba:      lea     ecx, [rax - 1]
0x1a3bd:      cmp     ecx, edx
0x1a3bf:      je      0x1a4b0
0x1a3c5:      add     edx, 1
0x1a3c8:      mov     dword ptr [rip + 0x11716], edx
0x1a3ce:      mov     edi, 0x10
0x1a3d3:      call    0x1d670
0x1a3d8:      mov     rdi, rbp
0x1a3db:      mov     rbx, rax
0x1a3de:      call    0x49b0
0x1a3e3:      lea     rdi, [rax + 1]
0x1a3e7:      call    0x1d670
0x1a3ec:      mov     rsi, rbp
0x1a3ef:      mov     rdi, rax
0x1a3f2:      call    0x48b0
0x1a3f7:      movsxd  rdx, dword ptr [rip + 0x116e6]
0x1a3fe:      mov     qword ptr [rbx + 8], 0
0x1a406:      mov     qword ptr [rbx], rax
0x1a409:      mov     rax, qword ptr [rip + 0x116f0]
0x1a410:      mov     qword ptr [rax + rdx*8], 0
0x1a418:      mov     qword ptr [rax + rdx*8 - 8], rbx
0x1a41d:      add     rsp, 8
0x1a421:      pop     rbx
0x1a422:      pop     rbp
0x1a423:      ret
0x1a424:      nop     dword ptr [rax]
0x1a428:      mov     dword ptr [rip + 0x116c6], 0x32
0x1a432:      mov     edi, 0x190
0x1a437:      call    0x1d670
0x1a43c:      mov     dword ptr [rip + 0x1169e], 1
0x1a446:      mov     qword ptr [rip + 0x116b3], rax
0x1a44d:      jmp     0x1a3ce
0x1a3ce:

```

```

0x1a452:      nop      word ptr [rax + rax]
0x1a458:      test     edx, edx
0x1a45a:      je       0x1a41d
0x1a45c:      mov      rax, qword ptr [rip + 0x1169d]
0x1a463:      mov      rax, qword ptr [rax]
0x1a466:      test     rax, rax
0x1a469:      je       0x1a489
0x1a46b:      mov      rdi, qword ptr [rax]
0x1a46e:      call     0x18d10
0x1a473:      mov      rax, qword ptr [rip + 0x11686]
0x1a47a:      mov      rdi, qword ptr [rax]
0x1a47d:      call     0x18d10
0x1a482:      movsxd   rdx, dword ptr [rip + 0x1165b]
0x1a489:      test     edx, edx
0x1a48b:      jle      0x1a4a1
0x1a48d:      mov      rdi, qword ptr [rip + 0x1166c]
0x1a494:      shl      rdx, 3
0x1a498:      lea      rsi, [rdi + 8]
0x1a49c:      call     0x4d40
0x1a4a1:      add      dword ptr [rip + 0x10340], 1
0x1a4a8:      jmp      0x1a3ce
0x1a3ce
0x1a4ad:      nop      dword ptr [rax]
0x1a4b0:      add      eax, 0x32
0x1a4b3:      mov      rdi, qword ptr [rip + 0x11646]
0x1a4ba:      mov      dword ptr [rip + 0x11638], eax
0x1a4c0:      cdqe
0x1a4c2:      lea      rsi, [rax*8]
0x1a4ca:      call     0x1d690
0x1a4cf:      mov      edx, dword ptr [rip + 0x1160f]
0x1a4d5:      mov      qword ptr [rip + 0x11624], rax
0x1a4dc:      jmp      0x1a3c5
0x1a3c5
0x1a4e1:      nop      word ptr cs:[rax + rax]
0x1a4ec:      nop      dword ptr [rax]
0x1a4f0:      endbr64
0x1a4f4:      push     r14
0x1a4f6:      xor      r14d, r14d
0x1a4f9:      push     r13
0x1a4fb:      mov      r13, rdx
0x1a4fe:      push     r12
0x1a500:      mov      r12, rsi
0x1a503:      push     rbp
0x1a504:      push     rbx
0x1a505:      movsxd   rbx, edi
0x1a508:      mov      edi, 0x10
0x1a50d:      call     0x1d670
0x1a512:      cmp      dword ptr [rip + 0x115cc], ebx

```

```

0x1a518:    jle     0x1a557
0x1a51a:    mov     rbp, rax
0x1a51d:    mov     rax, qword ptr [rip + 0x115dc]
0x1a524:    mov     rdi, r12
0x1a527:    mov     r14, qword ptr [rax + rbx*8]
0x1a52b:    call    0x49b0
0x1a530:    lea     rdi, [rax + 1]
0x1a534:    call    0x1d670
0x1a539:    mov     rsi, r12
0x1a53c:    mov     rdi, rax
0x1a53f:    call    0x48b0
0x1a544:    mov     qword ptr [rbp + 8], r13
0x1a548:    mov     qword ptr [rbp], rax
0x1a54c:    mov     rax, qword ptr [rip + 0x115ad]
0x1a553:    mov     qword ptr [rax + rbx*8], rbp
0x1a557:    pop     rbx
0x1a558:    mov     rax, r14
0x1a55b:    pop     rbp
0x1a55c:    pop     r12
0x1a55e:    pop     r13
0x1a560:    pop     r14
0x1a562:    ret
0x1a563:    nop     word ptr cs:[rax + rax]
0x1a56e:    nop
0x1a570:    endbr64
0x1a574:    mov     eax, dword ptr [rip + 0x11582]
0x1a57a:    ret
0x1a57b:    nop     dword ptr [rax + rax]
0x1a580:    endbr64
0x1a584:    xor     edx, edx
0x1a586:    xor     eax, eax
0x1a588:    jmp     0x19de0
0x19de0
0x1a58d:    nop     dword ptr [rax]
0x1a590:    endbr64
0x1a594:    mov     edx, 1
0x1a599:    xor     eax, eax
0x1a59b:    jmp     0x19de0
0x19de0
0x1a5a0:    endbr64
0x1a5a4:    push    r12
0x1a5a6:    push    rbx
0x1a5a7:    sub     rsp, 8
0x1a5ab:    mov     eax, dword ptr [rip + 0x11533]
0x1a5b1:    cmp     eax, edi
0x1a5b3:    jle     0x1a610
0x1a5b5:    test    eax, eax
0x1a5b7:    je      0x1a610

```

```

0x1a5b9:    lea     ebx, [rax - 1]
0x1a5bc:    movsxd  rdx, edi
0x1a5bf:    mov     rcx, qword ptr [rip + 0x1153a]
0x1a5c6:    mov     esi, ebx
0x1a5c8:    shl     rdx, 3
0x1a5cc:    sub     esi, edi
0x1a5ce:    cmp     eax, edi
0x1a5d0:    lea     r9, [rcx + rdx]
0x1a5d4:    mov     eax, 8
0x1a5d9:    lea     r8, [rsi*8 + 8]
0x1a5e1:    mov     r12, qword ptr [r9]
0x1a5e4:    mov     rdi, r9
0x1a5e7:    lea     rsi, [rcx + rdx + 8]
0x1a5ec:    cmovle  r8, rax
0x1a5f0:    mov     rdx, r8
0x1a5f3:    call    0x4d40
0x1a5f8:    mov     dword ptr [rip + 0x114e6], ebx
0x1a5fe:    add     rsp, 8
0x1a602:    mov     rax, r12
0x1a605:    pop     rbx
0x1a606:    pop     r12
0x1a608:    ret
0x1a609:    nop     dword ptr [rax]
0x1a610:    xor     r12d, r12d
0x1a613:    add     rsp, 8
0x1a617:    mov     rax, r12
0x1a61a:    pop     rbx
0x1a61b:    pop     r12
0x1a61d:    ret
0x1a61e:    nop
0x1a620:    endbr64
0x1a624:    push    r12
0x1a626:    mov     eax, dword ptr [rip + 0x114b8]
0x1a62c:    push    rbp
0x1a62d:    xor     ebp, ebp
0x1a62f:    test    edi, edi
0x1a631:    cmovns  ebp, edi
0x1a634:    push    rbx
0x1a635:    cmp     eax, ebp
0x1a637:    jle     0x1a6ce
0x1a63d:    sub     eax, ebp
0x1a63f:    mov     r12d, edi
0x1a642:    test    eax, eax
0x1a644:    jle     0x1a6e8
0x1a64a:    xor     ebx, ebx
0x1a64c:    nop     dword ptr [rax]
0x1a650:    mov     rax, qword ptr [rip + 0x114a9]
0x1a657:    mov     rax, qword ptr [rax + rbx*8]

```

```

0x1a65b:    mov     rdi, qword ptr [rax]
0x1a65e:    call   0x18d10
0x1a663:    mov     rax, qword ptr [rip + 0x11496]
0x1a66a:    mov     rdi, qword ptr [rax + rbx*8]
0x1a66e:    call   0x18d10
0x1a673:    mov     eax, dword ptr [rip + 0x1146b]
0x1a679:    lea     edx, [rbx + 1]
0x1a67c:    add     rbx, 1
0x1a680:    sub     eax, ebp
0x1a682:    cmp     eax, ebx
0x1a684:    jg      0x1a650
0x1a686:    mov     dword ptr [rip + 0x1015c], edx
0x1a68c:    mov     rcx, qword ptr [rip + 0x1146d]
0x1a693:    test    r12d, r12d
0x1a696:    jle     0x1a6f0
0x1a698:    cdqe
0x1a69a:    lea     rsi, [rcx + rax*8]
0x1a69e:    xor     eax, eax
0x1a6a0:    mov     rdx, qword ptr [rsi + rax*8]
0x1a6a4:    mov     qword ptr [rcx + rax*8], rdx
0x1a6a8:    add     rax, 1
0x1a6ac:    cmp     ebp, eax
0x1a6ae:    jg      0x1a6a0
0x1a6b0:    test    ebp, ebp
0x1a6b2:    mov     eax, 1
0x1a6b7:    cmovg   eax, ebp
0x1a6ba:    movsxd  rdx, eax
0x1a6bd:    lea     rcx, [rcx + rdx*8]
0x1a6c1:    mov     qword ptr [rcx], 0
0x1a6c8:    mov     dword ptr [rip + 0x11416], eax
0x1a6ce:    pop     rbx
0x1a6cf:    mov     dword ptr [rip + 0x11413], ebp
0x1a6d5:    mov     dword ptr [rip + 0x1140d], 1
0x1a6df:    pop     rbp
0x1a6e0:    pop     r12
0x1a6e2:    ret
0x1a6e3:    nop     dword ptr [rax + rax]
0x1a6e8:    xor     edx, edx
0x1a6ea:    jmp     0x1a686
0x1a686:
0x1a6ec:    nop     dword ptr [rax]
0x1a6f0:    xor     eax, eax
0x1a6f2:    jmp     0x1a6c1
0x1a6c1:
0x1a6f4:    nop     word ptr cs:[rax + rax]
0x1a6ff:    nop
0x1a700:    endbr64
0x1a704:    mov     edx, dword ptr [rip + 0x113e2]

```



```

0x1a70a:    mov     eax, dword ptr [rip + 0x113d8]
0x1a710:    test    edx, edx
0x1a712:    je      0x1a720
0x1a714:    mov     dword ptr [rip + 0x113ce], 0
0x1a71e:    neg     eax
0x1a720:    ret
0x1a721:    nop     word ptr cs:[rax + rax]
0x1a72c:    nop     dword ptr [rax]
0x1a730:    endbr64
0x1a734:    push    r15
0x1a736:    push    r14
0x1a738:    mov     r14d, esi
0x1a73b:    push    r13
0x1a73d:    push    r12
0x1a73f:    mov     r12d, edx
0x1a742:    push    rbp
0x1a743:    push    rbx
0x1a744:    sub     rsp, 0xa8
0x1a74b:    mov     rax, qword ptr fs:[0x28]
0x1a754:    mov     qword ptr [rsp + 0x98], rax
0x1a75c:    xor     eax, eax
0x1a75e:    call    0x1a080
0x1a763:    xor     esi, esi
0x1a765:    mov     edx, 0x1b6
0x1a76a:    mov     rbp, rax
0x1a76d:    mov     rdi, rax
0x1a770:    xor     eax, eax
0x1a772:    call    0x4d80
0x1a777:    test    eax, eax
0x1a779:    js      0x1a80f
0x1a77f:    mov     rsi, rsp
0x1a782:    mov     edi, eax
0x1a784:    mov     r15d, eax
0x1a787:    call    0x18ef0
0x1a78c:    cmp     eax, -1
0x1a78f:    je      0x1a807
0x1a791:    mov     eax, dword ptr [rsp + 0x30]
0x1a795:    lea     edi, [rax + 1]
0x1a798:    movsxd  rdi, edi
0x1a79b:    call    0x1d670
0x1a7a0:    mov     rdx, qword ptr [rsp + 0x30]
0x1a7a5:    mov     edi, r15d
0x1a7a8:    mov     rsi, rax
0x1a7ab:    mov     r13, rax
0x1a7ae:    call    0x47a0
0x1a7b3:    cmp     qword ptr [rsp + 0x30], rax
0x1a7b8:    je      0x1a819
0x1a7ba:    mov     edi, r15d

```

```

0x1a7bd:      call    0x4ab0
0x1a7c2:      mov     rdi, rbp
0x1a7c5:      call    0x18d10
0x1a7ca:      test    r13, r13
0x1a7cd:      je      0x1a7d7
0x1a7cf:      mov     rdi, r13
0x1a7d2:      call    0x18d10
0x1a7d7:      call    0x4860
0x1a7dc:      mov     eax, dword ptr [rax]
0x1a7de:      mov     rdx, qword ptr [rsp + 0x98]
0x1a7e6:      sub     rdx, qword ptr fs:[0x28]
0x1a7ef:      jne     0x1a8f0
0x1a7f5:      add     rsp, 0xa8
0x1a7fc:      pop     rbx
0x1a7fd:      pop     rbp
0x1a7fe:      pop     r12
0x1a800:      pop     r13
0x1a802:      pop     r14
0x1a804:      pop     r15
0x1a806:      ret
0x1a807:      mov     edi, r15d
0x1a80a:      call    0x4ab0
0x1a80f:      mov     rdi, rbp
0x1a812:      call    0x18d10
0x1a817:      jmp     0x1a7d7
0x1a7d7:
0x1a819:      mov     edi, r15d
0x1a81c:      call    0x4ab0
0x1a821:      mov     rcx, qword ptr [rsp + 0x30]
0x1a826:      test    r12d, r12d
0x1a829:      jns     0x1a82e
0x1a82b:      mov     r12d, ecx
0x1a82e:      xor     ebx, ebx
0x1a830:      xor     esi, esi
0x1a832:      movsxd  rdx, esi
0x1a835:      cmp     ebx, r14d
0x1a838:      jge     0x1a885
0x1a83a:      cmp     rdx, rcx
0x1a83d:      jge     0x1a885
0x1a83f:      lea     eax, [rsi + 1]
0x1a842:      cdqe
0x1a844:      cmp     rdx, rcx
0x1a847:      jl      0x1a85c
0x1a849:      jmp     0x1a87b
0x1a87b:
0x1a84b:      nop     dword ptr [rax + rax]
0x1a850:      lea     rdx, [rax + 1]
0x1a854:      cmp     rax, rcx

```

```

0x1a857:      je      0x1a878
0x1a859:      mov     rax, rdx
0x1a85c:      cmp     byte ptr [r13 + rax - 1], 0xa
0x1a862:      jne     0x1a850
0x1a864:      add     ebx, 1
0x1a867:      movsxd  rsi, eax
0x1a86a:      cmp     r14d, ebx
0x1a86d:      je      0x1a832
0x1a86f:      lea     rdx, [rax + 1]
0x1a873:      cmp     rax, rcx
0x1a876:      jne     0x1a859
0x1a878:      movsxd  rdx, esi
0x1a87b:      cmp     rcx, rdx
0x1a87e:      jle     0x1a885
0x1a880:      cmp     r14d, ebx
0x1a883:      jg      0x1a83f
0x1a885:      lea     r14, [r13 + rdx]
0x1a88a:      lea     r15d, [rsi + 1]
0x1a88e:      cmp     rcx, rdx
0x1a891:      jg      0x1a8aa
0x1a893:      jmp     0x1a8d4
0x1a8d4
0x1a895:      nop     dword ptr [rax]
0x1a898:      movsxd  rax, r15d
0x1a89b:      add     r14, 1
0x1a89f:      add     r15d, 1
0x1a8a3:      cmp     rax, qword ptr [rsp + 0x30]
0x1a8a8:      jge     0x1a8d4
0x1a8aa:      cmp     byte ptr [r14], 0xa
0x1a8ae:      jne     0x1a898
0x1a8b0:      lea     rdi, [r13 + rsi]
0x1a8b5:      mov     byte ptr [r14], 0
0x1a8b9:      cmp     byte ptr [rdi], 0
0x1a8bc:      jne     0x1a8cb
0x1a8be:      add     ebx, 1
0x1a8c1:      cmp     r12d, ebx
0x1a8c4:      jle     0x1a8d4
0x1a8c6:      movsxd  rsi, r15d
0x1a8c9:      jmp     0x1a898
0x1a898
0x1a8cb:      xor     eax, eax
0x1a8cd:      call    0x1a380
0x1a8d2:      jmp     0x1a8be
0x1a8be
0x1a8d4:      mov     rdi, rbp
0x1a8d7:      call    0x18d10
0x1a8dc:      test    r13, r13
0x1a8df:      je      0x1a8e9

```

```

0x1a8e1:      mov     rdi, r13
0x1a8e4:      call    0x18d10
0x1a8e9:      xor     eax, eax
0x1a8eb:      jmp     0x1a7de
0x1a7de
0x1a8f0:      call    0x49d0
0x1a8f5:      nop     word ptr cs:[rax + rax]
0x1a900:      endbr64
0x1a904:      mov     edx, 0xffffffff
0x1a909:      xor     esi, esi
0x1a90b:      xor     eax, eax
0x1a90d:      jmp     0x1a730
0x1a730
0x1a912:      nop     word ptr cs:[rax + rax]
0x1a91d:      nop     dword ptr [rax]
0x1a920:      endbr64
0x1a924:      push    r14
0x1a926:      push    r13
0x1a928:      push    r12
0x1a92a:      mov     r12d, esi
0x1a92d:      push    rbp
0x1a92e:      push    rbx
0x1a92f:      sub     rsp, 0xa0
0x1a936:      mov     rax, qword ptr fs:[0x28]
0x1a93f:      mov     qword ptr [rsp + 0x98], rax
0x1a947:      xor     eax, eax
0x1a949:      call    0x1a080
0x1a94e:      xor     esi, esi
0x1a950:      mov     edx, 0x1b6
0x1a955:      mov     rbp, rax
0x1a958:      mov     rdi, rax
0x1a95b:      xor     eax, eax
0x1a95d:      call    0x4d80
0x1a962:      cmp     eax, -1
0x1a965:      je      0x1aa58
0x1a96b:      mov     rsi, rsp
0x1a96e:      mov     edi, eax
0x1a970:      mov     r14d, eax
0x1a973:      call    0x18ef0
0x1a978:      cmp     eax, -1
0x1a97b:      je      0x1aa58
0x1a981:      mov     eax, dword ptr [rsp + 0x30]
0x1a985:      lea     edi, [rax + 1]
0x1a988:      movsxd  rdi, edi
0x1a98b:      call    0x1d670
0x1a990:      mov     rdx, qword ptr [rsp + 0x30]
0x1a995:      mov     edi, r14d
0x1a998:      mov     rsi, rax

```

```

0x1a99b:    mov     r13, rax
0x1a99e:    call   0x47a0
0x1a9a3:    mov     edi, r14d
0x1a9a6:    mov     rbx, rax
0x1a9a9:    call   0x4ab0
0x1a9ae:    test    ebx, ebx
0x1a9b0:    jle     0x1aa88
0x1a9b6:    lea     eax, [rbx - 1]
0x1a9b9:    cmp     ebx, 1
0x1a9bc:    je      0x1a9f3
0x1a9be:    test    r12d, r12d
0x1a9c1:    je      0x1a9f3
0x1a9c3:    lea     edx, [rbx - 2]
0x1a9c6:    movsxd  rdx, edx
0x1a9c9:    nop     dword ptr [rax]
0x1a9d0:    cmp     byte ptr [r13 + rdx + 1], 0xa
0x1a9d6:    jne     0x1a9dc
0x1a9d8:    sub     r12d, 1
0x1a9dc:    test    r12d, r12d
0x1a9df:    mov     eax, edx
0x1a9e1:    setne   sil
0x1a9e5:    test    edx, edx
0x1a9e7:    setne   cl
0x1a9ea:    sub     rdx, 1
0x1a9ee:    test    sil, cl
0x1a9f1:    jne     0x1a9d0
0x1a9f3:    test    eax, eax
0x1a9f5:    je      0x1aa88
0x1a9fb:    cdqe
0x1a9fd:    jmp     0x1aa08
0x1aa08:
0x1a9ff:    nop
0x1aa00:    sub     rax, 1
0x1aa04:    test    eax, eax
0x1aa06:    je      0x1aa50
0x1aa08:    cmp     byte ptr [r13 + rax], 0xa
0x1aa0e:    mov     ebx, eax
0x1aa10:    jne     0x1aa00
0x1aa12:    mov     edx, 0x1b6
0x1aa17:    mov     esi, 0x201
0x1aa1c:    mov     rdi, rbp
0x1aa1f:    xor     eax, eax
0x1aa21:    call   0x4d80
0x1aa26:    mov     r12d, eax
0x1aa29:    cmp     eax, -1
0x1aa2c:    je      0x1aa50
0x1aa2e:    mov     rdx, qword ptr [rsp + 0x30]
0x1aa33:    lea     esi, [rbx + 1]

```

```

0x1aa36:      mov     edi, eax
0x1aa38:      movsxd  rsi, esi
0x1aa3b:      sub     rdx, rsi
0x1aa3e:      add     rsi, r13
0x1aa41:      call    0x4790
0x1aa46:      mov     edi, r12d
0x1aa49:      call    0x4ab0
0x1aa4e:      nop
0x1aa50:      mov     rdi, r13
0x1aa53:      call    0x18d10
0x1aa58:      mov     rdi, rbp
0x1aa5b:      call    0x18d10
0x1aa60:      mov     rax, qword ptr [rsp + 0x98]
0x1aa68:      sub     rax, qword ptr fs:[0x28]
0x1aa71:      jne     0x1aa8f
0x1aa73:      add     rsp, 0xa0
0x1aa7a:      xor     eax, eax
0x1aa7c:      pop     rbx
0x1aa7d:      pop     rbp
0x1aa7e:      pop     r12
0x1aa80:      pop     r13
0x1aa82:      pop     r14
0x1aa84:      ret
0x1aa85:      nop     dword ptr [rax]
0x1aa88:      test    r13, r13
0x1aa8b:      je      0x1aa58
0x1aa8d:      jmp     0x1aa50
0x1aa50:
0x1aa8f:      call    0x49d0
0x1aa94:      nop     word ptr cs:[rax + rax]
0x1aa9f:      nop
0x1aaa0:      endbr64
0x1aaa4:      mov     r8d, edi
0x1aaa7:      xor     edx, edx
0x1aaa9:      mov     rdi, rsi
0x1aaac:      xor     eax, eax
0x1aaae:      mov     esi, r8d
0x1aab1:      jmp     0x1a130
0x1a130:
0x1aab6:      nop     word ptr cs:[rax + rax]
0x1aac0:      endbr64
0x1aac4:      mov     esi, dword ptr [rip + 0x1101a]
0x1aaca:      mov     edx, 1
0x1aacf:      xor     eax, eax
0x1aad1:      jmp     0x1a130
0x1a130:
0x1aad6:      nop     word ptr cs:[rax + rax]
0x1aae0:      endbr64

```

```

0x1aae4:    movsxd  rdx, dword ptr [rip + 0x11011]
0x1aaeb:    cmp     edx, dword ptr [rip + 0x10ff3]
0x1aaf1:    je      0x1ab08
0x1aaf3:    mov     rax, qword ptr [rip + 0x11006]
0x1aafa:    test    rax, rax
0x1aafd:    je      0x1ab0a
0x1aaff:    mov     rax, qword ptr [rax + rdx*8]
0x1ab03:    ret
0x1ab04:    nop     dword ptr [rax]
0x1ab08:    xor     eax, eax
0x1ab0a:    ret
0x1ab0b:    nop     dword ptr [rax + rax]
0x1ab10:    endbr64
0x1ab14:    mov     eax, dword ptr [rip + 0x10fe2]
0x1ab1a:    xor     r8d, r8d
0x1ab1d:    test    eax, eax
0x1ab1f:    je      0x1ab37
0x1ab21:    sub     eax, 1
0x1ab24:    mov     rdx, qword ptr [rip + 0x10fd5]
0x1ab2b:    mov     dword ptr [rip + 0x10fcb], eax
0x1ab31:    cdqe
0x1ab33:    mov     r8, qword ptr [rdx + rax*8]
0x1ab37:    mov     rax, r8
0x1ab3a:    ret
0x1ab3b:    nop     dword ptr [rax + rax]
0x1ab40:    endbr64
0x1ab44:    mov     eax, dword ptr [rip + 0x10fb2]
0x1ab4a:    cmp     eax, dword ptr [rip + 0x10f94]
0x1ab50:    je      0x1ab70
0x1ab52:    add     eax, 1
0x1ab55:    mov     rdx, qword ptr [rip + 0x10fa4]
0x1ab5c:    mov     dword ptr [rip + 0x10f9a], eax
0x1ab62:    cdqe
0x1ab64:    mov     rax, qword ptr [rdx + rax*8]
0x1ab68:    ret
0x1ab69:    nop     dword ptr [rax]
0x1ab70:    xor     eax, eax
0x1ab72:    ret
0x1ab73:    nop     word ptr cs:[rax + rax]
0x1ab7e:    nop
0x1ab80:    endbr64
0x1ab84:    mov     rax, qword ptr [rip + 0x10f75]
0x1ab8b:    ret
0x1ab8c:    nop     dword ptr [rax]
0x1ab90:    endbr64
0x1ab94:    sub     edi, dword ptr [rip + 0xfc4e]
0x1ab9a:    cmp     dword ptr [rip + 0x10f44], edi
0x1aba0:    jle     0x1abc0

```

```

0x1aba2:    test    edi, edi
0x1aba4:    js      0x1abc0
0x1aba6:    mov     rax, qword ptr [rip + 0x10f53]
0x1abad:    test    rax, rax
0x1abb0:    je      0x1abc2
0x1abb2:    movsxd  rdi, edi
0x1abb5:    mov     rax, qword ptr [rax + rdi*8]
0x1abb9:    ret
0x1abba:    nop     word ptr [rax + rax]
0x1abc0:    xor     eax, eax
0x1abc2:    ret
0x1abc3:    nop     word ptr cs:[rax + rax]
0x1abce:    nop
0x1abd0:    endbr64
0x1abd4:    cmp     edx, dword ptr [rip + 0x10f0a]
0x1abda:    push    rbx
0x1abdb:    mov     ebx, dword ptr [rip + 0x10f1b]
0x1abe1:    jg      0x1abf7
0x1abe3:    test    edx, edx
0x1abe5:    js      0x1abf7
0x1abe7:    cmp     qword ptr [rip + 0x10f11], 0
0x1abef:    je      0x1abf7
0x1abf1:    mov     dword ptr [rip + 0x10f05], edx
0x1abf7:    xor     edx, edx
0x1abf9:    xor     eax, eax
0x1abfb:    call    0x19de0
0x1ac00:    mov     edx, ebx
0x1ac02:    shr     edx, 0x1f
0x1ac05:    cmp     eax, -1
0x1ac08:    je      0x1ac10
0x1ac0a:    mov     eax, dword ptr [rip + 0x10eec]
0x1ac10:    cmp     ebx, dword ptr [rip + 0x10ece]
0x1ac16:    jg      0x1ac2c
0x1ac18:    test    dl, dl
0x1ac1a:    jne     0x1ac2c
0x1ac1c:    cmp     qword ptr [rip + 0x10edc], 0
0x1ac24:    je      0x1ac2c
0x1ac26:    mov     dword ptr [rip + 0x10ed0], ebx
0x1ac2c:    pop     rbx
0x1ac2d:    ret
0x1ac2e:    nop
0x1ac30:    endbr64
0x1ac34:    cmp     dword ptr [rip + 0x10eaa], edi
0x1ac3a:    jl      0x1ac60
0x1ac3c:    test    edi, edi
0x1ac3e:    js      0x1ac60
0x1ac40:    cmp     qword ptr [rip + 0x10eb8], 0
0x1ac48:    je      0x1ac60

```



```

0x1ac4a:    mov     dword ptr [rip + 0x10eac], edi
0x1ac50:    mov     eax, 1
0x1ac55:    ret
0x1ac56:    nop     word ptr cs:[rax + rax]
0x1ac60:    xor     eax, eax
0x1ac62:    ret
0x1ac63:    nop     word ptr cs:[rax + rax]
0x1ac6e:    nop
0x1ac70:    endbr64
0x1ac74:    movsxd  rax, dword ptr [rsi]
0x1ac77:    movzx   r8d, byte ptr [rip + 0xfb6e]
0x1ac7f:    cmp     byte ptr [rdi + rax], r8b
0x1ac83:    jne     0x1afd0
0x1ac89:    push    r15
0x1ac8b:    lea     ecx, [rax + 1]
0x1ac8e:    mov     r9d, edx
0x1ac91:    mov     rdx, rax
0x1ac94:    push    r14
0x1ac96:    movsxd  r10, ecx
0x1ac99:    push    r13
0x1ac9b:    mov     r13, rsi
0x1ac9e:    lea     rsi, [rdi + r10]
0x1aca2:    push    r12
0x1aca4:    push    rbp
0x1aca5:    push    rbx
0x1aca6:    sub     rsp, 0x18
0x1acaa:    movzx   eax, byte ptr [rsi]
0x1acad:    cmp     r8b, al
0x1acb0:    je      0x1af20
0x1acb6:    mov     r8d, 1
0x1acbc:    cmp     al, 0x2d
0x1acbe:    je      0x1aef8
0x1acc4:    lea     edx, [rax - 0x30]
0x1acc7:    cmp     dl, 9
0x1acca:    jbe     0x1ae68
0x1acd0:    mov     ebp, 1
0x1acd5:    xor     ebx, ebx
0x1acd7:    cmp     al, 0x3f
0x1acd9:    jne     0x1acef
0x1acdb:    add     ecx, 1
0x1acde:    xor     ebp, ebp
0x1ace0:    mov     ebx, 1
0x1ace5:    movsxd  r10, ecx
0x1ace8:    lea     rsi, [rdi + r10]
0x1acec:    movzx   eax, byte ptr [rsi]
0x1acef:    test    al, al
0x1acf1:    je      0x1b042
0x1acf7:    lea     rdx, [rdi + r10 + 1]

```

```

0x1acfc:      mov     r12d, ecx
0x1acff:      mov     r8, rsi
0x1ad02:      movabs  r10, 0x400000100000200
0x1ad0c:      jmp     0x1ad48
0x1ad48:
0x1ad0e:      nop
0x1ad10:      cmp     al, 0x3a
0x1ad12:      ja     0x1ad1e
0x1ad14:      bt     r10, rax
0x1ad18:      jb     0x1afd8
0x1ad1e:      movsx   edi, al
0x1ad21:      cmp     edi, r9d
0x1ad24:      je     0x1b010
0x1ad2a:      cmp     al, 0xa
0x1ad2c:      je     0x1ae10
0x1ad32:      mov     r8, rdx
0x1ad35:      movzx   eax, byte ptr [rdx]
0x1ad38:      lea     rdx, [rdx + 1]
0x1ad3c:      add     r12d, 1
0x1ad40:      test    al, al
0x1ad42:      je     0x1ae10
0x1ad48:      test    ebx, ebx
0x1ad4a:      je     0x1ad10
0x1ad4c:      cmp     al, 0xa
0x1ad4e:      je     0x1ae10
0x1ad54:      cmp     al, 0x3f
0x1ad56:      jne     0x1ad32
0x1ad58:      mov     r15d, r12d
0x1ad5b:      mov     qword ptr [rsp + 8], r8
0x1ad60:      sub     r15d, ecx
0x1ad63:      mov     qword ptr [rsp], rsi
0x1ad67:      lea     edi, [r15 + 1]
0x1ad6b:      movsxd  r15, r15d
0x1ad6e:      movsxd  rdi, edi
0x1ad71:      call    0x1d670
0x1ad76:      mov     rsi, qword ptr [rsp]
0x1ad7a:      mov     rdx, r15
0x1ad7d:      mov     rdi, rax
0x1ad80:      mov     r14, rax
0x1ad83:      call    0x4880
0x1ad88:      mov     r8, qword ptr [rsp + 8]
0x1ad8d:      xor     eax, eax
0x1ad8f:      mov     byte ptr [r14 + r15], 0
0x1ad94:      cmp     byte ptr [r8], 0x3f
0x1ad98:      sete    al
0x1ad9b:      add     r12d, eax
0x1ad9e:      nop
0x1ada0:      mov     dword ptr [r13], r12d

```

```

0x1ada4:      mov     r12d, ebx
0x1ada7:      and     r12d, 1
0x1adab:      jmp     0x1add3
0x1add3:
0x1adad:      nop     dword ptr [rax]
0x1adb0:      movsxd  rax, dword ptr [rip + 0x10d45]
0x1adb7:      je      0x1af78
0x1adbd:      test    r12b, r12b
0x1adc0:      jne     0x1af78
0x1adc6:      test    eax, eax
0x1adc8:      je      0x1ade8
0x1adca:      sub     eax, 1
0x1adcd:      mov     dword ptr [rip + 0x10d29], eax
0x1add3:      xor     eax, eax
0x1add5:      mov     edx, ebp
0x1add7:      mov     esi, 0xffffffff
0x1addc:      mov     rdi, r14
0x1addf:      call    0x19de0
0x1ade4:      test    eax, eax
0x1ade6:      jns     0x1adb0
0x1ade8:      mov     eax, dword ptr [rip + 0x10cf6]
0x1adee:      mov     rdi, r14
0x1adf1:      mov     dword ptr [rip + 0x10d05], eax
0x1adf7:      call    0x18d10
0x1adfc:      add     rsp, 0x18
0x1ae00:      xor     eax, eax
0x1ae02:      pop     rbx
0x1ae03:      pop     rbp
0x1ae04:      pop     r12
0x1ae06:      pop     r13
0x1ae08:      pop     r14
0x1ae0a:      pop     r15
0x1ae0c:      ret
0x1ae0d:      nop     dword ptr [rax]
0x1ae10:      mov     r15d, r12d
0x1ae13:      sub     r15d, ecx
0x1ae16:      lea     edi, [r15 + 1]
0x1ae1a:      movsxd  r15, r15d
0x1ae1d:      movsxd  rdi, edi
0x1ae20:      mov     qword ptr [rsp + 8], r8
0x1ae25:      mov     qword ptr [rsp], rsi
0x1ae29:      call    0x1d670
0x1ae2e:      mov     rsi, qword ptr [rsp]
0x1ae32:      mov     rdx, r15
0x1ae35:      mov     rdi, rax
0x1ae38:      mov     r14, rax
0x1ae3b:      call    0x4880
0x1ae40:      test    ebx, ebx

```

```

0x1ae42:    mov     byte ptr [r14 + r15], 0
0x1ae47:    mov     r8, qword ptr [rsp + 8]
0x1ae4c:    je      0x1ada0
0x1ae52:    xor     eax, eax
0x1ae54:    cmp     byte ptr [r8], 0x3f
0x1ae58:    sete    al
0x1ae5b:    add     r12d, eax
0x1ae5e:    jmp     0x1ada0
0x1ada0:
0x1ae63:    nop     dword ptr [rax + rax]
0x1ae68:    lea     edx, [rcx + 1]
0x1ae6b:    xor     ecx, ecx
0x1ae6d:    movsxd  rdx, edx
0x1ae70:    sub     eax, 0x30
0x1ae73:    lea     ecx, [rcx + rcx*4]
0x1ae76:    mov     r9d, edx
0x1ae79:    movsx   eax, al
0x1ae7c:    lea     ecx, [rax + rcx*2]
0x1ae7f:    movzx   eax, byte ptr [rdi + rdx]
0x1ae83:    add     rdx, 1
0x1ae87:    lea     esi, [rax - 0x30]
0x1ae8a:    cmp     sil, 9
0x1ae8e:    jbe     0x1ae70
0x1ae90:    mov     dword ptr [r13], r9d
0x1ae94:    mov     eax, dword ptr [rip + 0x10c4a]
0x1ae9a:    mov     edx, dword ptr [rip + 0xf948]
0x1aea0:    cmp     r8d, -1
0x1aea4:    jne     0x1aead
0x1aea6:    lea     esi, [rax + rdx]
0x1aea9:    sub     esi, ecx
0x1aeab:    mov     ecx, esi
0x1aead:    sub     ecx, edx
0x1aeaf:    cmp     ecx, eax
0x1aeb1:    jge     0x1adfc
0x1aeb7:    test    ecx, ecx
0x1aeb9:    js      0x1adfc
0x1aebf:    mov     rax, qword ptr [rip + 0x10c3a]
0x1aec6:    test    rax, rax
0x1aec9:    je      0x1adfc
0x1aecf:    movsxd  rcx, ecx
0x1aed2:    mov     rax, qword ptr [rax + rcx*8]
0x1aed6:    test    rax, rax
0x1aed9:    je      0x1adfc
0x1aedef:    mov     rax, qword ptr [rax]
0x1aee2:    add     rsp, 0x18
0x1aee6:    pop     rbx
0x1aee7:    pop     rbp
0x1aee8:    pop     r12

```

```

0x1ae68:      pop     r13
0x1ae69:      pop     r14
0x1ae6a:      pop     r15
0x1ae6b:      ret
0x1ae6c:      nop     dword ptr [rax]
0x1ae6d:      lea     ecx, [rdx + 2]
0x1ae6e:      mov     r8d, 0xffffffff
0x1ae6f:      movsxd  r10, ecx
0x1ae70:      lea     rsi, [rdi + r10]
0x1ae71:      movzx   eax, byte ptr [rsi]
0x1ae72:      lea     edx, [rax - 0x30]
0x1ae73:      cmp     dl, 9
0x1ae74:      ja      0x1ae75
0x1ae75:      jmp     0x1ae68
0x1ae76:
0x1ae77:      nop     dword ptr [rax]
0x1ae78:      mov     ecx, dword ptr [rip + 0x10bbe]
0x1ae79:      mov     eax, dword ptr [rip + 0xf8bc]
0x1ae7a:      add     edx, 2
0x1ae7b:      mov     dword ptr [r13], edx
0x1ae7c:      lea     eax, [rcx + rax - 1]
0x1ae7d:      sub     eax, dword ptr [rip + 0xf8ab]
0x1ae7e:      cmp     eax, dword ptr [rip + 0x10ba1]
0x1ae7f:      jge     0x1ae80
0x1ae80:      test    eax, eax
0x1ae81:      js      0x1ae82
0x1ae82:      mov     rdx, qword ptr [rip + 0x10ba8]
0x1ae83:      test    rdx, rdx
0x1ae84:      je      0x1ae85
0x1ae85:      cdqe
0x1ae86:      mov     rax, qword ptr [rdx + rax*8]
0x1ae87:      test    rax, rax
0x1ae88:      jne     0x1ae89
0x1ae89:      jmp     0x1ae8a
0x1ae8a:
0x1ae8b:      nop     dword ptr [rax]
0x1ae8c:      mov     edx, dword ptr [rip + 0x10b66]
0x1ae8d:      cmp     edx, eax
0x1ae8e:      je      0x1ae8f
0x1ae8f:      mov     rbp, qword ptr [rip + 0x10b73]
0x1ae90:      test    rbp, rbp
0x1ae91:      je      0x1ae92
0x1ae92:      mov     rbp, qword ptr [rbp + rax*8]
0x1ae93:      mov     dword ptr [rip + 0x10b5f], edx
0x1ae94:      test    ebx, ebx
0x1ae95:      je      0x1ae96
0x1ae96:      mov     rdi, qword ptr [rip + 0x10b44]
0x1ae97:      test    rdi, rdi

```

```

0x1afaf:      je      0x1afb6
0x1afb1:      call   0x18d10
0x1afb6:      mov     qword ptr [rip + 0x10b33], r14
0x1afbd:      mov     rax, qword ptr [rbp]
0x1afc1:      add     rsp, 0x18
0x1afc5:      pop     rbx
0x1afc6:      pop     rbp
0x1afc7:      pop     r12
0x1afc9:      pop     r13
0x1afcb:      pop     r14
0x1afcd:      pop     r15
0x1afcf:      ret
0x1afd0:      xor     eax, eax
0x1afd2:      ret
0x1afd3:      nop     dword ptr [rax + rax]
0x1afd8:      mov     eax, r12d
0x1afdb:      mov     qword ptr [rsp], rsi
0x1afdf:      sub     eax, ecx
0x1afe1:      movsxd  r15, eax
0x1afe4:      add     eax, 1
0x1afe7:      movsxd  rdi, eax
0x1afea:      call    0xd670
0x1afef:      mov     r14, rax
0x1aff2:      mov     rsi, qword ptr [rsp]
0x1aff6:      mov     rdx, r15
0x1aff9:      mov     rdi, r14
0x1affc:      call    0x4880
0x1b001:      mov     byte ptr [r14 + r15], 0
0x1b006:      jmp     0x1ada0
0x1ada0
0x1b00b:      nop     dword ptr [rax + rax]
0x1b010:      mov     r15d, r12d
0x1b013:      mov     qword ptr [rsp], rsi
0x1b017:      sub     r15d, ecx
0x1b01a:      lea     edi, [r15 + 1]
0x1b01e:      movsxd  r15, r15d
0x1b021:      movsxd  rdi, edi
0x1b024:      call    0xd670
0x1b029:      mov     r14, rax
0x1b02c:      jmp     0x1aff2
0x1aff2
0x1b02e:      mov     rdi, r14
0x1b031:      call    0x18d10
0x1b036:      jmp     0x1afbd
0x1afbd
0x1b03b:      xor     ebp, ebp
0x1b03d:      jmp     0x1af97
0x1af97

```

```

0x1b042:    mov     r12d, ecx
0x1b045:    mov     r8, rsi
0x1b048:    xor     r15d, r15d
0x1b04b:    mov     edi, 1
0x1b050:    jmp     0x1ae20
0x1ae20:
0x1b055:    nop     word ptr cs:[rax + rax]
0x1b060:    endbr64
0x1b064:    push    r15
0x1b066:    push    r14
0x1b068:    push    r13
0x1b06a:    push    r12
0x1b06c:    push    rbp
0x1b06d:    push    rbx
0x1b06e:    sub     rsp, 0x28
0x1b072:    movzx   r14d, byte ptr [rdi]
0x1b076:    test    r14b, r14b
0x1b079:    je      0x1b30d
0x1b07f:    mov     qword ptr [rsp], 0
0x1b087:    mov     r12, rdi
0x1b08a:    mov     r13, rdi
0x1b08d:    xor     ebx, ebx
0x1b08f:    mov     dword ptr [rsp + 8], 0
0x1b097:    xor     ebp, ebp
0x1b099:    mov     dword ptr [rsp + 0xc], 0
0x1b0a1:    jmp     0x1b0c0
0x1b0c0:
0x1b0a3:    nop     dword ptr [rax + rax]
0x1b0a8:    add     ebx, 1
0x1b0ab:    movsxd  rbp, ebx
0x1b0ae:    lea     r13, [r12 + rbp]
0x1b0b2:    movzx   r14d, byte ptr [r13]
0x1b0b7:    test    r14b, r14b
0x1b0ba:    je      0x1b315
0x1b0c0:    cmp     r14b, 0x20
0x1b0c4:    je      0x1b0a8
0x1b0c6:    cmp     r14b, 9
0x1b0ca:    je      0x1b0a8
0x1b0cc:    cmp     byte ptr [rip + 0x10a0d], r14b
0x1b0d3:    je      0x1b315
0x1b0d9:    movsx   r15d, r14b
0x1b0dd:    lea     rdi, [rip + 0x7379]
0x1b0e4:    mov     esi, r15d
0x1b0e7:    call    0x4a00
0x1b0ec:    test    rax, rax
0x1b0ef:    je      0x1b190
0x1b0f5:    add     ebx, 1
0x1b0f8:    mov     r14d, 1

```

```

0x1b0fe:      mov     r9d, 2
0x1b104:      movsxd  rbp, ebx
0x1b107:      lea     rcx, [r12 + rbp]
0x1b10b:      mov     eax, dword ptr [rsp + 8]
0x1b10f:      add     eax, 2
0x1b112:      cmp     eax, dword ptr [rsp + 0xc]
0x1b116:      jge     0x1b2a0
0x1b11c:      movsxd  r8, dword ptr [rsp + 8]
0x1b121:      mov     rdx, qword ptr [rsp]
0x1b125:      mov     rdi, r9
0x1b128:      mov     qword ptr [rsp + 0x18], rcx
0x1b12d:      shl     r8, 3
0x1b131:      lea     r15, [rdx + r8]
0x1b135:      mov     qword ptr [rsp + 0x10], r8
0x1b13a:      call    0x1d670
0x1b13f:      mov     rdx, r14
0x1b142:      mov     rsi, r13
0x1b145:      mov     qword ptr [r15], rax
0x1b148:      mov     rdi, rax
0x1b14b:      call    0x4880
0x1b150:      mov     rax, qword ptr [r15]
0x1b153:      mov     r8, qword ptr [rsp + 0x10]
0x1b158:      mov     rcx, qword ptr [rsp + 0x18]
0x1b15d:      add     dword ptr [rsp + 8], 1
0x1b162:      mov     byte ptr [rax + r14], 0
0x1b167:      mov     rax, qword ptr [rsp]
0x1b16b:      mov     qword ptr [rax + r8 + 8], 0
0x1b174:      movzxx  r14d, byte ptr [rcx]
0x1b178:      test    r14b, r14b
0x1b17b:      je      0x1b315
0x1b181:      mov     r13, rcx
0x1b184:      jmp     0x1b0c0
0x1b0c0:
0x1b189:      nop     dword ptr [rax]
0x1b190:      mov     esi, r15d
0x1b193:      lea     rdi, [rip + 0x72c7]
0x1b19a:      call    0x4a00
0x1b19f:      test    rax, rax
0x1b1a2:      je      0x1b369
0x1b1a8:      movsx   edx, byte ptr [r12 + rbp + 1]
0x1b1ae:      cmp     dl, r14b
0x1b1b1:      je      0x1b2d8
0x1b1b7:      cmp     edx, 0x26
0x1b1ba:      je      0x1b340
0x1b1c0:      cmp     edx, 0x3e
0x1b1c3:      jne     0x1b3ac
0x1b1c9:      cmp     r14b, 0x26
0x1b1cd:      je      0x1b34e

```



```

0x1b1d3:      cmp     r14b, 0x24
0x1b1d7:      jne     0x1b0f5
0x1b1dd:      mov     edx, ebx
0x1b1df:      xor     r15d, r15d
0x1b1e2:      movsxd  r8, edx
0x1b1e5:      lea     rcx, [r12 + r8]
0x1b1e9:      mov     rbp, r8
0x1b1ec:      movzx   eax, byte ptr [rcx]
0x1b1ef:      test    al, al
0x1b1f1:      je      0x1b404
0x1b1f7:      mov     qword ptr [rsp + 0x10], r13
0x1b1fc:      lea     r14, [rip + 0x7265]
0x1b203:      mov     r13, r8
0x1b206:      mov     dword ptr [rsp + 0x18], ebx
0x1b20a:      mov     rbx, r12
0x1b20d:      mov     r12d, edx
0x1b210:      jmp     0x1b265
0x1b265:
0x1b212:      nop     word ptr [rax + rax]
0x1b218:      movsx   ebp, al
0x1b21b:      test    r15d, r15d
0x1b21e:      jne     0x1b290
0x1b220:      mov     esi, ebp
0x1b222:      lea     rdi, [rip + 0x7243]
0x1b229:      call    0x4a00
0x1b22e:      test    rax, rax
0x1b231:      jne     0x1b3c0
0x1b237:      mov     esi, ebp
0x1b239:      mov     rdi, r14
0x1b23c:      call    0x4a00
0x1b241:      lea     esi, [r12 + 1]
0x1b246:      test    rax, rax
0x1b249:      cmovne  r15d, ebp
0x1b24d:      movsxd  rbp, esi
0x1b250:      mov     r12d, esi
0x1b253:      lea     rcx, [rbx + rbp]
0x1b257:      mov     r13, rbp
0x1b25a:      movzx   eax, byte ptr [rcx]
0x1b25d:      test    al, al
0x1b25f:      je      0x1b389
0x1b265:      cmp     al, 0x5c
0x1b267:      jne     0x1b218
0x1b269:      cmp     byte ptr [rbx + r13 + 1], 0xa
0x1b26f:      lea     esi, [r12 + 1]
0x1b274:      je      0x1b330
0x1b27a:      xor     eax, eax
0x1b27c:      cmp     r15d, 0x27
0x1b280:      setne   al

```

```

0x1b283:      add     esi, eax
0x1b285:      jmp     0x1b24d
0x1b24d
0x1b287:      nop     word ptr [rax + rax]
0x1b290:      xor     eax, eax
0x1b292:      cmp     r15d, ebp
0x1b295:      lea     esi, [r12 + 1]
0x1b29a:      cmov     r15d, eax
0x1b29e:      jmp     0x1b24d
0x1b24d
0x1b2a0:      add     dword ptr [rsp + 0xc], 0xa
0x1b2a5:      movsxd   rsi, dword ptr [rsp + 0xc]
0x1b2aa:      mov     rdi, qword ptr [rsp]
0x1b2ae:      mov     qword ptr [rsp + 0x18], rcx
0x1b2b3:      shl     rsi, 3
0x1b2b7:      mov     qword ptr [rsp + 0x10], r9
0x1b2bc:      call    0x1d690
0x1b2c1:      mov     rcx, qword ptr [rsp + 0x18]
0x1b2c6:      mov     r9, qword ptr [rsp + 0x10]
0x1b2cb:      mov     qword ptr [rsp], rax
0x1b2cf:      jmp     0x1b11c
0x1b11c
0x1b2d4:      nop     dword ptr [rax]
0x1b2d8:      cmp     edx, 0x24
0x1b2db:      je      0x1b1d3
0x1b2e1:      mov     eax, ebx
0x1b2e3:      cmp     edx, 0x3c
0x1b2e6:      je      0x1b3f0
0x1b2ec:      add     eax, 2
0x1b2ef:      mov     r14d, eax
0x1b2f2:      movsxd   rbp, eax
0x1b2f5:      sub     r14d, ebx
0x1b2f8:      lea     rcx, [r12 + rbp]
0x1b2fc:      mov     ebx, eax
0x1b2fe:      lea     r9d, [r14 + 1]
0x1b302:      movsxd   r14, r14d
0x1b305:      movsxd   r9, r9d
0x1b308:      jmp     0x1b10b
0x1b10b
0x1b30d:      mov     qword ptr [rsp], 0
0x1b315:      mov     rax, qword ptr [rsp]
0x1b319:      add     rsp, 0x28
0x1b31d:      pop     rbx
0x1b31e:      pop     rbp
0x1b31f:      pop     r12
0x1b321:      pop     r13
0x1b323:      pop     r14
0x1b325:      pop     r15

```

```

0x1b327:      ret
0x1b328:      nop      dword ptr [rax + rax]
0x1b330:      lea      esi, [r12 + 2]
0x1b335:      jmp      0x1b24d
0x1b24d
0x1b33a:      nop      word ptr [rax + rax]
0x1b340:      mov      eax, r14d
0x1b343:      and      eax, 0xffffffff
0x1b346:      cmp      al, 0x3c
0x1b348:      jne      0x1b1d3
0x1b34e:      add      ebx, 2
0x1b351:      mov      r14d, 2
0x1b357:      mov      r9d, 3
0x1b35d:      movsxd   rbp, ebx
0x1b360:      lea      rcx, [r12 + rbp]
0x1b364:      jmp      0x1b10b
0x1b10b
0x1b369:      mov      esi, r15d
0x1b36c:      lea      rdi, [rip + 0x70f5]
0x1b373:      call     0x4a00
0x1b378:      test     rax, rax
0x1b37b:      je       0x1b1dd
0x1b381:      lea      edx, [rbx + 1]
0x1b384:      jmp      0x1b1e2
0x1b1e2
0x1b389:      mov      r12, rbx
0x1b38c:      mov      ebx, dword ptr [rsp + 0x18]
0x1b390:      mov      r14d, esi
0x1b393:      mov      r13, qword ptr [rsp + 0x10]
0x1b398:      sub      r14d, ebx
0x1b39b:      mov      ebx, esi
0x1b39d:      lea      r9d, [r14 + 1]
0x1b3a1:      movsxd   r14, r14d
0x1b3a4:      movsxd   r9, r9d
0x1b3a7:      jmp      0x1b10b
0x1b10b
0x1b3ac:      cmp      r14b, 0x24
0x1b3b0:      jne      0x1b0f5
0x1b3b6:      cmp      edx, 0x28
0x1b3b9:      je       0x1b34e
0x1b3bb:      jmp      0x1b1d3
0x1b1d3
0x1b3c0:      mov      edx, r12d
0x1b3c3:      mov      r12, rbx
0x1b3c6:      mov      ebx, dword ptr [rsp + 0x18]
0x1b3ca:      mov      r8, r13
0x1b3cd:      mov      r14d, edx
0x1b3d0:      mov      r13, qword ptr [rsp + 0x10]

```

```

0x1b3d5:      mov     rbp, r8
0x1b3d8:      lea     rcx, [r12 + r8]
0x1b3dc:      sub     r14d, ebx
0x1b3df:      mov     ebx, edx
0x1b3e1:      lea     r9d, [r14 + 1]
0x1b3e5:      movsxd  r14, r14d
0x1b3e8:      movsxd  r9, r9d
0x1b3eb:      jmp     0x1b10b
0x1b10b
0x1b3f0:      cmp     byte ptr [r12 + rbp + 2], 0x2d
0x1b3f6:      jne     0x1b2ec
0x1b3fc:      lea     eax, [rbx + 1]
0x1b3ff:      jmp     0x1b2ec
0x1b2ec
0x1b404:      mov     r14d, edx
0x1b407:      sub     r14d, ebx
0x1b40a:      mov     ebx, edx
0x1b40c:      lea     r9d, [r14 + 1]
0x1b410:      movsxd  r14, r14d
0x1b413:      movsxd  r9, r9d
0x1b416:      jmp     0x1b10b
0x1b10b
0x1b41b:      nop     dword ptr [rax + rax]
0x1b420:      endbr64
0x1b424:      push    r15
0x1b426:      xor     eax, eax
0x1b428:      push    r14
0x1b42a:      push    r13
0x1b42c:      push    r12
0x1b42e:      mov     r12d, esi
0x1b431:      push    rbp
0x1b432:      push    rbx
0x1b433:      mov     ebx, edi
0x1b435:      mov     rdi, rdx
0x1b438:      sub     rsp, 8
0x1b43c:      call    0x1b060
0x1b441:      test    rax, rax
0x1b444:      je      0x1b598
0x1b44a:      cmp     qword ptr [rax], 0
0x1b44e:      mov     rbp, rax
0x1b451:      je      0x1b590
0x1b457:      mov     eax, 1
0x1b45c:      nop     dword ptr [rax]
0x1b460:      mov     r14d, eax
0x1b463:      add     rax, 1
0x1b467:      cmp     qword ptr [rbp + rax*8 - 8], 0
0x1b46d:      jne     0x1b460
0x1b46f:      lea     eax, [r14 + r12 - 1]

```

```

0x1b474:      test    r12d, r12d
0x1b477:      cmovs   r12d, eax
0x1b47b:      lea     eax, [r14 + rbx - 1]
0x1b480:      test    ebx, ebx
0x1b482:      cmovs   ebx, eax
0x1b485:      lea     eax, [r12 + 1]
0x1b48a:      cmp     r12d, 0x24
0x1b48e:      jne     0x1b497
0x1b490:      lea     r12d, [r14 - 1]
0x1b494:      mov     eax, r14d
0x1b497:      lea     edx, [r14 - 1]
0x1b49b:      cmp     ebx, 0x24
0x1b49e:      cmov     ebx, edx
0x1b4a1:      mov     edx, eax
0x1b4a3:      cmp     ebx, eax
0x1b4a5:      cmovge   edx, ebx
0x1b4a8:      xor     r13d, r13d
0x1b4ab:      cmp     edx, r14d
0x1b4ae:      jg      0x1b547
0x1b4b4:      mov     ecx, ebx
0x1b4b6:      or      ecx, eax
0x1b4b8:      js      0x1b547
0x1b4be:      cmp     ebx, eax
0x1b4c0:      jge     0x1b59d
0x1b4c6:      movsxd   rbx, ebx
0x1b4c9:      xor     r15d, r15d
0x1b4cc:      mov     r13, rbx
0x1b4cf:      nop
0x1b4d0:      mov     rdi, qword ptr [rbp + r13*8]
0x1b4d5:      add     r13, 1
0x1b4d9:      call    0x49b0
0x1b4de:      lea     r15d, [r15 + rax + 1]
0x1b4e3:      lea     eax, [r13 - 1]
0x1b4e7:      cmp     r12d, eax
0x1b4ea:      jg      0x1b4d0
0x1b4ec:      lea     edi, [r15 + 1]
0x1b4f0:      xor     r15d, r15d
0x1b4f3:      movsxd   rdi, edi
0x1b4f6:      call    0x1d670
0x1b4fb:      mov     r13, rax
0x1b4fe:      nop
0x1b500:      mov     rsi, qword ptr [rbp + rbx*8]
0x1b505:      movsxd   rdi, r15d
0x1b508:      add     rdi, r13
0x1b50b:      call    0x48b0
0x1b510:      mov     rdi, qword ptr [rbp + rbx*8]
0x1b515:      call    0x49b0
0x1b51a:      add     r15d, eax

```

```

0x1b51d:      cmp     r12d, ebx
0x1b520:      jle     0x1b53b
0x1b522:      lea     eax, [r15 + 1]
0x1b526:      movsxd  r15, r15d
0x1b529:      movsxd  rdx, eax
0x1b52c:      mov     byte ptr [r13 + r15], 0x20
0x1b532:      mov     r15d, eax
0x1b535:      mov     byte ptr [r13 + rdx], 0
0x1b53b:      add     rbx, 1
0x1b53f:      lea     eax, [rbx - 1]
0x1b542:      cmp     r12d, eax
0x1b545:      jg      0x1b500
0x1b547:      test    r14d, r14d
0x1b54a:      je      0x1b571
0x1b54c:      lea     eax, [r14 - 1]
0x1b550:      mov     rbx, rbp
0x1b553:      lea     r12, [rbp + rax*8 + 8]
0x1b558:      nop     dword ptr [rax + rax]
0x1b560:      mov     rdi, qword ptr [rbx]
0x1b563:      add     rbx, 8
0x1b567:      call    0x18d10
0x1b56c:      cmp     r12, rbx
0x1b56f:      jne     0x1b560
0x1b571:      mov     rdi, rbp
0x1b574:      call    0x18d10
0x1b579:      add     rsp, 8
0x1b57d:      mov     rax, r13
0x1b580:      pop     rbx
0x1b581:      pop     rbp
0x1b582:      pop     r12
0x1b584:      pop     r13
0x1b586:      pop     r14
0x1b588:      pop     r15
0x1b58a:      ret
0x1b58b:      nop     dword ptr [rax + rax]
0x1b590:      xor     r14d, r14d
0x1b593:      jmp     0x1b46f
0x1b46f
0x1b598:      xor     r13d, r13d
0x1b59b:      jmp     0x1b579
0x1b579
0x1b59d:      mov     edi, 1
0x1b5a2:      call    0x1d670
0x1b5a7:      mov     r13, rax
0x1b5aa:      jmp     0x1b547
0x1b547
0x1b5ac:      nop     dword ptr [rax]
0x1b5b0:      endbr64

```

```

0x1b5b4:    push    r15
0x1b5b6:    push    r14
0x1b5b8:    push    r13
0x1b5ba:    push    r12
0x1b5bc:    push    rbp
0x1b5bd:    push    rbx
0x1b5be:    mov     rbx, rdi
0x1b5c1:    sub     rsp, 0x88
0x1b5c8:    mov     qword ptr [rsp + 0x20], rdi
0x1b5cd:    mov     qword ptr [rsp + 0x30], rsi
0x1b5d2:    mov     rax, qword ptr fs:[0x28]
0x1b5db:    mov     qword ptr [rsp + 0x78], rax
0x1b5e0:    xor     eax, eax
0x1b5e2:    cmp     byte ptr [rip + 0xf204], 0
0x1b5e9:    je      0x1b656
0x1b5eb:    mov     edi, 0x100
0x1b5f0:    call    0x1d670
0x1b5f5:    mov     rbx, qword ptr [rsp + 0x20]
0x1b5fa:    mov     byte ptr [rax], 0
0x1b5fd:    mov     r15, rax
0x1b600:    mov     rdi, rbx
0x1b603:    call    0x49b0
0x1b608:    mov     dword ptr [rsp + 0xc], eax
0x1b60c:    mov     r12, rax
0x1b60f:    movzx   eax, byte ptr [rbx]
0x1b612:    cmp     al, byte ptr [rip + 0xf1d4]
0x1b618:    je      0x1c1e0
0x1b61e:    movzx   ebp, byte ptr [rip + 0xf1c8]
0x1b625:    mov     r13, qword ptr [rip + 0xf1c4]
0x1b62c:    lea     rbx, [rbx + 1]
0x1b630:    test    al, al
0x1b632:    jne     0x1b6c7
0x1b638:    nop     dword ptr [rax + rax]
0x1b640:    test    bpl, bpl
0x1b643:    je      0x1b6d8
0x1b649:    mov     rdi, r15
0x1b64c:    call    0x18d10
0x1b651:    mov     rbx, qword ptr [rsp + 0x20]
0x1b656:    mov     rdi, rbx
0x1b659:    call    0x49b0
0x1b65e:    lea     rdi, [rax + 1]
0x1b662:    call    0x1d670
0x1b667:    mov     rsi, rbx
0x1b66a:    mov     rdi, rax
0x1b66d:    call    0x48b0
0x1b672:    mov     rcx, qword ptr [rsp + 0x30]
0x1b677:    mov     qword ptr [rcx], rax
0x1b67a:    xor     eax, eax

```

```

0x1b67c:    mov     rdx, qword ptr [rsp + 0x78]
0x1b681:    sub     rdx, qword ptr fs:[0x28]
0x1b68a:    jne     0x1c487
0x1b690:    add     rsp, 0x88
0x1b697:    pop     rbx
0x1b698:    pop     rbp
0x1b699:    pop     r12
0x1b69b:    pop     r13
0x1b69d:    pop     r14
0x1b69f:    pop     r15
0x1b6a1:    ret
0x1b6a2:    nop     word ptr [rax + rax]
0x1b6a8:    mov     rdi, r13
0x1b6ab:    call    0x4a00
0x1b6b0:    test    rax, rax
0x1b6b3:    je      0x1b6d8
0x1b6b5:    nop     dword ptr [rax]
0x1b6b8:    movzx   eax, byte ptr [rbx]
0x1b6bb:    add     rbx, 1
0x1b6bf:    test    al, al
0x1b6c1:    je      0x1b640
0x1b6c7:    movsx   esi, byte ptr [rbx]
0x1b6ca:    cmp     al, bpl
0x1b6cd:    jne     0x1b6b8
0x1b6cf:    test    esi, esi
0x1b6d1:    je      0x1b6b8
0x1b6d3:    jmp     0x1b6a8
0x1b6a8:
0x1b6d5:    nop     dword ptr [rax]
0x1b6d8:    mov     rbp, qword ptr [rsp + 0x20]
0x1b6dd:    test    r12d, r12d
0x1b6e0:    jle     0x1c40a
0x1b6e6:    mov     dword ptr [rsp + 4], 0x100
0x1b6ee:    xor     r13d, r13d
0x1b6f1:    xor     ebx, ebx
0x1b6f3:    mov     dword ptr [rsp + 0x28], 0
0x1b6fb:    mov     dword ptr [rsp + 0x3c], 0
0x1b703:    mov     dword ptr [rsp + 8], 0
0x1b70b:    jmp     0x1b758
0x1b758:
0x1b70d:    nop     dword ptr [rax]
0x1b710:    cmp     r12b, byte ptr [rip + 0xf0d6]
0x1b717:    je      0x1b7d0
0x1b71d:    cmp     r12b, 0xfd
0x1b721:    je      0x1b7d0
0x1b727:    cmp     r12b, 0x5c
0x1b72b:    je      0x1bde0
0x1b731:    cmp     ebx, dword ptr [rsp + 4]

```



```

0x1b735:    jge     0x1bfe0
0x1b73b:    lea     eax, [rbx + 1]
0x1b73e:    mov     byte ptr [r15 + rbx], r12b
0x1b742:    movsxd  rdx, eax
0x1b745:    movsxd  rbx, eax
0x1b748:    mov     byte ptr [r15 + rdx], 0
0x1b74d:    add     r13d, 1
0x1b751:    cmp     r13d, dword ptr [rsp + 0xc]
0x1b756:    jge     0x1b79d
0x1b758:    movsxd  r14, r13d
0x1b75b:    mov     r10d, dword ptr [rsp + 8]
0x1b760:    lea     rdx, [rbp + r14]
0x1b765:    movzx   r12d, byte ptr [rdx]
0x1b769:    test    r10d, r10d
0x1b76c:    je      0x1b710
0x1b76e:    cmp     ebx, dword ptr [rsp + 4]
0x1b772:    jge     0x1be10
0x1b778:    lea     eax, [rbx + 1]
0x1b77b:    mov     byte ptr [r15 + rbx], r12b
0x1b77f:    add     r13d, 1
0x1b783:    movsxd  rdx, eax
0x1b786:    mov     dword ptr [rsp + 8], 0
0x1b78e:    movsxd  rbx, eax
0x1b791:    mov     byte ptr [r15 + rdx], 0
0x1b796:    cmp     r13d, dword ptr [rsp + 0xc]
0x1b79b:    jl      0x1b758
0x1b79d:    mov     rax, qword ptr [rsp + 0x30]
0x1b7a2:    mov     qword ptr [rax], r15
0x1b7a5:    cmp     qword ptr [rsp + 0x20], rbp
0x1b7aa:    je      0x1b7b4
0x1b7ac:    mov     rdi, rbp
0x1b7af:    call    0x18d10
0x1b7b4:    mov     ecx, dword ptr [rsp + 0x28]
0x1b7b8:    test    ecx, ecx
0x1b7ba:    jne     0x1c35b
0x1b7c0:    mov     edx, dword ptr [rsp + 0x3c]
0x1b7c4:    xor     eax, eax
0x1b7c6:    test    edx, edx
0x1b7c8:    setne   al
0x1b7cb:    jmp     0x1b67c
0x1b67c:
0x1b7d0:    lea     rcx, [rbp + r14 + 1]
0x1b7d5:    mov     qword ptr [rsp + 0x10], rdx
0x1b7da:    movsx   esi, byte ptr [rcx]
0x1b7dd:    test    esi, esi
0x1b7df:    je      0x1b830
0x1b7e1:    mov     rdi, qword ptr [rip + 0xf008]
0x1b7e8:    mov     qword ptr [rsp + 0x18], rcx

```

```

0x1b7ed:      call    0x4a00
0x1b7f2:      mov     rcx, qword ptr [rsp + 0x18]
0x1b7f7:      test    rax, rax
0x1b7fa:      je       0x1b830
0x1b7fc:      cmp     ebx, dword ptr [rsp + 4]
0x1b800:      jl       0x1b73b
0x1b806:      add     dword ptr [rsp + 4], 0x40
0x1b80b:      movsxd   rsi, dword ptr [rsp + 4]
0x1b810:      mov     rdi, r15
0x1b813:      call    0x1d690
0x1b818:      mov     rdx, qword ptr [rsp + 0x10]
0x1b81d:      mov     r15, rax
0x1b820:      movzx    r12d, byte ptr [rdx]
0x1b824:      jmp     0x1b73b
0x1b73b:
0x1b829:      nop     dword ptr [rax]
0x1b830:      mov     edi, 0x80
0x1b835:      mov     qword ptr [rsp + 0x18], rcx
0x1b83a:      call    0x1d670
0x1b83f:      mov     rcx, qword ptr [rsp + 0x18]
0x1b844:      mov     dword ptr [rsp + 0x6c], r13d
0x1b849:      mov     qword ptr [rsp + 0x10], rax
0x1b84e:      movzx    r12d, byte ptr [rcx]
0x1b852:      test    r12b, r12b
0x1b855:      je       0x1be30
0x1b85b:      movsx    esi, r12b
0x1b85f:      lea     rdi, [rip + 0x6c11]
0x1b866:      call    0x4a00
0x1b86b:      test    rax, rax
0x1b86e:      je       0x1bfc0
0x1b874:      lea     eax, [r13 + 1]
0x1b878:      lea     rsi, [rsp + 0x70]
0x1b87d:      xor     edx, edx
0x1b87f:      mov     byte ptr [rsp + 0x77], 0
0x1b884:      mov     dword ptr [rsp + 0x6c], eax
0x1b888:      movzx    eax, byte ptr [rip + 0xef5e]
0x1b88f:      lea     rdi, [rsp + 0x75]
0x1b894:      mov     dword ptr [rsp + 0x70], 0
0x1b89c:      mov     byte ptr [rsp + 0x76], al
0x1b8a0:      mov     byte ptr [rsp + 0x75], al
0x1b8a4:      xor     eax, eax
0x1b8a6:      call    0x1ac70
0x1b8ab:      mov     r12d, dword ptr [rsp + 0x6c]
0x1b8b0:      mov     r14, rax
0x1b8b3:      test    r14, r14
0x1b8b6:      je       0x1c435
0x1b8bc:      movsxd   rax, r12d
0x1b8bf:      xor     esi, esi

```

```

0x1b8c1:      mov     edx, r12d
0x1b8c4:      movzx   eax, byte ptr [rbp + rax]
0x1b8c9:      cmp     al, 0x3a
0x1b8cb:      jne     0x1b8df
0x1b8cd:      lea     edx, [r12 + 1]
0x1b8d2:      mov     esi, 1
0x1b8d7:      movsxd   rax, edx
0x1b8da:      movzx   eax, byte ptr [rbp + rax]
0x1b8df:      cmp     al, 0x25
0x1b8e1:      je      0x1c010
0x1b8e7:      cmp     al, 0x2d
0x1b8e9:      je      0x1c100
0x1b8ef:      jg      0x1bd48
0x1b8f5:      cmp     al, 0x24
0x1b8f7:      je      0x1bf80
0x1b8fd:      cmp     al, 0x2a
0x1b8ff:      jne     0x1bd54
0x1b905:      add     edx, 1
0x1b908:      mov     esi, 0x24
0x1b90d:      mov     edi, 1
0x1b912:      xor     eax, eax
0x1b914:      mov     dword ptr [rsp + 0x6c], edx
0x1b918:      mov     rdx, r14
0x1b91b:      call    0x1b420
0x1b920:      mov     r13, rax
0x1b923:      test    rax, rax
0x1b926:      je      0x1c080
0x1b92c:      lea     rax, [rip + 0x101ed]
0x1b933:      cmp     r13, rax
0x1b936:      je      0x1bdae
0x1b93c:      mov     rdi, r13
0x1b93f:      call    0x49b0
0x1b944:      lea     rdi, [rax + 1]
0x1b948:      call    0x1d670
0x1b94d:      mov     rsi, r13
0x1b950:      mov     rdi, rax
0x1b953:      call    0x48b0
0x1b958:      mov     rdi, r13
0x1b95b:      mov     r14, rax
0x1b95e:      call    0x18d10
0x1b963:      movsxd   rcx, dword ptr [rsp + 0x6c]
0x1b968:      cmp     byte ptr [rbp + rcx], 0x3a
0x1b96d:      mov     dword ptr [rsp + 0x44], ecx
0x1b971:      jne     0x1c3d4
0x1b977:      mov     dword ptr [rsp + 0x18], 0
0x1b97f:      lea     r12, [rip + 0x6afe]
0x1b986:      mov     dword ptr [rsp + 0x38], 0
0x1b98e:      mov     dword ptr [rsp + 0x40], ebx

```

```

0x1b992:      mov     qword ptr [rsp + 0x28], r15
0x1b997:      mov     r15, r14
0x1b99a:      mov     r14d, ecx
0x1b99d:      nop     dword ptr [rax]
0x1b9a0:      movzx   eax, byte ptr [rbp + rcx + 1]
0x1b9a5:      cmp     al, 0x67
0x1b9a7:      jne     0x1b9bf
0x1b9a9:      add     r14d, 1
0x1b9ad:      mov     dword ptr [rsp + 0x18], 1
0x1b9b5:      mov     dword ptr [rsp + 0x6c], r14d
0x1b9ba:      movzx   eax, byte ptr [rbp + rcx + 2]
0x1b9bf:      cmp     al, 0x26
0x1b9c1:      je      0x1be70
0x1b9c7:      sub     eax, 0x65
0x1b9ca:      cmp     al, 0xf
0x1b9cc:      ja      0x1b9e0
0x1b9ce:      movzx   eax, al
0x1b9d1:      movsxd  rax, dword ptr [r12 + rax*4]
0x1b9d5:      add     rax, r12
0x1b9d8:      jmp     rax
rax
                                ->->->                                indirect JMP
0x7518: ret
0x7519: nop     dword ptr [rax]
0x7520: endbr64
0x7524: cmp     byte ptr [rip + 0x2335d], 0
0x752b: jne     0x7558
0x752d: push    rbp
0x752e: cmp     qword ptr [rip + 0x21ac2], 0
0x7536: mov     rbp, rsp
0x7539: je      0x7547
0x753b: mov     rdi, qword ptr [rip + 0x21ac6]
0x7542: call    0x47b0
0x7547: call    0x74b0
0x754c: mov     byte ptr [rip + 0x23335], 1
0x7553: pop     rbp
0x7554: ret
0x7555: nop     dword ptr [rax]
0x7558: ret
0x7559: nop     dword ptr [rax]
0x7560: endbr64
0x7564: jmp     0x74e0
0x74e0
0x7569: nop     dword ptr [rax]
0x7570: push    r12
0x7572: lea     eax, [rdi + 0x42]
0x7575: mov     r12d, esi
0x7578: sub     rsp, 0x10

```

```

0x757c: cmp     eax, 0x17
0x757f: ja      0x77e0
0x7585: lea     rdx, [rip + 0x18a78]
0x758c: movsxd  rax, dword ptr [rdx + rax*4]
0x7590: add     rax, rdx
0x7593: jmp     rax
rax
                                ->->->                                indirect JMP
0x925c: mov     eax, 1
0x9261: jmp     0x8d3f
0x8d3f
0x9266: mov     byte ptr [rbp], 0x7d
0x926a: mov     rdi, r10
0x926d: mov     dword ptr [rsp + 8], eax
0x9271: call    0x18d10
0x9276: mov     eax, dword ptr [rsp + 8]
0x927a: test    eax, eax
0x927c: je      0x8c38
0x9282: jmp     0x8d61
0x8d61
0x9287: nop     word ptr [rax + rax]
0x9290: mov     byte ptr [rbp], 0x2c
0x9294: je      0x8c38
0x929a: jmp     0x8d61
0x8d61
0x929f: call    0x49d0
0x92a4: nop     word ptr cs:[rax + rax]
0x92af: nop
0x92b0: endbr64
0x92b4: push    r13
0x92b6: mov     r13, rdx
0x92b9: push    r12
0x92bb: mov     r12, rsi
0x92be: push    rbp
0x92bf: mov     rbp, rdi
0x92c2: call    0x47e0
0x92c7: test    rax, rax
0x92ca: je      0x92d8
0x92cc: pop     rbp
0x92cd: pop     r12
0x92cf: pop     r13
0x92d1: ret
0x92d2: nop     word ptr [rax + rax]
0x92d8: test    r12, r12
0x92db: je      0x9300
0x92dd: mov     rdi, r12
0x92e0: call    0x47e0
0x92e5: mov     rsi, rax

```

```

0x92e8: test    rax, rax
0x92eb: je      0x9300
0x92ed: mov     rdi, rbp
0x92f0: pop     rbp
0x92f1: pop     r12
0x92f3: pop     r13
0x92f5: jmp     0x18380
0x18380
0x92fa: nop     word ptr [rax + rax]
0x9300: mov     rsi, r13
0x9303: mov     rdi, rbp
0x9306: pop     rbp
0x9307: pop     r12
0x9309: pop     r13
0x930b: jmp     0x18380
0x18380
0x9310: endbr64
0x9314: push    r15
0x9316: push    r14
0x9318: push    r13
0x931a: push    r12
0x931c: push    rbp
0x931d: push    rbx
0x931e: mov     ebx, edi
0x9320: sub     rsp, 0xb8
0x9327: mov     qword ptr [rsp + 0x18], rsi
0x932c: mov     rax, qword ptr fs:[0x28]
0x9335: mov     qword ptr [rsp + 0xa8], rax
0x933d: xor     eax, eax
0x933f: mov     dword ptr [rsi], 0
0x9345: cmp     edi, 0x7ff
0x934b: jg      0x95d8
0x9351: lea     rax, [rsp + 0x20]
0x9356: lea     r12, [rsp + 0x50]
0x935b: mov     qword ptr [rsp + 8], rax
0x9360: lea     r13, [rip + 0x2089e]
0x9367: jmp     0x94df
0x94df
0x936c: nop     dword ptr [rax]
0x9370: mov     rdi, r12
0x9373: call    0xbd80
0x9378: test    rax, rax
0x937b: je      0x9508
0x9381: cmp     byte ptr [rax], 0
0x9384: je      0x94d0
0x938a: mov     edx, 0x50
0x938f: mov     rsi, rax
0x9392: mov     rdi, r12

```

```

0x9395: call    0x4cf0
0x939a: mov     dword ptr [rsp + 4], 0
0x93a2: mov     edi, 1
0x93a7: mov     esi, 0x30
0x93ac: call    0x1d740
0x93b1: mov     rdi, qword ptr [rsp + 0x20]
0x93b6: mov     rbp, rax
0x93b9: test    rdi, rdi
0x93bc: je       0x9560
0x93c2: call    0x1d7a0
0x93c7: mov     edi, 8
0x93cc: mov     qword ptr [rbp], rax
0x93d0: call    0x19100
0x93d5: mov     rdi, qword ptr [rsp + 0x30]
0x93da: mov     qword ptr [rbp + 0x28], rax
0x93de: test    rdi, rdi
0x93e1: je       0x93ec
0x93e3: call    0x1d7a0
0x93e8: mov     qword ptr [rbp + 0x10], rax
0x93ec: mov     rdi, qword ptr [rsp + 0x28]
0x93f1: test    rdi, rdi
0x93f4: je       0x9474
0x93f6: call    0x1d7a0
0x93fb: mov     qword ptr [rbp + 8], rax
0x93ff: mov     r14, qword ptr [rsp + 0x38]
0x9404: mov     eax, 1
0x9409: test    r14, r14
0x940c: je       0x9421
0x940e: call    0x4ef0
0x9413: movsx    rdx, byte ptr [r14]
0x9417: mov     rax, qword ptr [rax]
0x941a: cmp     dword ptr [rax + rdx*4], 0x79
0x941e: sete    al
0x9421: mov     byte ptr [rbp + 0x18], al
0x9424: mov     rdx, qword ptr [rsp + 0x40]
0x9429: test    rdx, rdx
0x942c: je       0x944c
0x942e: mov     qword ptr [rsp + 0x10], rdx
0x9433: call    0x4ef0
0x9438: mov     rdx, qword ptr [rsp + 0x10]
0x943d: mov     rax, qword ptr [rax]
0x9440: movsx    rdx, byte ptr [rdx]
0x9444: cmp     dword ptr [rax + rdx*4], 0x79
0x9448: sete    byte ptr [rbp + 0x19]
0x944c: mov     rdx, qword ptr [rsp + 0x48]
0x9451: test    rdx, rdx
0x9454: je       0x9474
0x9456: mov     qword ptr [rsp + 0x10], rdx

```

```

0x945b: call    0x4ef0
0x9460: mov     rdx, qword ptr [rsp + 0x10]
0x9465: mov     rax, qword ptr [rax]
0x9468: movsx   rdx, byte ptr [rdx]
0x946c: cmp     dword ptr [rax + rdx*4], 0x79
0x9470: sete    byte ptr [rbp + 0x1a]
0x9474: call    0x4f00
0x9479: mov     r15, qword ptr [rbp]
0x947d: mov     rax, qword ptr [rax]
0x9480: movsx   rdx, byte ptr [r15]
0x9484: test    byte ptr [rax + rdx*2 + 1], 2
0x9489: jne     0x9518
0x948f: mov     rdi, r12
0x9492: call    0x1d7a0
0x9497: cmp     byte ptr [rbp + 0x1b], 0
0x949b: mov     qword ptr [rbp + 0x20], rax
0x949f: jne     0x94ac
0x94a1: cmp     qword ptr [rbp + 8], 0
0x94a6: je      0x9590
0x94ac: mov     eax, dword ptr [rsp + 4]
0x94b0: test    eax, eax
0x94b2: je      0x94c5
0x94b4: mov     rdi, r12
0x94b7: call    0x9f00
0x94bc: test    rax, rax
0x94bf: je      0x95a0
0x94c5: mov     rsi, rbp
0x94c8: mov     rdi, r12
0x94cb: call    0xaf70
0x94d0: add     ebx, 1
0x94d3: cmp     ebx, 0x800
0x94d9: je      0x9580
0x94df: mov     rsi, qword ptr [rsp + 8]
0x94e4: xor     ecx, ecx
0x94e6: mov     edx, 6
0x94eb: mov     rdi, r12
0x94ee: call    0x17320
0x94f3: movzx   eax, byte ptr [rsp + 0x50]
0x94f8: test    al, al
0x94fa: je      0x95d8
0x9500: cmp     al, 0x5e
0x9502: jne     0x9370
0x9508: mov     dword ptr [rsp + 4], 1
0x9510: jmp     0x93a2
0x93a2:
0x9515: nop     dword ptr [rax]
0x9518: lea     r14, [rip + 0x1fb01]
0x951f: jmp     0x9535

```



```

0x9535
0x9521: nop      dword ptr [rax]
0x9528: add      r14, 0x23
0x952c: cmp      r13, r14
0x952f: je       0x948f
0x9535: mov      rsi, r14
0x9538: mov      rdi, r15
0x953b: call     0x4b30
0x9540: test     eax, eax
0x9542: jne     0x9528
0x9544: mov      rdi, r15
0x9547: call     0x18d10
0x954c: mov      qword ptr [rbp], r14
0x9550: mov      byte ptr [rbp + 0x1b], 1
0x9554: jmp      0x948f
0x948f
0x9559: nop      dword ptr [rax]
0x9560: mov      rdi, rax
0x9563: add      ebx, 1
0x9566: call     0x18d10
0x956b: cmp      ebx, 0x800
0x9571: jne     0x94df
0x9577: nop      word ptr [rax + rax]
0x9580: mov      eax, 0x800
0x9585: jmp      0x95da
0x95da
0x9587: nop      word ptr [rax + rax]
0x9590: cmp      qword ptr [rbp + 0x10], 0
0x9595: jne     0x94ac
0x959b: jmp      0x94d0
0x94d0
0x95a0: mov      rcx, qword ptr [rip + 0x21341]
0x95a7: mov      rdi, qword ptr [rip + 0x212d2]
0x95ae: mov      r8, r12
0x95b1: xor      eax, eax
0x95b3: lea      rdx, [rip + 0x1720e]
0x95ba: mov      esi, 1
0x95bf: call     0x4e60
0x95c4: mov      rax, qword ptr [rsp + 0x18]
0x95c9: add      dword ptr [rax], 1
0x95cc: jmp      0x94d0
0x94d0
0x95d1: nop      dword ptr [rax]
0x95d8: mov      eax, ebx
0x95da: mov      rdx, qword ptr [rsp + 0xa8]
0x95e2: sub      rdx, qword ptr fs:[0x28]
0x95eb: jne     0x95ff
0x95ed: add      rsp, 0xb8

```

```

0x95f4: pop    rbx
0x95f5: pop    rbp
0x95f6: pop    r12
0x95f8: pop    r13
0x95fa: pop    r14
0x95fc: pop    r15
0x95fe: ret
0x95ff: call   0x49d0
0x9604: nop    word ptr cs:[rax + rax]
0x960f: nop
0x9610: endbr64
0x9614: sub     rsp, 8
0x9618: xor     edi, edi
0x961a: call   0x9b00
0x961f: xor     eax, eax
0x9621: call   0xa6c0
0x9626: mov     rdi, qword ptr [rip + 0x2129b]
0x962d: add     rsp, 8
0x9631: jmp     0xbb40
0xbb40:
0x9636: nop    word ptr cs:[rax + rax]
0x9640: endbr64
0x9644: push    rbp
0x9645: lea     rbp, [rip + 0x205b9]
0x964c: push    rbx
0x964d: mov     rdi, rbp
0x9650: sub     rsp, 0x48
0x9654: mov     rax, qword ptr fs:[0x28]
0x965d: mov     qword ptr [rsp + 0x38], rax
0x9662: xor     eax, eax
0x9664: call   0x49b0
0x9669: mov     rdi, qword ptr [rip + 0x21240]
0x9670: mov     rbx, rax
0x9673: call   0x11100
0x9678: mov     edx, 0x33
0x967d: mov     rsi, rsp
0x9680: sub     edx, ebx
0x9682: mov     rdi, rax
0x9685: call   0x182b0
0x968a: mov     rsi, rbp
0x968d: mov     rdi, rax
0x9690: call   0x4df0
0x9695: mov     rdi, rax
0x9698: call   0xd540
0x969d: mov     rax, qword ptr [rsp + 0x38]
0x96a2: sub     rax, qword ptr fs:[0x28]
0x96ab: jne     0x96b4
0x96ad: add     rsp, 0x48

```

```

0x96b1: pop      rbx
0x96b2: pop      rbp
0x96b3: ret
0x96b4: call     0x49d0
0x96b9: nop       dword ptr [rax]
0x96c0: endbr64
0x96c4: sub       rsp, 8
0x96c8: mov       r8d, 1
0x96ce: mov       rdx, qword ptr [rip + 0x211db]
0x96d5: mov       rdi, qword ptr [rip + 0x211cc]
0x96dc: mov       rcx, -1
0x96e3: mov       esi, 0x22
0x96e8: call     0x13e30
0x96ed: mov       r8d, 1
0x96f3: mov       rdx, qword ptr [rip + 0x211ae]
0x96fa: mov       rdi, qword ptr [rip + 0x211af]
0x9701: mov       rcx, -1
0x9708: mov       esi, 0x22
0x970d: add       rsp, 8
0x9711: jmp      0x13e30
0x13e30
0x9716: nop       word ptr cs:[rax + rax]
0x9720: endbr64
0x9724: sub       rsp, 8
0x9728: mov       rdx, qword ptr [rip + 0x211b9]
0x972f: mov       edi, 1
0x9734: xor       eax, eax
0x9736: lea       rsi, [rip + 0x170b3]
0x973d: call     0x4d20
0x9742: lea       rdi, [rip + 0x170cf]
0x9749: call     0x48d0
0x974e: lea       rdi, [rip + 0x170eb]
0x9755: call     0x48d0
0x975a: lea       rdi, [rip + 0x16dde]
0x9761: call     0x48d0
0x9766: lea       rdi, [rip + 0x170fb]
0x976d: call     0x48d0
0x9772: lea       rdi, [rip + 0x17117]
0x9779: call     0x48d0
0x977e: lea       rdi, [rip + 0x1713b]
0x9785: add       rsp, 8
0x9789: jmp      0x48d0
0x48d0
0x978e: nop
0x9790: cmp       dword ptr [rip + 0x21ba9], 1
0x9797: push     rbx
0x9798: mov       ebx, edi
0x979a: je        0x9830

```

```

0x97a0: cmp     ebx, 1
0x97a3: je      0x984a
0x97a9: mov     rdi, qword ptr [rip + 0x20598]
0x97b0: test    rdi, rdi
0x97b3: je      0x97c6
0x97b5: lea     rdx, [rip + 0x2b4]
0x97bc: mov     esi, 1
0x97c1: call    0x4820
0x97c6: movzx   edx, byte ptr [rip + 0x21b83]
0x97cd: mov     dword ptr [rip + 0x21b75], 0
0x97d7: mov     dword ptr [rip + 0x21b67], 0
0x97e1: mov     eax, edx
0x97e3: and     eax, 0x7f
0x97e6: and     edx, 0x40
0x97e9: mov     byte ptr [rip + 0x21b61], al
0x97ef: jne     0x9810
0x97f1: mov     dword ptr [rip + 0x21b45], 1
0x97fb: and     eax, 0x7f
0x97fe: shl     ebx, 7
0x9801: or      eax, ebx
0x9803: mov     byte ptr [rip + 0x21b47], al
0x9809: pop     rbx
0x980a: ret
0x980b: nop     dword ptr [rax + rax]
0x9810: mov     dword ptr [rip + 0x21b2a], 0
0x981a: mov     edi, 1
0x981f: call    0x9870
0x9824: movzx   eax, byte ptr [rip + 0x21b25]
0x982b: jmp     0x97f1
0x97f1
0x982d: nop     dword ptr [rax]
0x9830: movzx   eax, byte ptr [rip + 0x21b19]
0x9837: shr     al, 7
0x983a: movzx   eax, al
0x983d: cmp     eax, edi
0x983f: je      0x9809
0x9841: cmp     ebx, 1
0x9844: jne     0x97a9
0x984a: mov     rdi, qword ptr [rip + 0x20517]
0x9851: test    rdi, rdi
0x9854: je      0x9824
0x9856: lea     rdx, [rip + 0x213]
0x985d: mov     esi, 1
0x9862: call    0x4820
0x9867: movzx   eax, byte ptr [rip + 0x21ae2]
0x986e: jmp     0x97f1
0x97f1
0x9870: cmp     dword ptr [rip + 0x21acd], 1

```

```

0x9877: push    rbx
0x9878: mov     ebx, edi
0x987a: je      0x98e0
0x987c: cmp     ebx, 1
0x987f: je      0x98f6
0x9881: mov     rdi, qword ptr [rip + 0x204c0]
0x9888: test    rdi, rdi
0x988b: je      0x989e
0x988d: lea     rdx, [rip + 0x1dc]
0x9894: mov     esi, 1
0x9899: call    0x4820
0x989e: movzx   eax, byte ptr [rip + 0x21aab]
0x98a5: mov     dword ptr [rip + 0x21a9d], 0
0x98af: mov     dword ptr [rip + 0x21a8f], 0
0x98b9: and     eax, 0xffffffffbf
0x98bc: mov     byte ptr [rip + 0x21a8e], al
0x98c2: test    al, al
0x98c4: js      0x9920
0x98c6: mov     dword ptr [rip + 0x21a74], 1
0x98d0: and     eax, 0xffffffffbf
0x98d3: shl     ebx, 6
0x98d6: or      eax, ebx
0x98d8: mov     byte ptr [rip + 0x21a72], al
0x98de: pop     rbx
0x98df: ret
0x98e0: movzx   eax, byte ptr [rip + 0x21a69]
0x98e7: sar     eax, 6
0x98ea: and     eax, 1
0x98ed: cmp     eax, edi
0x98ef: je      0x98de
0x98f1: cmp     ebx, 1
0x98f4: jne     0x9881
0x98f6: mov     rdi, qword ptr [rip + 0x2048b]
0x98fd: test    rdi, rdi
0x9900: je      0x9934
0x9902: lea     rdx, [rip + 0x167]
0x9909: mov     esi, 1
0x990e: call    0x4820
0x9913: movzx   eax, byte ptr [rip + 0x21a36]
0x991a: jmp     0x98c6
0x98c6:
0x991c: nop     dword ptr [rax]
0x9920: mov     dword ptr [rip + 0x21a16], 0
0x992a: mov     edi, 1
0x992f: call    0x9790
0x9934: movzx   eax, byte ptr [rip + 0x21a15]
0x993b: jmp     0x98c6
0x98c6:

```

```

0x993d: nop      dword ptr [rax]
0x9940: push     r12
0x9942: mov      edx, 5
0x9947: lea      rsi, [rip + 0x17293]
0x994e: mov      r12d, 1
0x9954: push     rbp
0x9955: mov      rbp, rdi
0x9958: sub      rsp, 8
0x995c: call     0x4890
0x9961: test     eax, eax
0x9963: je       0x99d3
0x9965: mov      edx, 4
0x996a: lea      rsi, [rip + 0x1725f]
0x9971: mov      rdi, rbp
0x9974: call     0x4890
0x9979: test     eax, eax
0x997b: je       0x99d3
0x997d: mov      edx, 9
0x9982: lea      rsi, [rip + 0x1724c]
0x9989: mov      rdi, rbp
0x998c: call     0x4890
0x9991: test     eax, eax
0x9993: je       0x99d3
0x9995: lea      rsi, [rip + 0x17243]
0x999c: mov      rdi, rbp
0x999f: call     0x4b30
0x99a4: test     eax, eax
0x99a6: je       0x99d3
0x99a8: lea      rsi, [rip + 0x17238]
0x99af: mov      rdi, rbp
0x99b2: call     0x4b30
0x99b7: test     eax, eax
0x99b9: je       0x99d3
0x99bb: lea      rsi, [rip + 0x1722b]
0x99c2: mov      rdi, rbp
0x99c5: xor      r12d, r12d
0x99c8: call     0x4b30
0x99cd: test     eax, eax
0x99cf: sete     r12b
0x99d3: add      rsp, 8
0x99d7: mov      eax, r12d
0x99da: pop      rbp
0x99db: pop      r12
0x99dd: ret
0x99de: nop
0x99e0: push     r13
0x99e2: push     r12
0x99e4: mov      r12, qword ptr [rip + 0x20f8d]

```

```

0x99eb: push    rbp
0x99ec: test     r12, r12
0x99ef: je       0x9a58
0x99f1: mov      r13, rdi
0x99f4: jmp      0x9a13
0x9a13:
0x99f6: nop      word ptr cs:[rax + rax]
0x9a00: js       0x9a30
0x9a02: mov      r12, qword ptr [r12 + 8]
0x9a07: mov      qword ptr [rip + 0x20f6a], r12
0x9a0e: test     r12, r12
0x9a11: je       0x9a58
0x9a13: mov      rbp, qword ptr [r12]
0x9a17: mov      rdi, r13
0x9a1a: mov      rsi, rbp
0x9a1d: call     0x4b30
0x9a22: test     eax, eax
0x9a24: jne      0x9a00
0x9a26: mov      rax, r12
0x9a29: pop      rbp
0x9a2a: pop      r12
0x9a2c: pop      r13
0x9a2e: ret
0x9a2f: nop
0x9a30: mov      rdi, r13
0x9a33: call     0x49b0
0x9a38: mov      rsi, rbp
0x9a3b: mov      rdi, r13
0x9a3e: mov      rdx, rax
0x9a41: call     0x4890
0x9a46: pop      rbp
0x9a47: neg      eax
0x9a49: sbb      r12, r12
0x9a4c: mov      rax, r12
0x9a4f: pop      r12
0x9a51: pop      r13
0x9a53: ret
0x9a54: nop      dword ptr [rax]
0x9a58: mov      r12, -1
0x9a5f: pop      rbp
0x9a60: mov      rax, r12
0x9a63: pop      r12
0x9a65: pop      r13
0x9a67: ret
0x9a68: nop      dword ptr [rax + rax]
0x9a70: endbr64
0x9a74: movsxd   rdx, dword ptr [rip + 0x2148d]
0x9a7b: push     r12

```

```

0x9a7d: push    rbp
0x9a7e: mov     ebp, edi
0x9a80: push    rbx
0x9a81: cmp     edx, 0x400
0x9a87: je      0x9ab0
0x9a89: lea     eax, [rdx + 1]
0x9a8c: lea     r12, [rip + 0x2148d]
0x9a93: mov     byte ptr [r12 + rdx], bpl
0x9a97: pop     rbx
0x9a98: mov     dword ptr [rip + 0x2146a], eax
0x9a9e: pop     rbp
0x9a9f: mov     eax, 1
0x9aa4: pop     r12
0x9aa6: ret
0x9aa7: nop     word ptr [rax + rax]
0x9ab0: xor     ebx, ebx
0x9ab2: lea     r12, [rip + 0x21467]
0x9ab9: jmp     0x9acc
0x9acc:
0x9abb: nop     dword ptr [rax + rax]
0x9ac0: mov     edx, dword ptr [rip + 0x21442]
0x9ac6: add     ebx, eax
0x9ac8: cmp     edx, ebx
0x9aca: jle     0x9ae5
0x9acc: sub     edx, ebx
0x9ace: movsxd  rsi, ebx
0x9ad1: mov     edi, 1
0x9ad6: add     rsi, r12
0x9ad9: movsxd  rdx, edx
0x9adc: call    0x18da0
0x9ae1: test    eax, eax
0x9ae3: jns     0x9ac0
0x9ae5: mov     eax, 1
0x9aea: xor     edx, edx
0x9aec: jmp     0x9a93
0x9a93:
0x9aee: nop
0x9af0: endbr64
0x9af4: mov     dword ptr [rip + 0x21862], edi
0x9afa: ret
0x9afb: nop     dword ptr [rax + rax]
0x9b00: endbr64
0x9b04: push    r12
0x9b06: push    rbp
0x9b07: push    rbx
0x9b08: mov     ebx, edi
0x9b0a: sub     rsp, 0x50
0x9b0e: mov     rax, qword ptr fs:[0x28]

```



```

0x9b17: mov     qword ptr [rsp + 0x48], rax
0x9b1c: xor     eax, eax
0x9b1e: cmp     edi, 1
0x9b21: je      0x9b60
0x9b23: lea     rdx, [rip + 0x21896]
0x9b2a: mov     esi, 1
0x9b2f: mov     edi, 1
0x9b34: call    0x4d70
0x9b39: mov     dword ptr [rip + 0x20e19], ebx
0x9b3f: mov     rax, qword ptr [rsp + 0x48]
0x9b44: sub     rax, qword ptr fs:[0x28]
0x9b4d: jne     0x9c7e
0x9b53: add     rsp, 0x50
0x9b57: pop     rbx
0x9b58: pop     rbp
0x9b59: pop     r12
0x9b5b: ret
0x9b5c: nop     dword ptr [rax]
0x9b60: mov     rax, qword ptr [rip + 0x21889]
0x9b67: mov     rdx, qword ptr [rip + 0x21852]
0x9b6e: lea     r12, [rip + 0x2180b]
0x9b75: mov     esi, 1
0x9b7a: movdqa  xmm0, xmmword ptr [rip + 0x2183e]
0x9b82: movdqa  xmm1, xmmword ptr [rip + 0x21846]
0x9b8a: mov     edi, 1
0x9b8f: mov     qword ptr [rip + 0x2181a], rax
0x9b96: mov     eax, dword ptr [rip + 0x2185c]
0x9b9c: movaps  xmmword ptr [rip + 0x217dd], xmm0
0x9ba3: movdqa  xmm2, xmmword ptr [rip + 0x21835]
0x9bab: mov     dword ptr [rip + 0x21807], eax
0x9bb1: movabs  rax, 0xfffffffffffffa3c
0x9bbb: and     rax, rdx
0x9bbe: mov     rdx, r12
0x9bc1: movaps  xmmword ptr [rip + 0x217c8], xmm1
0x9bc8: mov     qword ptr [rip + 0x217b1], rax
0x9bcf: mov     eax, dword ptr [rip + 0x217b7]
0x9bd5: movaps  xmmword ptr [rip + 0x217c4], xmm2
0x9bdc: and     eax, 0xffffffff5
0x9bdf: mov     byte ptr [rip + 0x217bb], 0
0x9be6: mov     dword ptr [rip + 0x217a1], 7
0x9bf0: or      al, 0x81
0x9bf2: mov     dword ptr [rip + 0x21794], eax
0x9bf8: mov     eax, 0x100
0x9bfd: mov     word ptr [rip + 0x21792], ax
0x9c04: mov     qword ptr [rip + 0x2178a], 0x1a0000
0x9c0f: call    0x4d70
0x9c14: mov     rdi, r12
0x9c17: mov     r12, rsp

```

```

0x9c1a: call    0x4d10
0x9c1f: mov     edx, 1
0x9c24: mov     esi, 0x540a
0x9c29: mov     edi, 1
0x9c2e: mov     dword ptr [rip + 0x20cfc], eax
0x9c34: xor     eax, eax
0x9c36: call    0x4a70
0x9c3b: mov     esi, 1
0x9c40: mov     edi, 1
0x9c45: call    0x4db0
0x9c4a: mov     ebp, dword ptr [rip + 0x200d4]
0x9c50: mov     rsi, r12
0x9c53: mov     edi, 1
0x9c58: call    0x4d60
0x9c5d: mov     rdx, r12
0x9c60: mov     esi, 1
0x9c65: mov     edi, 1
0x9c6a: mov     byte ptr [rsp + 0x11], bpl
0x9c6f: mov     byte ptr [rsp + 0x12], 0
0x9c74: call    0x4d70
0x9c79: jmp     0x9b39
0x9b39
0x9c7e: call    0x49d0
0x9c83: nop     word ptr cs:[rax + rax]
0x9c8e: nop
0x9c90: endbr64
0x9c94: mov     eax, dword ptr [rip + 0x20cbe]
0x9c9a: ret
0x9c9b: nop     dword ptr [rax + rax]
0x9ca0: endbr64
0x9ca4: push    r12
0x9ca6: push    rbx
0x9ca7: mov     ebx, edi
0x9ca9: mov     edi, 1
0x9cae: sub     rsp, 0x58
0x9cb2: mov     rax, qword ptr fs:[0x28]
0x9cbb: mov     qword ptr [rsp + 0x48], rax
0x9cc0: xor     eax, eax
0x9cc2: mov     r12, rsp
0x9cc5: mov     rsi, r12
0x9cc8: call    0x4d60
0x9ccd: mov     rdx, r12
0x9cd0: mov     esi, 1
0x9cd5: mov     edi, 1
0x9cda: mov     byte ptr [rsp + 0x11], bl
0x9cde: mov     byte ptr [rsp + 0x12], 0
0x9ce3: call    0x4d70
0x9ce8: mov     dword ptr [rip + 0x20036], ebx

```

```

0x9cee: mov     rax, qword ptr [rsp + 0x48]
0x9cf3: sub     rax, qword ptr fs:[0x28]
0x9cfc: jne     0x9d06
0x9cfe: add     rsp, 0x58
0x9d02: pop     rbx
0x9d03: pop     r12
0x9d05: ret
0x9d06: call    0x49d0
0x9d0b: nop     dword ptr [rax + rax]
0x9d10: endbr64
0x9d14: mov     edx, dword ptr [rip + 0x211ee]
0x9d1a: test    edx, edx
0x9d1c: jle     0x9d66
0x9d1e: push    rbp
0x9d1f: xor     ebp, ebp
0x9d21: push    rbx
0x9d22: lea     rbx, [rip + 0x211f7]
0x9d29: sub     rsp, 8
0x9d2d: jmp     0x9d3c
0x9d3c:
0x9d2f: nop
0x9d30: mov     edx, dword ptr [rip + 0x211d2]
0x9d36: add     ebp, eax
0x9d38: cmp     edx, ebp
0x9d3a: jle     0x9d55
0x9d3c: sub     edx, ebp
0x9d3e: movsxd  rsi, ebp
0x9d41: mov     edi, 1
0x9d46: add     rsi, rbx
0x9d49: movsxd  rdx, edx
0x9d4c: call    0x18da0
0x9d51: test    eax, eax
0x9d53: jns     0x9d30
0x9d55: mov     dword ptr [rip + 0x211a9], 0
0x9d5f: add     rsp, 8
0x9d63: pop     rbx
0x9d64: pop     rbp
0x9d65: ret
0x9d66: mov     dword ptr [rip + 0x21198], 0
0x9d70: ret
0x9d71: nop     word ptr cs:[rax + rax]
0x9d7c: nop     dword ptr [rax]
0x9d80: endbr64
0x9d84: push    rbp
0x9d85: lea     rdx, [rip - 0x31c]
0x9d8c: push    rbx
0x9d8d: sub     rsp, 8
0x9d91: mov     esi, dword ptr [rip + 0x20c05]

```

```

0x9d97: mov     rdi, qword ptr [rip + 0x2004a]
0x9d9e: call    0x4820
0x9da3: mov     edx, dword ptr [rip + 0x2115f]
0x9da9: test    edx, edx
0x9dab: jle     0x9de5
0x9dad: xor     ebp, ebp
0x9daf: lea     rbx, [rip + 0x2116a]
0x9db6: jmp     0x9dcc
0x9dcc:
0x9db8: nop     dword ptr [rax + rax]
0x9dc0: mov     edx, dword ptr [rip + 0x21142]
0x9dc6: add     ebp, eax
0x9dc8: cmp     edx, ebp
0x9dca: jle     0x9de5
0x9dcc: sub     edx, ebp
0x9dce: movsxd  rsi, ebp
0x9dd1: mov     edi, 1
0x9dd6: add     rsi, rbx
0x9dd9: movsxd  rdx, edx
0x9ddc: call    0x18da0
0x9de1: test    eax, eax
0x9de3: jns     0x9dc0
0x9de5: mov     dword ptr [rip + 0x21119], 0
0x9def: add     rsp, 8
0x9df3: pop     rbx
0x9df4: pop     rbp
0x9df5: ret
0x9df6: nop     word ptr cs:[rax + rax]
0x9e00: endbr64
0x9e04: push    rbp
0x9e05: lea     rdx, [rip - 0x39c]
0x9e0c: push    rbx
0x9e0d: sub     rsp, 8
0x9e11: mov     eax, dword ptr [rip + 0x20b85]
0x9e17: mov     rdi, qword ptr [rip + 0x2010a]
0x9e1e: lea     esi, [rax - 1]
0x9e21: call    0x4820
0x9e26: mov     edx, dword ptr [rip + 0x210dc]
0x9e2c: test    edx, edx
0x9e2e: jle     0x9e65
0x9e30: xor     ebp, ebp
0x9e32: lea     rbx, [rip + 0x210e7]
0x9e39: jmp     0x9e4c
0x9e4c:
0x9e3b: nop     dword ptr [rax + rax]
0x9e40: mov     edx, dword ptr [rip + 0x210c2]
0x9e46: add     ebp, eax
0x9e48: cmp     edx, ebp

```

```

0x9e4a: jle      0x9e65
0x9e4c: sub       edx, ebp
0x9e4e: movsxd    rsi, ebp
0x9e51: mov       edi, 1
0x9e56: add       rsi, rbx
0x9e59: movsxd    rdx, edx
0x9e5c: call      0x18da0
0x9e61: test      eax, eax
0x9e63: jns       0x9e40
0x9e65: mov       dword ptr [rip + 0x21099], 0
0x9e6f: add       rsp, 8
0x9e73: pop       rbx
0x9e74: pop       rbp
0x9e75: ret
0x9e76: nop       word ptr cs:[rax + rax]
0x9e80: endbr64
0x9e84: push      rbp
0x9e85: lea       rdx, [rip - 0x41c]
0x9e8c: push      rbx
0x9e8d: sub       rsp, 8
0x9e91: mov       eax, dword ptr [rip + 0x20b05]
0x9e97: mov       rdi, qword ptr [rip + 0x200aa]
0x9e9e: lea       esi, [rax - 1]
0x9ea1: call      0x4820
0x9ea6: mov       edx, dword ptr [rip + 0x2105c]
0x9eac: test      edx, edx
0x9eae: jle       0x9ee5
0x9eb0: xor       ebp, ebp
0x9eb2: lea       rbx, [rip + 0x21067]
0x9eb9: jmp       0x9ecc
0x9ecc
0x9ebb: nop       dword ptr [rax + rax]
0x9ec0: mov       edx, dword ptr [rip + 0x21042]
0x9ec6: add       ebp, eax
0x9ec8: cmp       edx, ebp
0x9eca: jle       0x9ee5
0x9ecc: sub       edx, ebp
0x9ece: movsxd    rsi, ebp
0x9ed1: mov       edi, 1
0x9ed6: add       rsi, rbx
0x9ed9: movsxd    rdx, edx
0x9edc: call      0x18da0
0x9ee1: test      eax, eax
0x9ee3: jns       0x9ec0
0x9ee5: mov       dword ptr [rip + 0x21019], 0
0x9eef: add       rsp, 8
0x9ef3: pop       rbx
0x9ef4: pop       rbp

```

```

0x9ef5: ret
0x9ef6: nop      word ptr cs:[rax + rax]
0x9f00: endbr64
0x9f04: mov      edx, dword ptr [rip + 0x214f2]
0x9f0a: movzx    eax, byte ptr [rdi]
0x9f0d: push     r14
0x9f0f: push     r13
0x9f11: mov      r13, rdi
0x9f14: push     r12
0x9f16: push     rbp
0x9f17: push     rbx
0x9f18: test     edx, edx
0x9f1a: jne      0x9fc8
0x9f20: cmp      al, 0x5e
0x9f22: jne      0x9fb8
0x9f28: lea      rbx, [r13 + 1]
0x9f2c: mov      rbp, r13
0x9f2f: lea      r14, [rip + 0x1fd6a]
0x9f36: jmp      0x9f56
0x9f56
0x9f38: nop      dword ptr [rax + rax]
0x9f40: add      rbp, 1
0x9f44: mov      byte ptr [rbx - 1], al
0x9f47: mov      rdx, rbx
0x9f4a: add      rbx, 1
0x9f4e: movzx    eax, byte ptr [rbp]
0x9f52: test     al, al
0x9f54: je       0x9fa6
0x9f56: cmp      al, 0x5e
0x9f58: jne      0x9f40
0x9f5a: movzx    r12d, byte ptr [rbp + 1]
0x9f5f: test     r12b, r12b
0x9f62: je       0x9fb8
0x9f64: call     0x47c0
0x9f69: mov      r8, rax
0x9f6c: movzx    eax, r12b
0x9f70: mov      rdx, qword ptr [r8]
0x9f73: mov      eax, dword ptr [rdx + rax*4]
0x9f76: cmp      eax, 0x47
0x9f79: je       0x9fb8
0x9f7b: cmp      eax, 0x5a
0x9f7e: je       0x9fb8
0x9f80: and      r12d, 0x7f
0x9f84: add      rbp, 2
0x9f88: mov      rdx, rbx
0x9f8b: add      rbx, 1
0x9f8f: sub      r12d, 0x20
0x9f93: movsxd   r12, r12d

```

```

0x9f96: movzx    eax, byte ptr [r14 + r12]
0x9f9b: mov         byte ptr [rbx - 2], al
0x9f9e: movzx    eax, byte ptr [rbp]
0x9fa2: test      al, al
0x9fa4: jne      0x9f56
0x9fa6: mov         byte ptr [rdx], 0
0x9fa9: mov         rax, r13
0x9fac: pop        rbx
0x9fad: pop        rbp
0x9fae: pop        r12
0x9fb0: pop        r13
0x9fb2: pop        r14
0x9fb4: ret
0x9fb5: nop        dword ptr [rax]
0x9fb8: pop        rbx
0x9fb9: xor        eax, eax
0x9fbb: pop        rbp
0x9fbc: pop        r12
0x9fbe: pop        r13
0x9fc0: pop        r14
0x9fc2: ret
0x9fc3: nop        dword ptr [rax + rax]
0x9fc8: test      al, al
0x9fca: jne      0x9f28
0x9fd0: mov         rdx, rdi
0x9fd3: jmp      0x9fa6
0x9fa6:
0x9fd5: nop        word ptr cs:[rax + rax]
0x9fe0: endbr64
0x9fe4: push       r15
0x9fe6: push       r14
0x9fe8: mov         r14, rdi
0x9feb: push       r13
0x9fed: lea        r13, [rip + 0x16c12]
0x9ff4: push       r12
0x9ff6: lea        r12, [rip + 0x209c3]
0x9ffd: push       rbp
0x9ffe: lea        rbp, [rip + 0x16bef]
0xa005: push       rbx
0xa006: mov         rbx, rdi
0xa009: sub         rsp, 0x18
0xa00d: mov         rax, qword ptr fs:[0x28]
0xa016: mov         qword ptr [rsp + 8], rax
0xa01b: xor        eax, eax
0xa01d: mov         byte ptr [rip + 0x2099c], 0
0xa024: movzx    r15d, byte ptr [rdi]
0xa028: test      r15b, r15b
0xa02b: jne      0xa0c2

```

```

0xa031: jmp      0xa0f0
0xa0f0:
0xa036: nop      word ptr cs:[rax + rax]
0xa040: cmp      r15b, 0x20
0xa044: je       0xa148
0xa04a: cmp      r15b, 0x7f
0xa04e: je       0xa160
0xa054: cmp      r15b, 0xff
0xa058: je       0xa180
0xa05e: call     0x4f00
0xa063: mov      r8, rax
0xa066: movzx    eax, r15b
0xa06a: mov      rsi, qword ptr [r8]
0xa06d: test     byte ptr [rsi + rax*2], 2
0xa071: je       0xa120
0xa077: lea      edx, [r15 + 0x40]
0xa07b: lea      rsi, [rsp + 5]
0xa080: mov      rdi, r12
0xa083: mov      byte ptr [rsp + 5], 0x5e
0xa088: mov      byte ptr [rsp + 6], dl
0xa08c: mov      edx, 0x12c
0xa091: mov      byte ptr [rsp + 7], 0
0xa096: call     0x47d0
0xa09b: add      r14, 1
0xa09f: cmp      byte ptr [r14], 0
0xa0a3: je       0xa0f0
0xa0a5: cmp      rbx, r14
0xa0a8: je       0xa0be
0xa0aa: mov      edx, 0x12c
0xa0af: lea      rsi, [rip + 0x1668f]
0xa0b6: mov      rdi, r12
0xa0b9: call     0x47d0
0xa0be: movzx    r15d, byte ptr [r14]
0xa0c2: cmp      r15b, 0x1b
0xa0c6: jne      0xa040
0xa0cc: mov      edx, 0x12c
0xa0d1: mov      rsi, rbp
0xa0d4: mov      rdi, r12
0xa0d7: add      r14, 1
0xa0db: call     0x47d0
0xa0e0: cmp      byte ptr [r14], 0
0xa0e4: jne      0xa0a5
0xa0e6: nop      word ptr cs:[rax + rax]
0xa0f0: mov      rax, qword ptr [rsp + 8]
0xa0f5: sub      rax, qword ptr fs:[0x28]
0xa0fe: jne      0xa199
0xa104: add      rsp, 0x18
0xa108: mov      rax, r12

```



```

0xa10b: pop    rbx
0xa10c: pop    rbp
0xa10d: pop    r12
0xa10f: pop    r13
0xa111: pop    r14
0xa113: pop    r15
0xa115: ret
0xa116: nop     word ptr cs:[rax + rax]
0xa120: lea     rsi, [rsp + 5]
0xa125: mov     edx, 0x12c
0xa12a: mov     rdi, r12
0xa12d: mov     byte ptr [rsp + 5], r15b
0xa132: mov     byte ptr [rsp + 6], 0
0xa137: call    0x47d0
0xa13c: jmp     0xa09b
0xa09b
0xa141: nop     dword ptr [rax]
0xa148: mov     edx, 0x12c
0xa14d: mov     rsi, r13
0xa150: mov     rdi, r12
0xa153: call    0x47d0
0xa158: jmp     0xa09b
0xa09b
0xa15d: nop     dword ptr [rax]
0xa160: mov     edx, 0x12c
0xa165: lea     rsi, [rip + 0x16a8f]
0xa16c: mov     rdi, r12
0xa16f: call    0x47d0
0xa174: jmp     0xa09b
0xa09b
0xa179: nop     dword ptr [rax]
0xa180: mov     edx, 0x12c
0xa185: lea     rsi, [rip + 0x16a79]
0xa18c: mov     rdi, r12
0xa18f: call    0x47d0
0xa194: jmp     0xa09b
0xa09b
0xa199: call    0x49d0
0xa19e: nop
0xa1a0: endbr64
0xa1a4: mov     eax, dword ptr [rip + 0x211ae]
0xa1aa: mov     r8, rdi
0xa1ad: mov     edi, dword ptr [rip + 0x207e5]
0xa1b3: push    r13
0xa1b5: push    r12
0xa1b7: xor     r12d, r12d
0xa1ba: lea     edx, [rax + rsi]
0xa1bd: push    rbx

```

```

0xa1be: mov     dword ptr [rip + 0x21194], edx
0xa1c4: cmp     edi, eax
0xa1c6: jle     0xa21c
0xa1c8: mov     ecx, dword ptr [rip + 0x21186]
0xa1ce: cmp     ecx, dword ptr [rip + 0x207c8]
0xa1d4: jge     0xa21c
0xa1d6: mov     r12d, edi
0xa1d9: sub     r12d, eax
0xa1dc: cmp     edx, edi
0xa1de: cmovle  r12d, esi
0xa1e2: imul    edi, ecx
0xa1e5: mov     rsi, r8
0xa1e8: movsxd  r13, r12d
0xa1eb: lea     ebx, [rdi + rax]
0xa1ee: mov     rdi, qword ptr [rip + 0x21143]
0xa1f5: mov     rdx, r13
0xa1f8: movsxd  rbx, ebx
0xa1fb: add     rdi, rbx
0xa1fe: call    0x4bb0
0xa203: add     rbx, qword ptr [rip + 0x21126]
0xa20a: movzx   esi, byte ptr [rip + 0x21140]
0xa211: mov     rdx, r13
0xa214: mov     rdi, rbx
0xa217: call    0x4a50
0xa21c: mov     eax, r12d
0xa21f: pop     rbx
0xa220: pop     r12
0xa222: pop     r13
0xa224: ret
0xa225: nop     word ptr cs:[rax + rax]
0xa230: endbr64
0xa234: push    r13
0xa236: push    r12
0xa238: xor     r12d, r12d
0xa23b: push    rbx
0xa23c: sub     rsp, 0x10
0xa240: mov     rax, qword ptr fs:[0x28]
0xa249: mov     qword ptr [rsp + 8], rax
0xa24e: xor     eax, eax
0xa250: mov     eax, dword ptr [rip + 0x21102]
0xa256: mov     byte ptr [rsp + 7], dil
0xa25b: mov     edi, dword ptr [rip + 0x20737]
0xa261: lea     edx, [rax + 1]
0xa264: mov     dword ptr [rip + 0x210ee], edx
0xa26a: cmp     eax, edi
0xa26c: jge     0xa2c2
0xa26e: mov     ecx, dword ptr [rip + 0x210e0]
0xa274: cmp     ecx, dword ptr [rip + 0x20722]

```

```

0xa27a: jge      0xa2c2
0xa27c: cmp      edx, edi
0xa27e: jle      0xa2e0
0xa280: mov      r12d, edi
0xa283: sub      r12d, eax
0xa286: movsxd   r13, r12d
0xa289: imul     edi, ecx
0xa28c: lea      rsi, [rsp + 7]
0xa291: mov      rdx, r13
0xa294: lea      ebx, [rdi + rax]
0xa297: mov      rdi, qword ptr [rip + 0x2109a]
0xa29e: movsxd   rbx, ebx
0xa2a1: add      rdi, rbx
0xa2a4: call     0x4bb0
0xa2a9: add      rbx, qword ptr [rip + 0x21080]
0xa2b0: movzx    esi, byte ptr [rip + 0x2109a]
0xa2b7: mov      rdx, r13
0xa2ba: mov      rdi, rbx
0xa2bd: call     0x4a50
0xa2c2: mov      rax, qword ptr [rsp + 8]
0xa2c7: sub      rax, qword ptr fs:[0x28]
0xa2d0: jne      0xa2ee
0xa2d2: add      rsp, 0x10
0xa2d6: mov      eax, r12d
0xa2d9: pop      rbx
0xa2da: pop      r12
0xa2dc: pop      r13
0xa2de: ret
0xa2df: nop
0xa2e0: mov      r13d, 1
0xa2e6: mov      r12d, 1
0xa2ec: jmp      0xa289
0xa289:
0xa2ee: call     0x49d0
0xa2f3: nop      word ptr cs:[rax + rax]
0xa2fe: nop
0xa300: endbr64
0xa304: push     rbp
0xa305: lea      rdx, [rip - 0x89c]
0xa30c: push     rbx
0xa30d: sub      rsp, 8
0xa311: mov      esi, dword ptr [rip + 0x20685]
0xa317: mov      rdi, qword ptr [rip + 0x1faca]
0xa31e: call     0x4820
0xa323: mov      edx, dword ptr [rip + 0x20bdf]
0xa329: test     edx, edx
0xa32b: jle      0xa365
0xa32d: xor      ebp, ebp

```

```

0xa32f: lea      rbx, [rip + 0x20bea]
0xa336: jmp      0xa34c
0xa34c:
0xa338: nop      dword ptr [rax + rax]
0xa340: mov      edx, dword ptr [rip + 0x20bc2]
0xa346: add      ebp, eax
0xa348: cmp      edx, ebp
0xa34a: jle      0xa365
0xa34c: sub      edx, ebp
0xa34e: movsxd   rsi, ebp
0xa351: mov      edi, 1
0xa356: add      rsi, rbx
0xa359: movsxd   rdx, edx
0xa35c: call     0x18da0
0xa361: test     eax, eax
0xa363: jns      0xa340
0xa365: mov      edx, dword ptr [rip + 0x20631]
0xa36b: xor      esi, esi
0xa36d: imul     edx, dword ptr [rip + 0x20624]
0xa374: mov      dword ptr [rip + 0x20b8a], 0
0xa37e: mov      rdi, qword ptr [rip + 0x20fb3]
0xa385: movsxd   rdx, edx
0xa388: call     0x4a50
0xa38d: mov      edx, dword ptr [rip + 0x20609]
0xa393: imul     edx, dword ptr [rip + 0x205fe]
0xa39a: xor      esi, esi
0xa39c: mov      rdi, qword ptr [rip + 0x20f8d]
0xa3a3: movsxd   rdx, edx
0xa3a6: call     0x4a50
0xa3ab: mov      edx, dword ptr [rip + 0x205eb]
0xa3b1: imul     edx, dword ptr [rip + 0x205e0]
0xa3b8: xor      esi, esi
0xa3ba: mov      rdi, qword ptr [rip + 0x20f67]
0xa3c1: movsxd   rdx, edx
0xa3c4: call     0x4a50
0xa3c9: mov      edx, dword ptr [rip + 0x205cd]
0xa3cf: imul     edx, dword ptr [rip + 0x205c2]
0xa3d6: xor      esi, esi
0xa3d8: mov      rdi, qword ptr [rip + 0x20f41]
0xa3df: movsxd   rdx, edx
0xa3e2: call     0x4a50
0xa3e7: mov      dword ptr [rip + 0x20f67], 0
0xa3f1: mov      dword ptr [rip + 0x20f59], 0
0xa3fb: add      rsp, 8
0xa3ff: pop      rbx
0xa400: pop      rbp
0xa401: ret
0xa402: nop      word ptr cs:[rax + rax]

```

```

0xa40d: nop      dword ptr [rax]
0xa410: endbr64
0xa414: sub      rsp, 8
0xa418: mov      edx, dword ptr [rip + 0x2057e]
0xa41e: imul     edx, dword ptr [rip + 0x20573]
0xa425: mov      esi, 0x20
0xa42a: mov      rdi, qword ptr [rip + 0x20f07]
0xa431: movsxd   rdx, edx
0xa434: call     0x4a50
0xa439: mov      edx, dword ptr [rip + 0x2055d]
0xa43f: imul     edx, dword ptr [rip + 0x20552]
0xa446: movzx    esi, byte ptr [rip + 0x20f04]
0xa44d: mov      rdi, qword ptr [rip + 0x20edc]
0xa454: movsxd   rdx, edx
0xa457: call     0x4a50
0xa45c: mov      edx, dword ptr [rip + 0x2053a]
0xa462: imul     edx, dword ptr [rip + 0x2052f]
0xa469: xor      esi, esi
0xa46b: mov      rdi, qword ptr [rip + 0x20eb6]
0xa472: add      rsp, 8
0xa476: movsxd   rdx, edx
0xa479: jmp      0x4a50
0x4a50
0xa47e: nop
0xa480: endbr64
0xa484: mov      edx, dword ptr [rip + 0x20512]
0xa48a: imul     edx, dword ptr [rip + 0x20507]
0xa491: xor      esi, esi
0xa493: mov      rdi, qword ptr [rip + 0x20e8e]
0xa49a: movsxd   rdx, edx
0xa49d: jmp      0x4a50
0x4a50
0xa4a2: nop      word ptr cs:[rax + rax]
0xa4ad: nop      dword ptr [rax]
0xa4b0: endbr64
0xa4b4: mov      dword ptr [rip + 0x20e9a], edi
0xa4ba: mov      dword ptr [rip + 0x20e98], esi
0xa4c0: ret
0xa4c1: nop      word ptr cs:[rax + rax]
0xa4cc: nop      dword ptr [rax]
0xa4d0: endbr64
0xa4d4: mov      eax, dword ptr [rip + 0x20e7a]
0xa4da: mov      dword ptr [rdi], eax
0xa4dc: mov      eax, dword ptr [rip + 0x20e76]
0xa4e2: mov      dword ptr [rsi], eax
0xa4e4: ret
0xa4e5: nop      word ptr cs:[rax + rax]
0xa4f0: endbr64

```

```

0xa4f4: movzx    eax, byte ptr [rip + 0x20e56]
0xa4fb: and      edi, 7
0xa4fe: and      eax, 0xffffffff8
0xa501: or       eax, edi
0xa503: mov      byte ptr [rip + 0x20e48], al
0xa509: ret
0xa50a: nop      word ptr [rax + rax]
0xa510: endbr64
0xa514: movzx    eax, byte ptr [rip + 0x20e36]
0xa51b: shl      edi, 3
0xa51e: and      edi, 0x38
0xa521: and      eax, 0xffffffffc7
0xa524: or       eax, edi
0xa526: mov      byte ptr [rip + 0x20e25], al
0xa52c: ret
0xa52d: nop      dword ptr [rax]
0xa530: endbr64
0xa534: movzx    eax, byte ptr [rip + 0x20e16]
0xa53b: shl      edi, 6
0xa53e: and      edi, 0x40
0xa541: and      eax, 0xffffffffbf
0xa544: or       eax, edi
0xa546: mov      byte ptr [rip + 0x20e05], al
0xa54c: ret
0xa54d: nop      dword ptr [rax]
0xa550: endbr64
0xa554: movzx    eax, byte ptr [rip + 0x20df6]
0xa55b: shl      edi, 7
0xa55e: and      eax, 0x7f
0xa561: or       eax, edi
0xa563: mov      byte ptr [rip + 0x20de8], al
0xa569: ret
0xa56a: nop      word ptr [rax + rax]
0xa570: endbr64
0xa574: shl      edi, 6
0xa577: and      esi, 7
0xa57a: shl      edx, 3
0xa57d: mov      eax, edi
0xa57f: movzx    edi, byte ptr [rip + 0x20dcb]
0xa586: and      edx, 0x38
0xa589: and      eax, 0x40
0xa58c: and      edi, 0xffffffffb8
0xa58f: or       edi, eax
0xa591: or       edi, esi
0xa593: and      edi, 0xffffffffc7
0xa596: or       edi, edx
0xa598: mov      byte ptr [rip + 0x20db2], dil
0xa59f: ret

```

```

0xa5a0: endbr64
0xa5a4: push    rbp
0xa5a5: push    rbx
0xa5a6: sub     rsp, 8
0xa5aa: movsxd  rax, dword ptr [rip + 0x20957]
0xa5b1: cmp     eax, 0x400
0xa5b6: je      0xa610
0xa5b8: lea     edx, [rax + 1]
0xa5bb: lea     rbx, [rip + 0x2095e]
0xa5c2: mov     dword ptr [rip + 0x20940], edx
0xa5c8: mov     byte ptr [rbx + rax], 7
0xa5cc: test    edx, edx
0xa5ce: jle     0xa5fd
0xa5d0: xor     ebp, ebp
0xa5d2: jmp     0xa5e4
0xa5e4:
0xa5d4: nop     dword ptr [rax]
0xa5d8: mov     edx, dword ptr [rip + 0x2092a]
0xa5de: add     ebp, eax
0xa5e0: cmp     edx, ebp
0xa5e2: jle     0xa5fd
0xa5e4: sub     edx, ebp
0xa5e6: movsxd  rsi, ebp
0xa5e9: mov     edi, 1
0xa5ee: add     rsi, rbx
0xa5f1: movsxd  rdx, edx
0xa5f4: call    0x18da0
0xa5f9: test    eax, eax
0xa5fb: jns     0xa5d8
0xa5fd: mov     dword ptr [rip + 0x20901], 0
0xa607: add     rsp, 8
0xa60b: pop     rbx
0xa60c: pop     rbp
0xa60d: ret
0xa60e: nop
0xa610: xor     ebp, ebp
0xa612: lea     rbx, [rip + 0x20907]
0xa619: jmp     0xa62c
0xa62c:
0xa61b: nop     dword ptr [rax + rax]
0xa620: add     ebp, eax
0xa622: mov     eax, dword ptr [rip + 0x208e0]
0xa628: cmp     eax, ebp
0xa62a: jle     0xa645
0xa62c: sub     eax, ebp
0xa62e: movsxd  rsi, ebp
0xa631: mov     edi, 1
0xa636: add     rsi, rbx

```

```

0xa639: movsxd  rdx, eax
0xa63c: call     0x18da0
0xa641: test     eax, eax
0xa643: jns      0xa620
0xa645: mov      dword ptr [rip + 0x208b9], 1
0xa64f: mov      edx, 1
0xa654: mov      byte ptr [rip + 0x208c5], 7
0xa65b: jmp      0xa5d0
0xa5d0
0xa660: endbr64
0xa664: test     edi, edi
0xa666: je       0xa688
0xa668: mov      rdi, qword ptr [rip + 0x1f759]
0xa66f: test     rdi, rdi
0xa672: je       0xa694
0xa674: lea      rdx, [rip - 0xc0b]
0xa67b: mov      esi, 1
0xa680: jmp      0x4820
0x4820
0xa685: nop      dword ptr [rax]
0xa688: mov      rdi, qword ptr [rip + 0x1f719]
0xa68f: test     rdi, rdi
0xa692: jne      0xa674
0xa694: ret
0xa695: nop      word ptr cs:[rax + rax]
0xa6a0: endbr64
0xa6a4: movzx    eax, byte ptr [rip + 0x20ca6]
0xa6ab: mov      byte ptr [rdi], al
0xa6ad: ret
0xa6ae: nop
0xa6b0: endbr64
0xa6b4: movzx    eax, byte ptr [rdi]
0xa6b7: mov      byte ptr [rip + 0x20c94], al
0xa6bd: ret
0xa6be: nop
0xa6c0: endbr64
0xa6c4: push     rbp
0xa6c5: push     rbx
0xa6c6: sub      rsp, 8
0xa6ca: cmp      dword ptr [rip + 0x1f58f], 1
0xa6d1: je       0xa720
0xa6d3: mov      rdi, qword ptr [rip + 0x1f66e]
0xa6da: test     rdi, rdi
0xa6dd: je       0xa6f0
0xa6df: lea      rdx, [rip - 0xc76]
0xa6e6: mov      esi, 1
0xa6eb: call     0x4820
0xa6f0: mov      dword ptr [rip + 0x20c52], 0

```



```

0xa6fa: mov     dword ptr [rip + 0x20c44], 0
0xa704: mov     dword ptr [rip + 0x20c36], 0
0xa70e: mov     dword ptr [rip + 0x20c28], 0
0xa718: add     rsp, 8
0xa71c: pop     rbx
0xa71d: pop     rbp
0xa71e: ret
0xa71f: nop
0xa720: mov     edx, dword ptr [rip + 0x207e2]
0xa726: cmp     edx, 0x3fb
0xa72c: jg      0xa750
0xa72e: movsxd  rax, edx
0xa731: lea     rcx, [rip + 0x207e8]
0xa738: add     edx, 4
0xa73b: add     rax, rcx
0xa73e: mov     ecx, dword ptr [rip + 0x1f5db]
0xa744: mov     dword ptr [rip + 0x207be], edx
0xa74a: mov     dword ptr [rax], ecx
0xa74c: jmp     0xa6d3
0xa6d3:
0xa74e: nop
0xa750: xor     ebx, ebx
0xa752: lea     rbp, [rip + 0x207c7]
0xa759: jmp     0xa76c
0xa76c:
0xa75b: nop     dword ptr [rax + rax]
0xa760: mov     edx, dword ptr [rip + 0x207a2]
0xa766: add     ebx, eax
0xa768: cmp     edx, ebx
0xa76a: jle     0xa785
0xa76c: sub     edx, ebx
0xa76e: movsxd  rsi, ebx
0xa771: mov     edi, 1
0xa776: add     rsi, rbp
0xa779: movsxd  rdx, edx
0xa77c: call    0x18da0
0xa781: test    eax, eax
0xa783: jns     0xa760
0xa785: mov     edx, 4
0xa78a: lea     rax, [rip + 0x2078f]
0xa791: jmp     0xa73e
0xa73e:
0xa793: nop     word ptr cs:[rax + rax]
0xa79e: nop
0xa7a0: push    r12
0xa7a2: mov     r12d, edi
0xa7a5: push    rbp
0xa7a6: mov     ebp, esi

```

```

0xa7a8: sub     rsp, 8
0xa7ac: mov     eax, dword ptr [rip + 0x1f71e]
0xa7b2: test    eax, eax
0xa7b4: je      0xa800
0xa7b6: mov     eax, ebp
0xa7b8: mov     esi, dword ptr [rip + 0x201da]
0xa7be: mov     edx, dword ptr [rip + 0x201d8]
0xa7c4: or      eax, r12d
0xa7c7: mov     rdi, qword ptr [rip + 0x1f63a]
0xa7ce: js      0xa7d4
0xa7d0: cmp     ebp, esi
0xa7d2: jl      0xa810
0xa7d4: sub     edx, 1
0xa7d7: sub     esi, 1
0xa7da: call    0x4960
0xa7df: add     rsp, 8
0xa7e3: mov     esi, 1
0xa7e8: lea     rdx, [rip - 0xd7f]
0xa7ef: pop     rbp
0xa7f0: mov     rdi, rax
0xa7f3: pop     r12
0xa7f5: jmp     0x4820
0x4820:
0xa7fa: nop     word ptr [rax + rax]
0xa800: xor     eax, eax
0xa802: call    0xa6c0
0xa807: jmp     0xa7b6
0xa7b6:
0xa809: nop     dword ptr [rax]
0xa810: cmp     r12d, edx
0xa813: jge     0xa7d4
0xa815: mov     edx, r12d
0xa818: mov     esi, ebp
0xa81a: jmp     0xa7da
0xa7da:
0xa81c: nop     dword ptr [rax]
0xa820: endbr64
0xa824: push    r15
0xa826: push    r14
0xa828: push    r13
0xa82a: push    r12
0xa82c: push    rbp
0xa82d: push    rbx
0xa82e: sub     rsp, 0x38
0xa832: mov     ebx, dword ptr [rip + 0x20160]
0xa838: imul    ebx, dword ptr [rip + 0x2015d]
0xa83f: mov     rax, qword ptr fs:[0x28]
0xa848: mov     qword ptr [rsp + 0x28], rax

```

```

0xa84d: xor     eax, eax
0xa84f: mov     rdi, qword ptr [rip + 0x1f552]
0xa856: cmp     dword ptr [rip + 0x20aff], 1
0xa85d: sbb     ebx, 0
0xa860: test    rdi, rdi
0xa863: je      0xa876
0xa865: lea     rdx, [rip - 0xdfc]
0xa86c: mov     esi, 1
0xa871: call    0x4820
0xa876: test    ebx, ebx
0xa878: jle     0xab70
0xa87e: movsxd  rax, ebx
0xa881: mov     dword ptr [rsp + 4], 0xffffffff
0xa889: xor     ebp, ebp
0xa88b: mov     r13d, 0xffffffff
0xa891: mov     qword ptr [rsp + 8], rax
0xa896: lea     r12, [rip + 0x20683]
0xa89d: jmp     0xa9ce
0xa9ce:
0xa8a2: nop     word ptr [rax + rax]
0xa8a8: cdq
0xa8a9: idiv    dword ptr [rip + 0x200e9]
0xa8af: mov     r14d, edx
0xa8b2: cmp     edx, r13d
0xa8b5: jne     0xaa20
0xa8bb: cmp     eax, dword ptr [rsp + 4]
0xa8bf: jne     0xaa20
0xa8c5: mov     rax, qword ptr [rip + 0x20a64]
0xa8cc: movzx   ebx, byte ptr [rax + rbp]
0xa8d0: mov     edi, ebx
0xa8d2: mov     r14d, ebx
0xa8d5: sar     edi, 6
0xa8d8: and     r14d, 7
0xa8dc: and     edi, 1
0xa8df: call    0x9870
0xa8e4: mov     eax, ebx
0xa8e6: and     eax, 7
0xa8e9: cmp     dword ptr [rip + 0x20a5c], 1
0xa8f0: je      0xaa40
0xa8f6: cmp     dword ptr [rip + 0x1f363], 1
0xa8fd: je      0xaa5f
0xa903: xor     edi, edi
0xa905: cmp     eax, 7
0xa908: setne   dil
0xa90c: call    0x9790
0xa911: movzx   eax, byte ptr [rip + 0x20a38]
0xa918: mov     dword ptr [rip + 0x20a2a], 1
0xa922: and     eax, 0xffffffff8

```

```

0xa925: or      eax, r14d
0xa928: mov     byte ptr [rip + 0x20a22], al
0xa92e: sar     ebx, 3
0xa931: and     ebx, 7
0xa934: cmp     dword ptr [rip + 0x20a0d], 1
0xa93b: je      0xaab0
0xa941: cmp     dword ptr [rip + 0x1f318], 1
0xa948: je      0xaad8
0xa94e: xor     edi, edi
0xa950: test    ebx, ebx
0xa952: setne   dil
0xa956: call    0x9790
0xa95b: mov     edx, dword ptr [rip + 0x205a7]
0xa961: movzx   eax, byte ptr [rip + 0x209e8]
0xa968: shl     ebx, 3
0xa96b: mov     dword ptr [rip + 0x209d3], 1
0xa975: and     eax, 0xffffffffc7
0xa978: or      ebx, eax
0xa97a: mov     byte ptr [rip + 0x209d0], bl
0xa980: mov     rax, qword ptr [rip + 0x209b1]
0xa987: movzx   ebx, byte ptr [rax + rbp]
0xa98b: cmp     edx, 0x400
0xa991: je      0xab30
0xa997: lea     eax, [rdx + 1]
0xa99a: movsxd  rdx, edx
0xa99d: add     r13d, 1
0xa9a1: cmp     dword ptr [rip + 0x1fff0], r13d
0xa9a8: mov     dword ptr [rip + 0x2055a], eax
0xa9ae: mov     byte ptr [r12 + rdx], bl
0xa9b2: jne     0xa9bf
0xa9b4: mov     dword ptr [rsp + 4], 0xffffffff
0xa9bc: xor     r13d, r13d
0xa9bf: add     rbp, 1
0xa9c3: cmp     rbp, qword ptr [rsp + 8]
0xa9c8: je      0xab70
0xa9ce: mov     rdx, qword ptr [rip + 0x20953]
0xa9d5: mov     rcx, qword ptr [rip + 0x2095c]
0xa9dc: mov     eax, ebp
0xa9de: movzx   ebx, byte ptr [rdx + rbp]
0xa9e2: cmp     byte ptr [rcx + rbp], bl
0xa9e5: jne     0xa8a8
0xa9eb: mov     rdx, qword ptr [rip + 0x2092e]
0xa9f2: mov     rcx, qword ptr [rip + 0x20937]
0xa9f9: movzx   edi, byte ptr [rdx + rbp]
0xa9fd: cmp     byte ptr [rcx + rbp], dil
0xaa01: je      0xa9bf
0xaa03: cdq
0xaa04: idiv    dword ptr [rip + 0x1ff8e]

```

```

0xaa0a: mov     r14d, edx
0xaa0d: cmp     edx, r13d
0xaa10: je      0xa8bb
0xaa16: nop     word ptr cs:[rax + rax]
0xaa20: mov     esi, r14d
0xaa23: mov     edi, eax
0xaa25: mov     dword ptr [rsp + 4], eax
0xaa29: mov     r13d, r14d
0xaa2c: call    0xa7a0
0xaa31: jmp     0xa8c5
0xa8c5:
0xaa36: nop     word ptr cs:[rax + rax]
0xaa40: movzx   edx, byte ptr [rip + 0x20909]
0xaa47: and     edx, 7
0xaa4a: cmp     eax, edx
0xaa4c: je      0xa92e
0xaa52: cmp     dword ptr [rip + 0x1f207], 1
0xaa59: jne     0xa903
0xaa5f: mov     eax, dword ptr [rip + 0x164b4]
0xaa65: mov     edx, dword ptr [rip + 0x2049d]
0xaa6b: mov     dword ptr [rsp + 0x10], eax
0xaa6f: movzx   eax, byte ptr [rip + 0x164a7]
0xaa76: mov     byte ptr [rsp + 0x14], al
0xaa7a: lea     eax, [r14 + 0x30]
0xaa7e: mov     byte ptr [rsp + 0x13], al
0xaa82: cmp     edx, 0x3fa
0xaa88: jg      0xac18
0xaa8e: movsxd  rax, edx
0xaa91: add     edx, 5
0xaa94: add     rax, r12
0xaa97: mov     ecx, dword ptr [rsp + 0x10]
0xaa9b: mov     dword ptr [rip + 0x20467], edx
0xaaa1: mov     dword ptr [rax], ecx
0xaaa3: movzx   ecx, byte ptr [rsp + 0x14]
0xaaa8: mov     byte ptr [rax + 4], cl
0xaaaab: jmp     0xa911
0xa911:
0xaab0: movzx   eax, byte ptr [rip + 0x20899]
0xaab7: mov     edx, dword ptr [rip + 0x2044b]
0xaabd: sar     eax, 3
0xaac0: and     eax, 7
0xaac3: cmp     ebx, eax
0xaac5: je      0xa980
0xaacb: cmp     dword ptr [rip + 0x1f18e], 1
0xaad2: jne     0xa94e
0xaad8: mov     eax, dword ptr [rip + 0x16436]
0xaade: mov     edx, dword ptr [rip + 0x20424]
0xaae4: mov     dword ptr [rsp + 0x10], eax

```

```

0xaae8: movzx    eax, byte ptr [rip + 0x16429]
0xaaef: mov      byte ptr [rsp + 0x14], al
0xaaf3: lea     eax, [rbx + 0x30]
0xaaf6: mov     byte ptr [rsp + 0x13], al
0xaafa: cmp     edx, 0x3fa
0xab00: jg      0xac58
0xab06: movsxd  rax, edx
0xab09: add     edx, 5
0xab0c: add     rax, r12
0xab0f: mov     ecx, dword ptr [rsp + 0x10]
0xab13: mov     dword ptr [rip + 0x203ef], edx
0xab19: mov     dword ptr [rax], ecx
0xab1b: movzx   ecx, byte ptr [rsp + 0x14]
0xab20: mov     byte ptr [rax + 4], cl
0xab23: jmp     0xa961
0xa961
0xab28: nop     dword ptr [rax + rax]
0xab30: xor     r14d, r14d
0xab33: jmp     0xab46
0xab46
0xab35: nop     dword ptr [rax]
0xab38: mov     edx, dword ptr [rip + 0x203ca]
0xab3e: add     r14d, eax
0xab41: cmp     edx, r14d
0xab44: jle     0xab60
0xab46: sub     edx, r14d
0xab49: movsxd  rsi, r14d
0xab4c: mov     edi, 1
0xab51: add     rsi, r12
0xab54: movsxd  rdx, edx
0xab57: call    0x18da0
0xab5c: test    eax, eax
0xab5e: jns     0xab38
0xab60: mov     eax, 1
0xab65: xor     edx, edx
0xab67: jmp     0xa99a
0xa99a
0xab6c: nop     dword ptr [rax]
0xab70: mov     edi, dword ptr [rip + 0x207de]
0xab76: mov     esi, dword ptr [rip + 0x207dc]
0xab7c: call    0xa7a0
0xab81: mov     rdi, qword ptr [rip + 0x1f240]
0xab88: test    rdi, rdi
0xab8b: je      0xab9e
0xab8d: lea     rdx, [rip - 0x1124]
0xab94: mov     esi, 1
0xab99: call    0x4820
0xab9e: mov     edx, dword ptr [rip + 0x20364]

```

```

0xaba4: test    edx, edx
0xaba6: jne      0xac98
0xabac: mov     edx, dword ptr [rip + 0x1fde6]
0xabb2: imul    edx, dword ptr [rip + 0x1fde3]
0xabb9: mov     rsi, qword ptr [rip + 0x20778]
0xabc0: mov     rdi, qword ptr [rip + 0x20761]
0xabc7: movsxd  rdx, edx
0xabca: call    0x4bb0
0xabcf: mov     edx, dword ptr [rip + 0x1fdc3]
0xabd5: imul    edx, dword ptr [rip + 0x1fdc0]
0xabdc: mov     rdi, qword ptr [rip + 0x2073d]
0xabe3: mov     rsi, qword ptr [rip + 0x20746]
0xabea: movsxd  rdx, edx
0xabed: mov     rax, qword ptr [rsp + 0x28]
0xabf2: sub     rax, qword ptr fs:[0x28]
0xabfb: jne     0xace4
0xac01: add     rsp, 0x38
0xac05: pop     rbx
0xac06: pop     rbp
0xac07: pop     r12
0xac09: pop     r13
0xac0b: pop     r14
0xac0d: pop     r15
0xac0f: jmp     0x4bb0
0x4bb0
0xac14: nop     dword ptr [rax]
0xac18: xor     r15d, r15d
0xac1b: jmp     0xac2e
0xac2e
0xac1d: nop     dword ptr [rax]
0xac20: mov     edx, dword ptr [rip + 0x202e2]
0xac26: add     r15d, eax
0xac29: cmp     edx, r15d
0xac2c: jle     0xac48
0xac2e: sub     edx, r15d
0xac31: movsxd  rsi, r15d
0xac34: mov     edi, 1
0xac39: add     rsi, r12
0xac3c: movsxd  rdx, edx
0xac3f: call    0x18da0
0xac44: test    eax, eax
0xac46: jns     0xac20
0xac48: mov     edx, 5
0xac4d: mov     rax, r12
0xac50: jmp     0xaa97
0xaa97
0xac55: nop     dword ptr [rax]
0xac58: xor     r14d, r14d

```

```

0xac5b: jmp      0xac6e
0xac6e
0xac5d: nop      dword ptr [rax]
0xac60: mov      edx, dword ptr [rip + 0x202a2]
0xac66: add      r14d, eax
0xac69: cmp      edx, r14d
0xac6c: jle      0xac88
0xac6e: sub      edx, r14d
0xac71: movsxd   rsi, r14d
0xac74: mov      edi, 1
0xac79: add      rsi, r12
0xac7c: movsxd   rdx, edx
0xac7f: call     0x18da0
0xac84: test     eax, eax
0xac86: jns      0xac60
0xac88: mov      edx, 5
0xac8d: mov      rax, r12
0xac90: jmp      0xab0f
0xab0f
0xac95: nop      dword ptr [rax]
0xac98: mov      ebx, 0
0xac9d: lea      rbp, [rip + 0x2027c]
0xaca4: jg       0xacbc
0xaca6: jmp      0xacd5
0xacd5
0xaca8: nop      dword ptr [rax + rax]
0xacb0: mov      edx, dword ptr [rip + 0x20252]
0xacb6: add      ebx, eax
0xacb8: cmp      edx, ebx
0xacba: jle      0xacd5
0xacbc: sub      edx, ebx
0xacbe: movsxd   rsi, ebx
0xacc1: mov      edi, 1
0xacc6: add      rsi, rbp
0xacc9: movsxd   rdx, edx
0xaccc: call     0x18da0
0xacd1: test     eax, eax
0xacd3: jns      0xacb0
0xacd5: mov      dword ptr [rip + 0x20229], 0
0xacdf: jmp      0xabac
0xabac
0xace4: call     0x49d0
0xace9: nop      dword ptr [rax]
0xacf0: endbr64
0xacf4: push     r12
0xacf6: xor      eax, eax
0xacf8: mov      r12, rdi
0xacfb: push     rbx

```



```

0xacfc: mov     ebx, esi
0xacfe: sub     rsp, 8
0xad02: call    0xa820
0xad07: mov     rdx, qword ptr [rip + 0x1fc1a]
0xad0e: test    rdx, rdx
0xad11: je      0xad17
0xad13: xor     eax, eax
0xad15: call    rdx
0xad17: movsxd  rdx, ebx
0xad1a: mov     rsi, r12
0xad1d: xor     edi, edi
0xad1f: call    0x18d30
0xad24: mov     rdx, qword ptr [rip + 0x1fbf5]
0xad2b: mov     r12d, eax
0xad2e: test    rdx, rdx
0xad31: je      0xad37
0xad33: xor     eax, eax
0xad35: call    rdx
0xad37: add     rsp, 8
0xad3b: mov     eax, r12d
0xad3e: pop     rbx
0xad3f: pop     r12
0xad41: ret
0xad42: nop     word ptr cs:[rax + rax]
0xad4d: nop     dword ptr [rax]
0xad50: push    r12
0xad52: mov     edi, 1
0xad57: push    rbp
0xad58: lea     rbp, [rip + 0x1fda1]
0xad5f: push    rbx
0xad60: sub     rsp, 0x50
0xad64: mov     rax, qword ptr fs:[0x28]
0xad6d: mov     qword ptr [rsp + 0x48], rax
0xad72: xor     eax, eax
0xad74: mov     r12, rsp
0xad77: mov     rsi, r12
0xad7a: call    0x4d60
0xad7f: mov     edx, 0xff
0xad84: mov     esi, 1
0xad89: mov     edi, 1
0xad8e: mov     word ptr [rsp + 0x11], dx
0xad93: mov     rdx, r12
0xad96: call    0x4d70
0xad9b: mov     edi, 1
0xada0: mov     dword ptr [rip + 0x1ef7a], 0xffffffff
0xadaa: call    0x18b70
0xadaf: mov     dword ptr [rip + 0x20147], 0
0xadb9: nop     dword ptr [rax]

```

```

0xadc0: xor     eax, eax
0xadc2: call    0xa820
0xadc7: mov     rdx, qword ptr [rip + 0x1fb5a]
0xadce: test    rdx, rdx
0xadd1: je      0xadd7
0xadd3: xor     eax, eax
0xadd5: call    rdx
0xadd7: mov     edx, 0x400
0xaddc: mov     rsi, rbp
0xaddf: xor     edi, edi
0xade1: call    0x18d30
0xade6: mov     rdx, qword ptr [rip + 0x1fb33]
0xaded: mov     ebx, eax
0xadeb: test    rdx, rdx
0xadf2: je      0xadf8
0xadf4: xor     eax, eax
0xadf6: call    rdx
0xadf8: mov     dword ptr [rip + 0x20106], ebx
0xadfe: test    ebx, ebx
0xae00: js      0xadc0
0xae02: xor     edi, edi
0xae04: call    0x18b70
0xae09: mov     rsi, r12
0xae0c: mov     edi, 1
0xae11: call    0x4d60
0xae16: mov     eax, 7
0xae1b: mov     rdx, r12
0xae1e: mov     esi, 1
0xae23: mov     edi, 1
0xae28: mov     word ptr [rsp + 0x11], ax
0xae2d: call    0x4d70
0xae32: mov     eax, dword ptr [rip + 0x200cc]
0xae38: mov     dword ptr [rip + 0x1eee2], 7
0xae42: test    eax, eax
0xae44: je      0xae9f
0xae46: movsxd  rsi, dword ptr [rip + 0x200b3]
0xae4d: movsx   r8d, byte ptr [rbp + rsi]
0xae53: mov     rdx, rsi
0xae56: mov     ecx, r8d
0xae59: test    r8b, r8b
0xae5c: js      0xae90
0xae5e: sub     eax, 1
0xae61: add     edx, 1
0xae64: mov     dword ptr [rip + 0x2009a], eax
0xae6a: mov     dword ptr [rip + 0x20090], edx
0xae70: mov     rax, qword ptr [rsp + 0x48]
0xae75: sub     rax, qword ptr fs:[0x28]
0xae7e: jne     0xaea7

```

```

0xae80: add     rsp, 0x50
0xae84: mov     eax, r8d
0xae87: pop     rbx
0xae88: pop     rbp
0xae89: pop     r12
0xae8b: ret
0xae8c: nop     dword ptr [rax]
0xae90: and     ecx, 0x7f
0xae93: mov     r8d, 0x1b
0xae99: mov     byte ptr [rbp + rsi], cl
0xae9d: jmp     0xae70
0xae70:
0xae9f: mov     r8d, 0xffffffff
0xaea5: jmp     0xae70
0xae70:
0xaea7: call    0x49d0
0xaeac: nop     dword ptr [rax]
0xaeb0: endbr64
0xaeb4: xor     eax, eax
0xaeb6: sub     rsp, 8
0xaeba: call    0x18ab0
0xaebf: mov     eax, dword ptr [rip + 0x2003f]
0xaec5: test    eax, eax
0xaec7: jne     0xaed8
0xaec9: xor     eax, eax
0xaecb: add     rsp, 8
0xaecf: jmp     0xad50
0xad50:
0xaed4: nop     dword ptr [rax]
0xaed8: movsxd  rsi, dword ptr [rip + 0x20021]
0xaedf: lea     rdi, [rip + 0x1fc1a]
0xae66: movsx   r8d, byte ptr [rdi + rsi]
0xae6b: mov     rdx, rsi
0xae6e: mov     ecx, r8d
0xae71: test    r8b, r8b
0xae74: js      0xaf10
0xae76: sub     eax, 1
0xae79: add     edx, 1
0xae7c: mov     dword ptr [rip + 0x20002], eax
0xaf02: mov     dword ptr [rip + 0x1fff8], edx
0xaf08: mov     eax, r8d
0xaf0b: add     rsp, 8
0xaf0f: ret
0xaf10: and     ecx, 0x7f
0xaf13: mov     r8d, 0x1b
0xaf19: mov     byte ptr [rdi + rsi], cl
0xaf1c: jmp     0xaf08
0xaf08:

```

```

0xaf1e: nop
0xaf20: endbr64
0xaf24: push    r13
0xaf26: mov     r13, rsi
0xaf29: push    r12
0xaf2b: mov     r12, rdx
0xaf2e: push    rbp
0xaf2f: mov     rbp, rdi
0xaf32: mov     edi, 0x18
0xaf37: push    rbx
0xaf38: sub     rsp, 8
0xaf3c: call    0x1d670
0xaf41: mov     rdi, r13
0xaf44: mov     rbx, rax
0xaf47: call    0x1d7a0
0xaf4c: mov     qword ptr [rbx + 0x10], r12
0xaf50: mov     qword ptr [rbx], rax
0xaf53: mov     rax, qword ptr [rbp]
0xaf57: mov     qword ptr [rbx + 8], rax
0xaf5b: mov     qword ptr [rbp], rbx
0xaf5f: add     rsp, 8
0xaf63: pop     rbx
0xaf64: pop     rbp
0xaf65: pop     r12
0xaf67: pop     r13
0xaf69: ret
0xaf6a: nop     word ptr [rax + rax]
0xaf70: endbr64
0xaf74: cmp     byte ptr [rdi], 0
0xaf77: je      0xb030
0xaf7d: push    r13
0xaf7f: push    r12
0xaf81: mov     r12, rsi
0xaf84: push    rbp
0xaf85: mov     rbp, rdi
0xaf88: push    rbx
0xaf89: sub     rsp, 8
0xaf8d: mov     r13, qword ptr [rip + 0x1fa0c]
0xaf94: test    r13, r13
0xaf97: je      0xb010
0xaf99: mov     rbx, qword ptr [r13]
0xaf9d: mov     rsi, qword ptr [rbx]
0xafa0: call    0x4b30
0xafa5: test    eax, eax
0xafa7: jle     0xb010
0xafa9: nop     dword ptr [rax]
0xafb0: lea     r13, [rbx + 8]
0xafb4: mov     rbx, qword ptr [rbx + 8]

```

```

0xafb8: mov     qword ptr [rip + 0x1f9e1], r13
0xafbf: test    rbx, rbx
0xafc2: je      0xafd3
0xafc4: mov     rsi, qword ptr [rbx]
0xafc7: mov     rdi, rbp
0xafca: call    0x4b30
0xafcf: test    eax, eax
0xafd1: jg      0xafb0
0xafd3: mov     edi, 0x18
0xafd8: call    0x1d670
0xafdd: mov     rdi, rbp
0xafe0: mov     rbx, rax
0xafe3: call    0x1d7a0
0xafe8: mov     qword ptr [rbx + 0x10], r12
0xafec: mov     qword ptr [rbx], rax
0xafef: mov     rax, qword ptr [r13]
0xaff3: mov     qword ptr [rbx + 8], rax
0xaff7: mov     qword ptr [r13], rbx
0xaffb: add     rsp, 8
0xafff: pop     rbx
0xb000: pop     rbp
0xb001: pop     r12
0xb003: pop     r13
0xb005: ret
0xb006: nop     word ptr cs:[rax + rax]
0xb010: mov     rbx, qword ptr [rip + 0x1f969]
0xb017: lea     r13, [rip + 0x1f962]
0xb01e: mov     qword ptr [rip + 0x1f97b], r13
0xb025: test    rbx, rbx
0xb028: jne     0xafc4
0xb02a: jmp     0xafd3
0xafd3:
0xb02c: nop     dword ptr [rax]
0xb030: ret
0xb031: nop     word ptr cs:[rax + rax]
0xb03c: nop     dword ptr [rax]
0xb040: endbr64
0xb044: mov     rax, qword ptr [rip + 0x1f935]
0xb04b: mov     qword ptr [rip + 0x1f926], rax
0xb052: ret
0xb053: nop     word ptr cs:[rax + rax]
0xb05e: nop
0xb060: endbr64
0xb064: cmp     qword ptr [rip + 0x1f90c], 0
0xb06c: je      0xb073
0xb06e: jmp     0x99e0
0x99e0:
0xb073: xor     eax, eax

```

```

0xb075: ret
0xb076: nop      word ptr cs:[rax + rax]
0xb080: endbr64
0xb084: mov      eax, dword ptr [rip + 0x1f912]
0xb08a: push     r12
0xb08c: mov      r12, rdi
0xb08f: push     rbp
0xb090: push     rbx
0xb091: sub      eax, 1
0xb094: movzx    ebx, byte ptr [rip + 0x202b6]
0xb09b: mov      dword ptr [rip + 0x202b3], eax
0xb0a1: mov      dword ptr [rip + 0x202ad], 0
0xb0ab: mov      eax, ebx
0xb0ad: and      eax, 0xffffffffc0
0xb0b0: or       eax, 0x38
0xb0b3: mov      byte ptr [rip + 0x20298], al
0xb0b9: mov      eax, dword ptr [rip + 0x1f8d9]
0xb0bf: lea      edi, [rax + 1]
0xb0c2: movsxd   rdi, edi
0xb0c5: call     0x1d670
0xb0ca: movsxd   rdx, dword ptr [rip + 0x1f8c7]
0xb0d1: mov      esi, 0x20
0xb0d6: mov      rdi, rax
0xb0d9: mov      rbp, rax
0xb0dc: call     0x4a50
0xb0e1: movsxd   rax, dword ptr [rip + 0x1f8b0]
0xb0e8: mov      rdi, rbp
0xb0eb: mov      byte ptr [rbp + rax], 0
0xb0f0: mov      esi, dword ptr [rip + 0x1f8a2]
0xb0f6: call     0xa1a0
0xb0fb: mov      rdi, rbp
0xb0fe: lea      rbp, [rip + 0x1f8bb]
0xb105: call     0x18d10
0xb10a: mov      eax, dword ptr [rip + 0x1f88c]
0xb110: mov      rdi, r12
0xb113: mov      dword ptr [rip + 0x2023b], 0
0xb11d: sub      eax, 1
0xb120: mov      dword ptr [rip + 0x2022e], eax
0xb126: call     0x9fe0
0xb12b: mov      esi, 0x13
0xb130: lea      rdi, [rip + 0x15ad5]
0xb137: call     0xa1a0
0xb13c: mov      rdi, rbp
0xb13f: call     0x49b0
0xb144: mov      rdi, rbp
0xb147: mov      esi, eax
0xb149: call     0xa1a0
0xb14e: mov      esi, 1

```

```

0xb153: lea      rdi, [rip + 0x155eb]
0xb15a: call     0xa1a0
0xb15f: xor      eax, eax
0xb161: call     0xa820
0xb166: mov      byte ptr [rip + 0x201e5], bl
0xb16c: pop      rbx
0xb16d: pop      rbp
0xb16e: pop      r12
0xb170: ret
0xb171: nop      word ptr cs:[rax + rax]
0xb17c: nop      dword ptr [rax]
0xb180: endbr64
0xb184: push     r15
0xb186: push     r14
0xb188: mov      r14, rdi
0xb18b: push     r13
0xb18d: push     r12
0xb18f: push     rbp
0xb190: lea      rbp, [rip + 0x1f969]
0xb197: push     rbx
0xb198: sub      rsp, 0x28
0xb19c: jmp      0xb1ae
0xb1ae:
0xb19e: nop
0xb1a0: xor      eax, eax
0xb1a2: call     0xad50
0xb1a7: mov      ebx, eax
0xb1a9: cmp      ebx, -1
0xb1ac: jne      0xb1ef
0xb1ae: xor      eax, eax
0xb1b0: call     0x18ab0
0xb1b5: mov      eax, dword ptr [rip + 0x1fd49]
0xb1bb: test     eax, eax
0xb1bd: je       0xb1a0
0xb1bf: movsxd   rsi, dword ptr [rip + 0x1fd3a]
0xb1c6: movsx    ebx, byte ptr [rbp + rsi]
0xb1cb: mov      rdx, rsi
0xb1ce: mov      ecx, ebx
0xb1d0: test     bl, bl
0xb1d2: js       0xb4a0
0xb1d8: sub      eax, 1
0xb1db: add      edx, 1
0xb1de: mov      dword ptr [rip + 0x1fd20], eax
0xb1e4: mov      dword ptr [rip + 0x1fd16], edx
0xb1ea: cmp      ebx, -1
0xb1ed: je       0xb1ae
0xb1ef: test     r14, r14
0xb1f2: je       0xb1fb

```

```

0xb1f4: mov     dword ptr [r14], 1
0xb1fb: mov     eax, dword ptr [rip + 0x201fb]
0xb201: test    ebx, ebx
0xb203: jne     0xb445
0xb209: mov     ebx, 0xff
0xb20e: test    eax, eax
0xb210: jne     0xb240
0xb212: xor     r12d, r12d
0xb215: mov     ebp, 0x1fe
0xb21a: mov     ebx, 0xff
0xb21f: call    0x4f00
0xb224: mov     rax, qword ptr [rax]
0xb227: movzx   eax, word ptr [rax + rbp]
0xb22b: test    ah, 0x40
0xb22e: jne     0xb47d
0xb234: test    r12b, r12b
0xb237: jne     0xb47d
0xb23d: nop     dword ptr [rax]
0xb240: mov     rax, qword ptr [rip + 0x1f739]
0xb247: xor     r12d, r12d
0xb24a: lea     r13, [rip + 0x1f8af]
0xb251: mov     dword ptr [rip + 0x1f751], 0
0xb25b: mov     dword ptr [rip + 0x1f743], 0
0xb265: mov     qword ptr [rip + 0x1f70c], rax
0xb26c: test    ebx, ebx
0xb26e: mov     eax, 0xffffffff
0xb273: movsxd  rbp, r12d
0xb276: cmovne  eax, ebx
0xb279: mov     rbx, qword ptr [rip + 0x200e0]
0xb280: mov     byte ptr [rbx + rbp], al
0xb283: add     rbp, 1
0xb287: mov     byte ptr [rbx + rbp], 0
0xb28b: cmp     qword ptr [rip + 0x1f6e5], 0
0xb293: je      0xb2b3
0xb295: mov     rdi, rbx
0xb298: call    0x99e0
0xb29d: mov     r15, rax
0xb2a0: cmp     rax, -1
0xb2a4: je      0xb33a
0xb2aa: test    rax, rax
0xb2ad: jne     0xb38f
0xb2b3: mov     ecx, dword ptr [rip + 0x1fc4b]
0xb2b9: test    ecx, ecx
0xb2bb: jne     0xb2ee
0xb2bd: mov     edx, dword ptr [rip + 0x1f6e5]
0xb2c3: test    edx, edx
0xb2c5: je      0xb380
0xb2cb: mov     rdi, rbx

```



```

0xb2ce: call    0xb080
0xb2d3: mov     dword ptr [rip + 0x1f6cf], 1
0xb2dd: jmp     0xb2ee
0xb2ee
0xb2df: nop
0xb2e0: xor     eax, eax
0xb2e2: call    0xad50
0xb2e7: mov     ebx, eax
0xb2e9: cmp     ebx, -1
0xb2ec: jne     0xb32c
0xb2ee: xor     eax, eax
0xb2f0: call    0x18ab0
0xb2f5: mov     eax, dword ptr [rip + 0x1fc09]
0xb2fb: test    eax, eax
0xb2fd: je      0xb2e0
0xb2ff: movsxd  rsi, dword ptr [rip + 0x1fbfa]
0xb306: movsx   ebx, byte ptr [r13 + rsi]
0xb30c: mov     rdx, rsi
0xb30f: mov     ecx, ebx
0xb311: test    bl, bl
0xb313: js      0xb360
0xb315: sub     eax, 1
0xb318: add     edx, 1
0xb31b: mov     dword ptr [rip + 0x1fbe3], eax
0xb321: mov     dword ptr [rip + 0x1fbd9], edx
0xb327: cmp     ebx, -1
0xb32a: je      0xb2ee
0xb32c: add     r12d, 1
0xb330: cmp     r12d, 0xf
0xb334: jne     0xb26c
0xb33a: mov     edi, 1
0xb33f: xor     r15d, r15d
0xb342: call    0x4aa0
0xb347: mov     dword ptr [rip + 0x1f65b], 0
0xb351: jmp     0xb433
0xb433
0xb356: nop     word ptr cs:[rax + rax]
0xb360: and     ecx, 0x7f
0xb363: add     r12d, 1
0xb367: mov     byte ptr [r13 + rsi], cl
0xb36c: cmp     r12d, 0xf
0xb370: je      0xb33a
0xb372: mov     eax, 0x1b
0xb377: jmp     0xb279
0xb279
0xb37c: nop     dword ptr [rax]
0xb380: mov     edi, 1
0xb385: call    0x4aa0

```

```

0xb38a: jmp      0xb2d3
0xb2d3:
0xb38f: test     r14, r14
0xb392: je       0xb41f
0xb398: mov      ecx, dword ptr [rip + 0x1fb66]
0xb39e: cmp      r12d, ecx
0xb3a1: jge      0xb41f
0xb3a3: movsxd   rax, dword ptr [rip + 0x1fb56]
0xb3aa: lea      r9d, [r12 + 1]
0xb3af: lea      rdx, [rip + 0x1f74a]
0xb3b6: movsxd   rdi, r9d
0xb3b9: mov      qword ptr [rsp + 0x18], rdi
0xb3be: lea      r8d, [r9 + rax]
0xb3c2: lea      r13, [rax + rdx]
0xb3c6: jmp      0xb3f1
0xb3f1:
0xb3c8: nop      dword ptr [rax + rax]
0xb3d0: sub      ecx, r9d
0xb3d3: add      dword ptr [r14], 1
0xb3d7: add      r13, qword ptr [rsp + 0x18]
0xb3dc: mov      dword ptr [rip + 0x1fb1d], r8d
0xb3e3: add      r8d, r9d
0xb3e6: mov      dword ptr [rip + 0x1fb18], ecx
0xb3ec: cmp      r12d, ecx
0xb3ef: jge      0xb41f
0xb3f1: mov      rdx, rbp
0xb3f4: mov      rsi, r13
0xb3f7: mov      rdi, rbx
0xb3fa: mov      dword ptr [rsp + 0x14], r8d
0xb3ff: mov      dword ptr [rsp + 0x10], r9d
0xb404: mov      dword ptr [rsp + 0xc], ecx
0xb408: call     0x4af0
0xb40d: mov      ecx, dword ptr [rsp + 0xc]
0xb411: mov      r9d, dword ptr [rsp + 0x10]
0xb416: test     eax, eax
0xb418: mov      r8d, dword ptr [rsp + 0x14]
0xb41d: je       0xb3d0
0xb41f: mov      edi, 1
0xb424: call     0x4aa0
0xb429: mov      dword ptr [rip + 0x1f579], 0
0xb433: add      rsp, 0x28
0xb437: mov      rax, r15
0xb43a: pop      rbx
0xb43b: pop      rbp
0xb43c: pop      r12
0xb43e: pop      r13
0xb440: pop      r14
0xb442: pop      r15

```

```

0xb444: ret
0xb445: test    eax, eax
0xb447: jne     0xb240
0xb44d: cmp     ebx, 0xa
0xb450: sete    bpl
0xb454: cmp     ebx, 0xd
0xb457: sete    al
0xb45a: or      ebp, eax
0xb45c: call    0x4f00
0xb461: mov     rax, qword ptr [rax]
0xb464: test    bpl, bpl
0xb467: je      0xb4c6
0xb469: mov     ebx, 0xa
0xb46e: test    byte ptr [rax + 0x15], 0x40
0xb472: je      0xb240
0xb478: mov     ebx, 0xa
0xb47d: mov     rax, qword ptr [rip + 0x1f4dc]
0xb484: lea     r15, [rip + 0x1f4d5]
0xb48b: mov     byte ptr [rax], bl
0xb48d: mov     rax, qword ptr [rip + 0x1f4cc]
0xb494: mov     byte ptr [rax + 1], 0
0xb498: jmp     0xb433
0xb433
0xb49a: nop     word ptr [rax + rax]
0xb4a0: and     ecx, 0x7f
0xb4a3: mov     byte ptr [rbp + rsi], cl
0xb4a7: test    r14, r14
0xb4aa: je      0xb4b3
0xb4ac: mov     dword ptr [r14], 1
0xb4b3: mov     eax, dword ptr [rip + 0x1ff43]
0xb4b9: mov     ebx, 0x1b
0xb4be: test    eax, eax
0xb4c0: jne     0xb240
0xb4c6: movsxd  rbp, ebx
0xb4c9: add     rbp, rbp
0xb4cc: cmp     ebx, 7
0xb4cf: sete    r12b
0xb4d3: jmp     0xb21f
0xb21f
0xb4d8: nop     dword ptr [rax + rax]
0xb4e0: endbr64
0xb4e4: mov     eax, dword ptr [rip + 0x1f4c2]
0xb4ea: test    eax, eax
0xb4ec: jne     0xb4f0
0xb4ee: ret
0xb4ef: nop
0xb4f0: sub     rsp, 8
0xb4f4: mov     rdi, qword ptr [rip + 0x1fe65]

```

```

0xb4fb: call    0xb080
0xb500: mov     dword ptr [rip + 0x1f49e], 1
0xb50a: add     rsp, 8
0xb50e: ret
0xb50f: nop
0xb510: endbr64
0xb514: mov     rax, qword ptr [rip + 0x1fe45]
0xb51b: ret
0xb51c: nop     dword ptr [rax]
0xb520: endbr64
0xb524: push    r15
0xb526: mov     esi, 0x5413
0xb52b: mov     edi, 1
0xb530: push    r14
0xb532: push    r13
0xb534: push    r12
0xb536: push    rbp
0xb537: push    rbx
0xb538: sub     rsp, 0x58
0xb53c: mov     rax, qword ptr fs:[0x28]
0xb545: mov     qword ptr [rsp + 0x48], rax
0xb54a: xor     eax, eax
0xb54c: lea     rdx, [rsp + 0x18]
0xb551: mov     dword ptr [rsp + 0x10], 0
0xb559: mov     dword ptr [rsp + 0x14], 0
0xb561: call    0x4a70
0xb566: test    eax, eax
0xb568: jne     0xb5a0
0xb56a: mov     ebx, eax
0xb56c: movzx   eax, word ptr [rsp + 0x1a]
0xb571: test    ax, ax
0xb574: je      0xb790
0xb57a: movzx   ebp, word ptr [rsp + 0x18]
0xb57f: test    bp, bp
0xb582: je      0xb790
0xb588: movzx   ebx, ax
0xb58b: lea     r12d, [rbp - 1]
0xb58f: lea     eax, [rbx - 1]
0xb592: mov     dword ptr [rsp + 0xc], eax
0xb596: jmp     0xb5b2
0xb5b2:
0xb598: nop     dword ptr [rax + rax]
0xb5a0: mov     dword ptr [rsp + 0xc], 0xffffffff
0xb5a8: mov     r12d, 0xffffffff
0xb5ae: xor     ebx, ebx
0xb5b0: xor     ebp, ebp
0xb5b2: lea     r13, [rip + 0x15667]
0xb5b9: lea     r15, [rip + 0x15a13]

```

```

0xb5c0: mov     rdi, r13
0xb5c3: call    0x47e0
0xb5c8: mov     rdi, rax
0xb5cb: test    rax, rax
0xb5ce: je      0xb5df
0xb5d0: lea     rdx, [rsp + 0x14]
0xb5d5: mov     rsi, r15
0xb5d8: xor     eax, eax
0xb5da: call    0x4c60
0xb5df: lea     r14, [rip + 0x15642]
0xb5e6: mov     rdi, r14
0xb5e9: call    0x47e0
0xb5ee: mov     rdi, rax
0xb5f1: test    rax, rax
0xb5f4: je      0xb605
0xb5f6: lea     rdx, [rsp + 0x10]
0xb5fb: mov     rsi, r15
0xb5fe: xor     eax, eax
0xb600: call    0x4c60
0xb605: cmp     dword ptr [rsp + 0xc], 0x3ff
0xb60d: mov     eax, dword ptr [rip + 0x1e8fd]
0xb613: jbe     0xb628
0xb615: mov     ebx, dword ptr [rsp + 0x14]
0xb619: lea     edx, [rbx - 1]
0xb61c: cmp     edx, 0x3ff
0xb622: ja      0xb7c0
0xb628: mov     dword ptr [rip + 0x1f36a], ebx
0xb62e: cmp     r12d, 0x3ff
0xb635: jbe     0xb64a
0xb637: mov     ebp, dword ptr [rsp + 0x10]
0xb63b: lea     edx, [rbp - 1]
0xb63e: cmp     edx, 0x3ff
0xb644: ja      0xb7a8
0xb64a: lea     r12, [rsp + 0x20]
0xb64f: mov     r8d, ebp
0xb652: mov     rcx, r15
0xb655: mov     edx, 0x20
0xb65a: mov     esi, 1
0xb65f: mov     rdi, r12
0xb662: xor     eax, eax
0xb664: mov     dword ptr [rip + 0x1f332], ebp
0xb66a: call    0x4f10
0xb66f: mov     rsi, r12
0xb672: mov     rdi, r14
0xb675: call    0x18380
0xb67a: mov     rcx, r15
0xb67d: mov     edx, 0x20
0xb682: mov     rdi, r12

```

```

0xb685: mov     r8d, dword ptr [rip + 0x1f30c]
0xb68c: mov     esi, 1
0xb691: xor     eax, eax
0xb693: call    0x4f10
0xb698: mov     rdi, r13
0xb69b: mov     rsi, r12
0xb69e: call    0x18380
0xb6a3: mov     rdi, qword ptr [rip + 0x1fc8e]
0xb6aa: test    rdi, rdi
0xb6ad: je      0xb6b4
0xb6af: call    0x18d10
0xb6b4: mov     rdi, qword ptr [rip + 0x1fc75]
0xb6bb: test    rdi, rdi
0xb6be: je      0xb6c5
0xb6c0: call    0x18d10
0xb6c5: mov     rdi, qword ptr [rip + 0x1fc5c]
0xb6cc: test    rdi, rdi
0xb6cf: je      0xb6d6
0xb6d1: call    0x18d10
0xb6d6: mov     rdi, qword ptr [rip + 0x1fc43]
0xb6dd: test    rdi, rdi
0xb6e0: je      0xb6e7
0xb6e2: call    0x18d10
0xb6e7: mov     edi, dword ptr [rip + 0x1f2ab]
0xb6ed: imul    edi, dword ptr [rip + 0x1f2a8]
0xb6f4: mov     esi, 1
0xb6f9: movsxd  rdi, edi
0xb6fc: call    0x1d740
0xb701: mov     edi, dword ptr [rip + 0x1f291]
0xb707: mov     esi, 1
0xb70c: imul    edi, dword ptr [rip + 0x1f289]
0xb713: mov     qword ptr [rip + 0x1fc1e], rax
0xb71a: movsxd  rdi, edi
0xb71d: call    0x1d740
0xb722: mov     edi, dword ptr [rip + 0x1f270]
0xb728: mov     esi, 1
0xb72d: imul    edi, dword ptr [rip + 0x1f268]
0xb734: mov     qword ptr [rip + 0x1fbf5], rax
0xb73b: movsxd  rdi, edi
0xb73e: call    0x1d740
0xb743: mov     edi, dword ptr [rip + 0x1f24f]
0xb749: mov     esi, 1
0xb74e: imul    edi, dword ptr [rip + 0x1f247]
0xb755: mov     qword ptr [rip + 0x1fbcc], rax
0xb75c: movsxd  rdi, edi
0xb75f: call    0x1d740
0xb764: mov     qword ptr [rip + 0x1fbb5], rax
0xb76b: mov     rax, qword ptr [rsp + 0x48]

```

```

0xb770: sub    rax, qword ptr fs:[0x28]
0xb779: jne    0xb7dc
0xb77b: add    rsp, 0x58
0xb77f: pop    rbx
0xb780: pop    rbp
0xb781: pop    r12
0xb783: pop    r13
0xb785: pop    r14
0xb787: pop    r15
0xb789: ret
0xb78a: nop    word ptr [rax + rax]
0xb790: mov    dword ptr [rsp + 0xc], 0xffffffff
0xb798: xor    ebp, ebp
0xb79a: mov    r12d, 0xffffffff
0xb7a0: jmp    0xb5b2
0xb5b2
0xb7a5: nop    dword ptr [rax]
0xb7a8: lea    edx, [rax - 1]
0xb7ab: mov    ebp, 0x18
0xb7b0: cmp    edx, 0x400
0xb7b6: cmovb  ebp, eax
0xb7b9: jmp    0xb64a
0xb64a
0xb7be: nop
0xb7c0: mov    ebx, dword ptr [rip + 0x1e72a]
0xb7c6: lea    edx, [rbx - 1]
0xb7c9: cmp    edx, 0x3ff
0xb7cf: mov    edx, 0x50
0xb7d4: cmova  ebx, edx
0xb7d7: jmp    0xb628
0xb628
0xb7dc: call   0x49d0
0xb7e1: nop    word ptr cs:[rax + rax]
0xb7ec: nop    dword ptr [rax]
0xb7f0: endbr64
0xb7f4: push   r15
0xb7f6: push   r14
0xb7f8: push   r13
0xb7fa: push   r12
0xb7fc: mov    r12, rdi
0xb7ff: mov    edi, 0x5858
0xb804: push   rbp
0xb805: push   rbx
0xb806: mov    ebx, esi
0xb808: sub    rsp, 0x38
0xb80c: mov    rax, qword ptr fs:[0x28]
0xb815: mov    qword ptr [rsp + 0x28], rax
0xb81a: movabs rax, 0x7363762f7665642f

```

```

0xb824: mov     word ptr [rsp + 8], di
0xb829: mov     byte ptr [rsp + 0xa], 0
0xb82e: mov     dword ptr [rsp + 0x18], 0x585861
0xb836: mov     qword ptr [rsp], rax
0xb83a: mov     qword ptr [rsp + 0x10], rax
0xb83f: cmp     esi, 1
0xb842: je      0xb978
0xb848: mov     esi, dword ptr [rip + 0x1fbb2]
0xb84e: mov     r13, qword ptr [rip + 0x1d783]
0xb855: mov     r14d, 1
0xb85b: test    esi, esi
0xb85d: jne     0xb90c
0xb863: mov     rax, qword ptr [rip + 0x1f126]
0xb86a: lea     r15, [rsp + 0x10]
0xb86f: mov     esi, r14d
0xb872: mov     dword ptr [rip + 0x1fb84], 0
0xb87c: mov     rdi, r15
0xb87f: movzx   edx, byte ptr [rax + 8]
0xb883: mov     byte ptr [rsp + 0x19], dl
0xb887: movzx   eax, byte ptr [rax + 9]
0xb88b: mov     byte ptr [rsp + 0x1a], al
0xb88f: call    0x4930
0xb894: mov     ebp, eax
0xb896: cmp     eax, -1
0xb899: je      0xb8f0
0xb89b: mov     eax, dword ptr [rip + 0x1f0fb]
0xb8a1: imul    eax, dword ptr [rip + 0x1f0f0]
0xb8a8: mov     edi, ebp
0xb8aa: mov     rsi, r12
0xb8ad: lea     edx, [rax + rax + 4]
0xb8b1: xor     eax, eax
0xb8b3: call    r13
0xb8b6: mov     edi, ebp
0xb8b8: call    0x4ab0
0xb8bd: cmp     ebx, 2
0xb8c0: je      0xb990
0xb8c6: mov     eax, 1
0xb8cb: mov     rdx, qword ptr [rsp + 0x28]
0xb8d0: sub     rdx, qword ptr fs:[0x28]
0xb8d9: jne     0xbb07
0xb8df: add     rsp, 0x38
0xb8e3: pop     rbx
0xb8e4: pop     rbp
0xb8e5: pop     r12
0xb8e7: pop     r13
0xb8e9: pop     r14
0xb8eb: pop     r15
0xb8ed: ret

```



```

0xb8ee: nop
0xb8f0: mov     ecx, 0x30
0xb8f5: mov     esi, r14d
0xb8f8: mov     rdi, r15
0xb8fb: mov     word ptr [rsp + 0x19], cx
0xb900: call    0x4930
0xb905: mov     ebp, eax
0xb907: cmp     eax, -1
0xb90a: jne     0xb89b
0xb90c: mov     dword ptr [rip + 0x1faea], 1
0xb916: cmp     ebx, 2
0xb919: je      0xba30
0xb91f: mov     rax, qword ptr [rip + 0x1f06a]
0xb926: mov     r15, rsp
0xb929: mov     esi, r14d
0xb92c: mov     rdi, r15
0xb92f: movzx   edx, byte ptr [rax + 8]
0xb933: movzx   eax, byte ptr [rax + 9]
0xb937: mov     byte ptr [rsp + 8], dl
0xb93b: mov     byte ptr [rsp + 9], al
0xb93f: call    0x4930
0xb944: mov     ebp, eax
0xb946: cmp     eax, -1
0xb949: je      0xba00
0xb94f: mov     edx, dword ptr [rip + 0x1f043]
0xb955: mov     edi, ebp
0xb957: mov     rsi, r12
0xb95a: xor     eax, eax
0xb95c: imul    edx, dword ptr [rip + 0x1f039]
0xb963: add     edx, 4
0xb966: call    r13
0xb969: mov     edi, ebp
0xb96b: call    0x4ab0
0xb970: jmp     0xb8c6
0xb8c6:
0xb975: nop     dword ptr [rax]
0xb978: mov     r13, qword ptr [rip + 0x1d661]
0xb97f: xor     r14d, r14d
0xb982: jmp     0xb863
0xb863:
0xb987: nop     word ptr [rax + rax]
0xb990: movsx   esi, byte ptr [r12 + 2]
0xb996: movsx   edi, byte ptr [r12 + 3]
0xb99c: call    0xa7a0
0xb9a1: mov     edx, dword ptr [rip + 0x1f561]
0xb9a7: test    edx, edx
0xb9a9: jle     0xb9e5
0xb9ab: xor     ebp, ebp

```

```

0xb9ad: lea      rbx, [rip + 0x1f56c]
0xb9b4: jmp      0xb9cc
0xb9cc:
0xb9b6: nop      word ptr cs:[rax + rax]
0xb9c0: mov      edx, dword ptr [rip + 0x1f542]
0xb9c6: add      ebp, eax
0xb9c8: cmp      edx, ebp
0xb9ca: jle      0xb9e5
0xb9cc: sub      edx, ebp
0xb9ce: movsxd   rsi, ebp
0xb9d1: mov      edi, 1
0xb9d6: add      rsi, rbx
0xb9d9: movsxd   rdx, edx
0xb9dc: call     0x18da0
0xb9e1: test     eax, eax
0xb9e3: jns      0xb9c0
0xb9e5: mov      dword ptr [rip + 0x1f519], 0
0xb9ef: mov      eax, 1
0xb9f4: jmp      0xb8cb
0xb8cb:
0xb9f9: nop      dword ptr [rax]
0xba00: mov      edx, 0x30
0xba05: mov      esi, r14d
0xba08: mov      rdi, r15
0xba0b: mov      word ptr [rsp + 8], dx
0xba10: call     0x4930
0xba15: mov      ebp, eax
0xba17: cmp      eax, -1
0xba1a: jne      0xb94f
0xba20: xor      eax, eax
0xba22: jmp      0xb8cb
0xb8cb:
0xba27: nop      word ptr [rax + rax]
0xba30: xor      eax, eax
0xba32: mov      r15, rsp
0xba35: call     0xa300
0xba3a: mov      rax, qword ptr [rip + 0x1ef4f]
0xba41: mov      esi, r14d
0xba44: mov      rdi, r15
0xba47: movzx    edx, byte ptr [rax + 8]
0xba4b: movzx    eax, byte ptr [rax + 9]
0xba4f: mov      byte ptr [rsp + 8], dl
0xba53: mov      byte ptr [rsp + 9], al
0xba57: call     0x4930
0xba5c: mov      ebp, eax
0xba5e: cmp      eax, -1
0xba61: je       0xbae0
0xba63: mov      edx, dword ptr [rip + 0x1ef2f]

```

```

0xba69: mov     rsi, r12
0xba6c: mov     edi, ebp
0xba6e: xor     eax, eax
0xba70: imul    edx, dword ptr [rip + 0x1ef25]
0xba77: add     edx, 4
0xba7a: call    r13
0xba7d: mov     edi, ebp
0xba7f: call    0x4ab0
0xba84: mov     eax, dword ptr [rip + 0x1ef12]
0xba8a: xor     esi, esi
0xba8c: lea     edi, [rax - 1]
0xba8f: call    0xa7a0
0xba94: mov     edx, dword ptr [rip + 0x1f46e]
0xba9a: test    edx, edx
0xba9c: jle     0xb9e5
0xbaa2: xor     ebp, ebp
0xbaa4: lea     rbx, [rip + 0x1f475]
0xbaab: jmp     0xbac0
0xbac0:
0xbaad: nop     dword ptr [rax]
0xbab0: mov     edx, dword ptr [rip + 0x1f452]
0xbab6: add     ebp, eax
0xbab8: cmp     edx, ebp
0xbaba: jle     0xb9e5
0xbac0: sub     edx, ebp
0xbac2: movsxd  rsi, ebp
0xbac5: mov     edi, 1
0xbaca: add     rsi, rbx
0xbacd: movsxd  rdx, edx
0xbad0: call    0x18da0
0xbad5: test    eax, eax
0xbad7: jns     0xbab0
0xbad9: jmp     0xb9e5
0xb9e5:
0xbade: nop
0xbae0: mov     eax, 0x30
0xbae5: mov     esi, r14d
0xbae8: mov     rdi, r15
0xbaeb: mov     word ptr [rsp + 8], ax
0xbaf0: call    0x4930
0xbaf5: mov     ebp, eax
0xbaf7: cmp     eax, -1
0xbafa: jne     0xba63
0xbb00: xor     eax, eax
0xbb02: jmp     0xb8cb
0xb8cb:
0xbb07: call    0x49d0
0xbb0c: nop     dword ptr [rax]

```

```

0xbb10: endbr64
0xbb14: mov     eax, dword ptr [rip + 0x1ee6e]
0xbb1a: test    eax, eax
0xbb1c: jne     0xbb20
0xbb1e: ret
0xbb1f: nop
0xbb20: sub     rsp, 8
0xbb24: mov     esi, 1
0xbb29: xor     eax, eax
0xbb2b: call    0xb7f0
0xbb30: mov     dword ptr [rip + 0x1f8ce], eax
0xbb36: add     rsp, 8
0xbb3a: ret
0xbb3b: nop     dword ptr [rax + rax]
0xbb40: endbr64
0xbb44: push    rbp
0xbb45: xor     eax, eax
0xbb47: mov     rbp, rdi
0xbb4a: call    0xa6c0
0xbb4f: mov     edx, dword ptr [rip + 0x1ee33]
0xbb55: test    edx, edx
0xbb57: je      0xbb94
0xbb59: mov     eax, dword ptr [rip + 0x1f8a5]
0xbb5f: test    eax, eax
0xbb61: je      0xbb94
0xbb63: mov     edx, dword ptr [rip + 0x1ee33]
0xbb69: imul    edx, dword ptr [rip + 0x1ee28]
0xbb70: xor     esi, esi
0xbb72: mov     rdi, qword ptr [rip + 0x1f7af]
0xbb79: movsxd  rdx, edx
0xbb7c: call    0x4a50
0xbb81: xor     eax, eax
0xbb83: mov     esi, 2
0xbb88: mov     rdi, rbp
0xbb8b: call    0xb7f0
0xbb90: test    eax, eax
0xbb92: jne     0xbba0
0xbb94: xor     eax, eax
0xbb96: pop     rbp
0xbb97: jmp     0xa300
0xa300:
0xbb9c: nop     dword ptr [rax]
0xbba0: mov     edx, dword ptr [rip + 0x1edf6]
0xbba6: imul    edx, dword ptr [rip + 0x1edeb]
0xbbad: xor     esi, esi
0xbbae: mov     rdi, qword ptr [rip + 0x1f782]
0xbbb6: pop     rbp
0xbbb7: movsxd  rdx, edx

```

```

0xbbbba: jmp      0x4a50
0x4a50
0xbbbbf: nop
0xbbbc0: endbr64
0xbbbc4: push     rbp
0xbbbc5: mov      rbp, rdi
0xbbbc8: push     rbx
0xbbbc9: sub      rsp, 8
0xbbbcd: cmp      dword ptr [rip + 0x1ed84], 1
0xbbbd4: je       0xbcd0
0xbbbda: xor      eax, eax
0xbbbdc: call     0xa6c0
0xbbbe1: test     rbp, rbp
0xbbbe4: je       0xbbf4
0xbbbe6: mov      eax, dword ptr [rip + 0x1ed9c]
0xbbec: test     eax, eax
0xbbee: jne      0xbcc0
0xbbf4: mov      eax, dword ptr [rip + 0x1eda2]
0xbbfa: mov      rdi, qword ptr [rip + 0x1e347]
0xbc01: lea      rdx, [rip - 0x2198]
0xbc08: lea      esi, [rax - 1]
0xbc0b: call     0x4820
0xbc10: mov      edx, dword ptr [rip + 0x1f2f2]
0xbc16: test     edx, edx
0xbc18: jle      0xbc4d
0xbc1a: xor      ebp, ebp
0xbc1c: lea      rbx, [rip + 0x1f2fd]
0xbc23: jmp      0xbc34
0xbc34
0xbc25: nop      dword ptr [rax]
0xbc28: mov      edx, dword ptr [rip + 0x1f2da]
0xbc2e: add      ebp, eax
0xbc30: cmp      edx, ebp
0xbc32: jle      0xbc4d
0xbc34: sub      edx, ebp
0xbc36: movsxd   rsi, ebp
0xbc39: mov      edi, 1
0xbc3e: add      rsi, rbx
0xbc41: movsxd   rdx, edx
0xbc44: call     0x18da0
0xbc49: test     eax, eax
0xbc4b: jns      0xbc28
0xbc4d: mov      edi, dword ptr [rip + 0x1ed49]
0xbc53: xor      esi, esi
0xbc55: mov      dword ptr [rip + 0x1f2a9], 0
0xbc5f: call     0xa7a0
0xbc64: mov      edx, dword ptr [rip + 0x1f29e]
0xbc6a: test     edx, edx

```

```

0xbc6c: jle      0xbca5
0xbc6e: xor      ebp, ebp
0xbc70: lea      rbx, [rip + 0x1f2a9]
0xbc77: jmp      0xbc8c
0xbc8c:
0xbc79: nop      dword ptr [rax]
0xbc80: mov      edx, dword ptr [rip + 0x1f282]
0xbc86: add      ebp, eax
0xbc88: cmp      edx, ebp
0xbc8a: jle      0xbca5
0xbc8c: sub      edx, ebp
0xbc8e: movsxd   rsi, ebp
0xbc91: mov      edi, 1
0xbc96: add      rsi, rbx
0xbc99: movsxd   rdx, edx
0xbc9c: call     0x18da0
0xbca1: test     eax, eax
0xbca3: jns      0xbc80
0xbca5: mov      dword ptr [rip + 0x1f259], 0
0xbcaf: add      rsp, 8
0xbcb3: mov      edi, 0xa
0xbcb8: pop      rbx
0xbcb9: pop      rbp
0xbcba: jmp      0x4810
0x4810:
0xbcbf: nop
0xbcc0: mov      rdi, rbp
0xbcc3: call     0xbb40
0xbcc8: jmp      0xbbf4
0xbbf4:
0xbccd: nop      dword ptr [rax]
0xbcd0: lea      rdx, [rip + 0x1f6e9]
0xbcd7: mov      esi, 1
0xbcdc: mov      edi, 1
0xbce1: call     0x4d70
0xbce6: mov      dword ptr [rip + 0x1ec68], 0
0xbcf0: jmp      0xbbda
0xbbda:
0xbcf5: nop      word ptr cs:[rax + rax]
0xbd00: endbr64
0xbd04: push     r13
0xbd06: lea      r13, [rip + 0x1cc33]
0xbd0d: push     r12
0xbd0f: push     rbp
0xbd10: mov      rbp, rdi
0xbd13: lea      rdi, [rip + 0x14f14]
0xbd1a: push     rbx
0xbd1b: xor      ebx, ebx

```

```

0xbd1d: sub    rsp, 8
0xbd21: jmp    0xbd2d
0xbd2d:
0xbd23: nop    dword ptr [rax + rax]
0xbd28: mov    rdi, qword ptr [r13 + rbx*8]
0xbd2d: mov    rsi, rbp
0xbd30: mov    r12d, ebx
0xbd33: call   0x4b30
0xbd38: test   eax, eax
0xbd3a: je     0xbd60
0xbd3c: add    rbx, 1
0xbd40: cmp    rbx, 0xa
0xbd44: jne    0xbd28
0xbd46: add    rsp, 8
0xbd4a: mov    r12d, 0xffffffff
0xbd50: pop    rbx
0xbd51: mov    eax, r12d
0xbd54: pop    rbp
0xbd55: pop    r12
0xbd57: pop    r13
0xbd59: ret
0xbd5a: nop    word ptr [rax + rax]
0xbd60: lea    eax, [rbx - 8]
0xbd63: cmp    ebx, 7
0xbd66: cmovg  r12d, eax
0xbd6a: add    rsp, 8
0xbd6e: pop    rbx
0xbd6f: pop    rbp
0xbd70: mov    eax, r12d
0xbd73: pop    r12
0xbd75: pop    r13
0xbd77: ret
0xbd78: nop    dword ptr [rax + rax]
0xbd80: endbr64
0xbd84: push   r12
0xbd86: mov    r12, rdi
0xbd89: push   rbp
0xbd8a: lea    rbp, [rip + 0x1e1e7]
0xbd91: push   rbx
0xbd92: mov    ebx, 0x11
0xbd97: jmp    0xbdac
0xbdac:
0xbd99: nop    dword ptr [rax]
0xbda0: add    ebx, 1
0xbda3: add    rbp, 0x20
0xbda7: cmp    ebx, 0x26
0xbdaa: je     0xbdd8
0xbdac: mov    rdi, qword ptr [rbp]

```

```

0xbdb0: mov     rsi, r12
0xbdb3: call    0x4b30
0xbdb8: test    eax, eax
0xbdba: jne     0xbda0
0xbdbc: movsxd  rbx, ebx
0xbdbf: lea     rax, [rip + 0x1df7a]
0xbdc6: shl     rbx, 5
0xbdca: mov     rax, qword ptr [rax + rbx + 8]
0xbdcf: pop     rbx
0xbdd0: pop     rbp
0xbdd1: pop     r12
0xbdd3: ret
0xbdd4: nop     dword ptr [rax]
0xbdd8: pop     rbx
0xbdd9: xor     eax, eax
0xbddb: pop     rbp
0xbddc: pop     r12
0xbdde: ret
0xbddf: nop
0xbde0: endbr64
0xbde4: push    r14
0xbde6: lea     rdi, [rip + 0x1479d]
0xbded: push    r13
0xbdef: push    r12
0xbdf1: push    rbp
0xbdf2: push    rbx
0xbdf3: sub     rsp, 0xb0
0xbdfa: mov     rax, qword ptr fs:[0x28]
0xbe03: mov     qword ptr [rsp + 0xa8], rax
0xbe0b: xor     eax, eax
0xbe0d: call    0x47e0
0xbe12: lea     rsi, [rsp + 0x10]
0xbe17: mov     edi, 1
0xbe1c: mov     rbp, rax
0xbe1f: call    0x4eb0
0xbe24: mov     rax, qword ptr [rsp + 0x38]
0xbe29: shr     rax, 8
0xbe2d: cmp     rax, 4
0xbe31: sete    al
0xbe34: movzx   eax, al
0xbe37: mov     dword ptr [rip + 0x1eb4b], eax
0xbe3d: test    rbp, rbp
0xbe40: je      0xc36d
0xbe46: mov     rdi, rbp
0xbe49: call    0x49b0
0xbe4e: cmp     rax, 0x3f
0xbe52: ja      0xc308
0xbe58: lea     r12, [rip + 0x14ff1]

```



```

0xbe5f: lea     rbx, [rip + 0x14d6f]
0xbe66: lea     r14, [rip + 0x14e02]
0xbe6d: lea     r13, [rip + 0x14e07]
0xbe74: nop     dword ptr [rax]
0xbe78: xor     edi, edi
0xbe7a: mov     rsi, rbp
0xbe7d: call    0x4e90
0xbe82: cmp     eax, -1
0xbe85: je      0xbf38
0xbe8b: test    eax, eax
0xbe8d: jne     0xbff8
0xbe93: mov     rdi, qword ptr [rip + 0x1e9e6]
0xbe9a: mov     esi, 1
0xbe9f: mov     r8, rbp
0xbea2: mov     rdx, r12
0xbea5: mov     rcx, qword ptr [rip + 0x1ea3c]
0xbeac: lea     r9, [rip + 0x1de5d]
0xeb3: call    0x4e60
0xeb8: mov     rsi, rbx
0xebb: mov     rdi, rbp
0xebe: call    0x4b30
0xbec3: test    eax, eax
0xbec5: je      0xbf98
0xbecb: mov     rdi, rbp
0xbece: call    0x9940
0xbed3: test    eax, eax
0xbed5: jne     0xbfc8
0xbedb: mov     rsi, r14
0xbede: mov     rdi, rbp
0xbee1: call    0x4b30
0xbee6: test    eax, eax
0xbee8: je      0xbf01
0xbeea: lea     rsi, [rip + 0x14d84]
0xbef1: mov     rdi, rbp
0xbef4: call    0x4b30
0xbef9: test    eax, eax
0xbefb: jne     0xc3b4
0xbf01: mov     rdi, qword ptr [rip + 0x1e978]
0xbf08: mov     rcx, qword ptr [rip + 0x1e9d9]
0xbf0f: mov     esi, 1
0xbf14: xor     eax, eax
0xbf16: mov     rdx, r13
0xbf19: lea     rbp, [rip + 0x14cba]
0xbf20: call    0x4e60
0xbf25: xor     edi, edi
0xbf27: mov     rsi, rbp
0xbf2a: call    0x4e90
0xbf2f: cmp     eax, -1

```

```

0xbf32: jne      0xbe8b
0xbf38: mov     rcx, qword ptr [rip + 0x1e9a9]
0xbf3f: mov     rdi, qword ptr [rip + 0x1e93a]
0xbf46: xor     eax, eax
0xbf48: lea     r8, [rip + 0x1ddc1]
0xbf4f: lea     rdx, [rip + 0x14ea2]
0xbf56: mov     esi, 1
0xbf5b: call    0x4e60
0xbf60: mov     rdi, qword ptr [rip + 0x1e919]
0xbf67: xor     eax, eax
0xbf69: mov     rcx, qword ptr [rip + 0x1e978]
0xbf70: lea     r8, [rip + 0x1dd89]
0xbf77: lea     rdx, [rip + 0x14ea2]
0xbf7e: mov     esi, 1
0xbf83: call    0x4e60
0xbf88: mov     edi, 1
0xbf8d: call    0x4e20
0xbf92: nop     word ptr [rax + rax]
0xbf98: mov     rcx, qword ptr [rip + 0x1e949]
0xbf9f: mov     rdi, qword ptr [rip + 0x1e8da]
0xbfa6: lea     rdx, [rip + 0x14c99]
0xbfad: mov     esi, 1
0xbfb2: lea     rbp, [rip + 0x14c21]
0xbfb9: call    0x4e60
0xbfbe: jmp     0xbe78
0xbe78:
0xbfc3: nop     dword ptr [rax + rax]
0xbfc8: mov     rcx, qword ptr [rip + 0x1e919]
0xbfcf: mov     rdi, qword ptr [rip + 0x1e8aa]
0xbfd6: lea     rdx, [rip + 0x14c7d]
0xbfdd: xor     eax, eax
0xbfdf: mov     esi, 1
0xbfe4: lea     rbp, [rip + 0x14bf6]
0xbfeb: call    0x4e60
0xbff0: jmp     0xbe78
0xbe78:
0xbff5: nop     dword ptr [rax]
0xbff8: mov     rdi, rbp
0xbffb: lea     r13, [rsp + 8]
0xc000: call    0x1d7a0
0xc005: mov     edi, 0x800
0xc00a: mov     qword ptr [rip + 0x1e93f], rax
0xc011: call    0x1d670
0xc016: mov     rdi, qword ptr [rip + 0x1de03]
0xc01d: mov     rsi, r13
0xc020: mov     qword ptr [rsp + 8], rax
0xc025: call    0x4ca0
0xc02a: xor     edx, edx

```

```

0xc02c: test    rax, rax
0xc02f: je      0xc034
0xc031: movzx   edx, byte ptr [rax]
0xc034: mov     rdi, qword ptr [rip + 0x1de25]
0xc03b: mov     rsi, r13
0xc03e: mov     byte ptr [rip + 0x1e904], dl
0xc044: call    0x4ca0
0xc049: mov     rdi, qword ptr [rip + 0x1ddf0]
0xc050: mov     rsi, r13
0xc053: mov     qword ptr [rip + 0x1e8e6], rax
0xc05a: call    0x4ca0
0xc05f: cmp     qword ptr [rip + 0x1e8d9], 0
0xc067: mov     qword ptr [rip + 0x1e8ca], rax
0xc06e: je      0xc2c0
0xc074: test    rax, rax
0xc077: je      0xc2c0
0xc07d: mov     rdi, qword ptr [rip + 0x1dcbc]
0xc084: mov     rsi, r13
0xc087: call    0x4ca0
0xc08c: mov     rdi, qword ptr [rip + 0x1dced]
0xc093: mov     rsi, r13
0xc096: mov     qword ptr [rip + 0x1dcab], rax
0xc09d: call    0x4ca0
0xc0a2: mov     rdi, qword ptr [rip + 0x1dcb7]
0xc0a9: mov     rsi, r13
0xc0ac: mov     qword ptr [rip + 0x1dcd5], rax
0xc0b3: call    0x4ca0
0xc0b8: cmp     qword ptr [rip + 0x1dc88], 0
0xc0c0: mov     qword ptr [rip + 0x1dca1], rax
0xc0c7: je      0xc352
0xc0cd: mov     rdi, qword ptr [rip + 0x1ddac]
0xc0d4: mov     rsi, r13
0xc0d7: call    0x4ca0
0xc0dc: xor     edx, edx
0xc0de: mov     qword ptr [rip + 0x1dda3], rax
0xc0e5: test    rax, rax
0xc0e8: je      0xc0ff
0xc0ea: mov     rdi, qword ptr [rip + 0x1ddcf]
0xc0f1: mov     qword ptr [rip + 0x1dc70], rax
0xc0f8: call    0x4c90
0xc0fd: mov     edx, eax
0xc0ff: mov     rdi, qword ptr [rip + 0x1dd9a]
0xc106: mov     dword ptr [rip + 0x1ddc4], edx
0xc10c: call    0x4c50
0xc111: mov     dword ptr [rip + 0x1dd99], eax
0xc117: test    eax, eax
0xc119: js      0xc13c
0xc11b: mov     qword ptr [rip + 0x1dc62], 0

```

```

0xc126: mov     qword ptr [rip + 0x1dc37], 0
0xc131: mov     qword ptr [rip + 0x1dc0c], 0
0xc13c: mov     rdi, qword ptr [rip + 0x1dd9d]
0xc143: call    0x4c50
0xc148: mov     rdi, qword ptr [rip + 0x1ddb1]
0xc14f: mov     dword ptr [rip + 0x1dd9b], eax
0xc155: call    0x4c50
0xc15a: mov     rdi, qword ptr [rip + 0x1dc3f]
0xc161: mov     rsi, r13
0xc164: mov     dword ptr [rip + 0x1dda6], eax
0xc16a: call    0x4ca0
0xc16f: mov     rdi, qword ptr [rip + 0x1dc4a]
0xc176: mov     rsi, r13
0xc179: mov     qword ptr [rip + 0x1dc28], rax
0xc180: call    0x4ca0
0xc185: cmp     qword ptr [rip + 0x1dc1b], 0
0xc18d: mov     qword ptr [rip + 0x1dc34], rax
0xc194: je      0xc2a0
0xc19a: test    rax, rax
0xc19d: je      0xc2a0
0xc1a3: mov     rdi, qword ptr [rip + 0x1dc36]
0xc1aa: mov     rsi, r13
0xc1ad: lea     rbp, [rip + 0x1ddac]
0xc1b4: lea     rbx, [rbp - 0x220]
0xc1bb: call    0x4ca0
0xc1c0: mov     rdi, qword ptr [rip + 0x1dc39]
0xc1c7: mov     rsi, r13
0xc1ca: lea     r12, [rbx + 0x4c0]
0xc1d1: mov     qword ptr [rip + 0x1dc10], rax
0xc1d8: call    0x4ca0
0xc1dd: mov     rdi, qword ptr [rip + 0x1dd3c]
0xc1e4: mov     rsi, r13
0xc1e7: mov     qword ptr [rip + 0x1dc1a], rax
0xc1ee: call    0x4ca0
0xc1f3: mov     rdi, qword ptr [rip + 0x1dd46]
0xc1fa: mov     rsi, r13
0xc1fd: mov     qword ptr [rip + 0x1dd24], rax
0xc204: call    0x4ca0
0xc209: mov     qword ptr [rip + 0x1dd38], rax
0xc210: mov     rdi, qword ptr [rbp]
0xc214: mov     rsi, r13
0xc217: add     rbp, 0x20
0xc21b: call    0x4ca0
0xc220: mov     qword ptr [rbp - 0x18], rax
0xc224: cmp     rbp, r12
0xc227: jne     0xc210
0xc229: xor     ebp, ebp
0xc22b: lea     r13, [rip + 0x161b3]

```

```

0xc232: lea      r14, [rip + 0x14c57]
0xc239: jmp      0xc249
0xc249:
0xc23b: nop      dword ptr [rax + rax]
0xc240: add      rbx, 0x20
0xc244: cmp      rbx, r12
0xc247: je       0xc268
0xc249: cmp      qword ptr [rbx + 8], 0
0xc24e: jne      0xc240
0xc250: mov      eax, dword ptr [rbx + 0x14]
0xc253: test     eax, eax
0xc255: jne      0xc2e0
0xc25b: mov      qword ptr [rbx + 8], r13
0xc25f: add      rbx, 0x20
0xc263: cmp      rbx, r12
0xc266: jne      0xc249
0xc268: test     ebp, ebp
0xc26a: jne      0xc390
0xc270: mov      rax, qword ptr [rsp + 0xa8]
0xc278: sub      rax, qword ptr fs:[0x28]
0xc281: jne      0xc3be
0xc287: add      rsp, 0xb0
0xc28e: pop      rbx
0xc28f: pop      rbp
0xc290: pop      r12
0xc292: pop      r13
0xc294: pop      r14
0xc296: ret
0xc297: nop      word ptr [rax + rax]
0xc2a0: mov      qword ptr [rip + 0x1dafd], 0
0xc2ab: mov      qword ptr [rip + 0x1db12], 0
0xc2b6: jmp      0xc1a3
0xc1a3:
0xc2bb: nop      dword ptr [rax + rax]
0xc2c0: mov      qword ptr [rip + 0x1e66d], 0
0xc2cb: mov      qword ptr [rip + 0x1e66a], 0
0xc2d6: jmp      0xc07d
0xc07d:
0xc2db: nop      dword ptr [rax + rax]
0xc2e0: mov      r8, qword ptr [rbx]
0xc2e3: mov      rcx, qword ptr [rip + 0x1e5fe]
0xc2ea: mov      rdx, r14
0xc2ed: xor      eax, eax
0xc2ef: mov      rdi, qword ptr [rip + 0x1e58a]
0xc2f6: mov      esi, 1
0xc2fb: add      ebp, 1
0xc2fe: call    0x4e60
0xc303: jmp      0xc240

```

```

0xc240
0xc308: mov     rcx, qword ptr [rip + 0x1e5d9]
0xc30f: mov     rdi, qword ptr [rip + 0x1e56a]
0xc316: lea     rdx, [rip + 0x14aab]
0xc31d: xor     eax, eax
0xc31f: mov     esi, 1
0xc324: call    0x4e60
0xc329: mov     rcx, qword ptr [rip + 0x1e550]
0xc330: mov     edx, 0x11
0xc335: mov     esi, 1
0xc33a: lea     rdi, [rip + 0x148f3]
0xc341: lea     rbp, [rip + 0x1d9d1]
0xc348: call    0x4e40
0xc34d: jmp     0xbe58
0xbe58
0xc352: mov     qword ptr [rip + 0x1da2b], 0
0xc35d: mov     qword ptr [rip + 0x1da00], 0
0xc368: jmp     0xc0cd
0xc0cd
0xc36d: mov     rcx, qword ptr [rip + 0x1e574]
0xc374: mov     rdi, qword ptr [rip + 0x1e505]
0xc37b: lea     rdx, [rip + 0x14a16]
0xc382: xor     eax, eax
0xc384: mov     esi, 1
0xc389: call    0x4e60
0xc38e: jmp     0xc329
0xc329
0xc390: mov     rcx, qword ptr [rip + 0x1e551]
0xc397: mov     rdi, qword ptr [rip + 0x1e4e2]
0xc39e: mov     r8d, ebp
0xc3a1: xor     eax, eax
0xc3a3: lea     rdx, [rip + 0x14b16]
0xc3aa: mov     esi, 1
0xc3af: call    0x4e60
0xc3b4: mov     edi, 1
0xc3b9: call    0x4e20
0xc3be: call    0x49d0
0xc3c3: nop     word ptr cs:[rax + rax]
0xc3ce: nop
0xc3d0: endbr64
0xc3d4: push    rbx
0xc3d5: mov     ebx, edi
0xc3d7: xor     edi, edi
0xc3d9: call    0x48f0
0xc3de: test    eax, eax
0xc3e0: je      0xc453
0xc3e2: mov     edi, 1
0xc3e7: call    0x48f0

```

```

0xc3ec: test    eax, eax
0xc3ee: je         0xc453
0xc3f0: mov       edi, 1
0xc3f5: call      0x4cd0
0xc3fa: mov       qword ptr [rip + 0x1e58f], rax
0xc401: test     rax, rax
0xc404: je        0xc47e
0xc406: lea       rsi, [rip + 0x1efb3]
0xc40d: mov       edi, 1
0xc412: call      0x4d60
0xc417: mov       edi, 0x40
0xc41c: call      0x1d670
0xc421: mov       dword ptr [rip + 0x1efd5], ebx
0xc427: pop       rbx
0xc428: mov       qword ptr [rip + 0x1ef31], rax
0xc42f: mov       qword ptr [rip + 0x1e52a], rax
0xc436: xor       eax, eax
0xc438: mov       qword ptr [rip + 0x1e52d], 0
0xc443: mov       qword ptr [rip + 0x1e51a], 0
0xc44e: jmp       0xbde0
0xbde0
0xc453: mov       rdi, qword ptr [rip + 0x1e426]
0xc45a: mov       rcx, qword ptr [rip + 0x1e487]
0xc461: lea       rdx, [rip + 0x14a88]
0xc468: xor       eax, eax
0xc46a: mov       esi, 1
0xc46f: call      0x4e60
0xc474: mov       edi, 1
0xc479: call      0x4e20
0xc47e: mov       rdi, qword ptr [rip + 0x1e3fb]
0xc485: mov       rcx, qword ptr [rip + 0x1e45c]
0xc48c: mov       esi, 1
0xc491: lea       rdx, [rip + 0x147f8]
0xc498: call      0x4e60
0xc49d: mov       edi, 1
0xc4a2: call      0x4e20
0xc4a7: nop       word ptr [rax + rax]
0xc4b0: endbr64
0xc4b4: push     r12
0xc4b6: push     rbp
0xc4b7: mov       rbp, rdi
0xc4ba: mov       rdi, qword ptr [rip + 0x1e48f]
0xc4c1: push     rbx
0xc4c2: call      0x9940
0xc4c7: test     eax, eax
0xc4c9: jne       0xc4d0
0xc4cb: pop       rbx
0xc4cc: pop       rbp

```

```

0xc4cd: pop     r12
0xc4cf: ret
0xc4d0: mov     rdi, rbp
0xc4d3: call    0x49b0
0xc4d8: lea     rdi, [rax + 0x81]
0xc4df: mov     rbx, rax
0xc4e2: call    0x1d670
0xc4e7: mov     rdi, rbp
0xc4ea: mov     r12, rax
0xc4ed: call    0x1d7a0
0xc4f2: mov     rsi, rbx
0xc4f5: add     ebx, 0x7f
0xc4f8: mov     rbp, rax
0xc4fb: mov     rdi, rax
0xc4fe: call    0x184d0
0xc503: push    7
0xc505: lea     rcx, [rip + 0x147a2]
0xc50c: xor     eax, eax
0xc50e: push    rbp
0xc50f: mov     rdx, -1
0xc516: mov     esi, 1
0xc51b: mov     rdi, r12
0xc51e: lea     r9, [rip + 0x14040]
0xc525: mov     r8d, 0x1b
0xc52b: call    0x4f10
0xc530: mov     eax, dword ptr [rip + 0x1e462]
0xc536: pop     rdx
0xc537: pop     rcx
0xc538: add     eax, eax
0xc53a: cmp     ebx, eax
0xc53c: jl      0xc555
0xc53e: cdqe
0xc540: mov     byte ptr [r12 + rax], 7
0xc545: mov     eax, dword ptr [rip + 0x1e44d]
0xc54b: add     eax, eax
0xc54d: cdqe
0xc54f: mov     byte ptr [r12 + rax + 1], 0
0xc555: mov     rdi, r12
0xc558: call    0x49b0
0xc55d: mov     rsi, r12
0xc560: mov     edi, 1
0xc565: mov     rdx, rax
0xc568: call    0x18da0
0xc56d: mov     rdi, rbp
0xc570: call    0x18d10
0xc575: mov     rdi, r12
0xc578: call    0x18d10
0xc57d: mov     rdi, qword ptr [rip + 0x1e2bc]

```



```

0xc584: pop      rbx
0xc585: pop      rbp
0xc586: pop      r12
0xc588: jmp      0x4c40
0x4c40
0xc58d: nop      dword ptr [rax]
0xc590: endbr64
0xc594: sub      rsp, 8
0xc598: mov      edi, 0x18
0xc59d: call     0x1d670
0xc5a2: mov      qword ptr [rax], 0
0xc5a9: mov      qword ptr [rax + 8], 0
0xc5b1: add      rsp, 8
0xc5b5: ret
0xc5b6: nop      word ptr cs:[rax + rax]
0xc5c0: endbr64
0xc5c4: test     rdi, rdi
0xc5c7: je       0xc5d0
0xc5c9: jmp      0x18d10
0x18d10
0xc5ce: nop
0xc5d0: ret
0xc5d1: nop      word ptr cs:[rax + rax]
0xc5dc: nop      dword ptr [rax]
0xc5e0: endbr64
0xc5e4: mov      dword ptr [rdi], esi
0xc5e6: mov      dword ptr [rdi + 4], edx
0xc5e9: mov      dword ptr [rdi + 8], ecx
0xc5ec: mov      dword ptr [rdi + 0xc], r8d
0xc5f0: ret
0xc5f1: nop      word ptr cs:[rax + rax]
0xc5fc: nop      dword ptr [rax]
0xc600: endbr64
0xc604: mov      rax, rdi
0xc607: mov      rdi, rsi
0xc60a: mov      esi, edx
0xc60c: mov      ecx, dword ptr [rax + 0x10]
0xc60f: lea      r8d, [rcx + rdx]
0xc613: mov      edx, dword ptr [rax + 0xc]
0xc616: mov      dword ptr [rax + 0x10], r8d
0xc61a: cmp      edx, ecx
0xc61c: jle      0xc640
0xc61e: mov      r9d, dword ptr [rax + 8]
0xc622: cmp      dword ptr [rax + 0x14], r9d
0xc626: jge      0xc640
0xc628: cmp      r8d, edx
0xc62b: jle      0xc631
0xc62d: sub      edx, ecx

```

```

0xc62f: mov     esi, edx
0xc631: jmp     0xa1a0
0xa1a0
0xc636: nop     word ptr cs:[rax + rax]
0xc640: xor     eax, eax
0xc642: ret
0xc643: nop     word ptr cs:[rax + rax]
0xc64e: nop
0xc650: endbr64
0xc654: mov     ecx, dword ptr [rdi + 0x10]
0xc657: lea     edx, [rcx + 1]
0xc65a: mov     dword ptr [rdi + 0x10], edx
0xc65d: cmp     edx, dword ptr [rdi + 0xc]
0xc660: jg      0xc66a
0xc662: mov     ecx, dword ptr [rdi + 8]
0xc665: cmp     dword ptr [rdi + 0x14], ecx
0xc668: jl      0xc670
0xc66a: xor     eax, eax
0xc66c: ret
0xc66d: nop     dword ptr [rax]
0xc670: mov     edi, esi
0xc672: jmp     0xa230
0xa230
0xc677: nop     word ptr [rax + rax]
0xc680: endbr64
0xc684: mov     rax, rdi
0xc687: mov     edi, esi
0xc689: mov     dword ptr [rax + 0x10], edx
0xc68c: add     edx, dword ptr [rax]
0xc68e: mov     dword ptr [rax + 0x14], esi
0xc691: add     edi, dword ptr [rax + 4]
0xc694: mov     esi, edx
0xc696: jmp     0xa4b0
0xa4b0
0xc69b: nop     dword ptr [rax + rax]
0xc6a0: endbr64
0xc6a4: mov     eax, dword ptr [rdi]
0xc6a6: ret
0xc6a7: nop     word ptr [rax + rax]
0xc6b0: endbr64
0xc6b4: mov     eax, dword ptr [rdi + 4]
0xc6b7: ret
0xc6b8: nop     dword ptr [rax + rax]
0xc6c0: endbr64
0xc6c4: mov     eax, dword ptr [rdi + 8]
0xc6c7: ret
0xc6c8: nop     dword ptr [rax + rax]
0xc6d0: endbr64

```

```

0xc6d4: mov     eax, dword ptr [rdi + 0xc]
0xc6d7: ret
0xc6d8: nop     dword ptr [rax + rax]
0xc6e0: push    r14
0xc6e2: push    r13
0xc6e4: push    r12
0xc6e6: mov     r12, qword ptr [rip + 0x1ed1b]
0xc6ed: push    rbp
0xc6ee: push    rbx
0xc6ef: mov     r13, qword ptr [r12 + 0x20]
0xc6f4: mov     rbx, qword ptr [r12 + 0x18]
0xc6f9: cmp     r13, rbx
0xc6fc: je      0xc761
0xc6fe: mov     ebp, edi
0xc700: jbe     0xc770
0xc702: test    bpl, 1
0xc706: je      0xc75c
0xc708: mov     rdi, qword ptr [r12 + 0x60]
0xc70d: test    rdi, rdi
0xc710: je      0xc71e
0xc712: call    0x18d10
0xc717: mov     r12, qword ptr [rip + 0x1ecea]
0xc71e: mov     r14, r13
0xc721: sub     r14, rbx
0xc724: lea     rdi, [r14 + 1]
0xc728: call    0x1d670
0xc72d: mov     rdx, r14
0xc730: mov     qword ptr [r12 + 0x60], rax
0xc735: mov     rax, qword ptr [rip + 0x1eccc]
0xc73c: mov     rsi, qword ptr [rax + 0x58]
0xc740: mov     rdi, qword ptr [rax + 0x60]
0xc744: add     rsi, rbx
0xc747: call    0x4bb0
0xc74c: mov     rax, qword ptr [rip + 0x1ecb5]
0xc753: mov     rax, qword ptr [rax + 0x60]
0xc757: mov     byte ptr [rax + r14], 0
0xc75c: and     ebp, 2
0xc75f: jne     0xc788
0xc761: pop     rbx
0xc762: pop     rbp
0xc763: pop     r12
0xc765: pop     r13
0xc767: pop     r14
0xc769: ret
0xc76a: nop     word ptr [rax + rax]
0xc770: mov     rax, rbx
0xc773: mov     rbx, r13
0xc776: mov     r13, rax

```

```

0xc779: test    bpl, 1
0xc77d: je      0xc75c
0xc77f: jmp     0xc708
0xc708:
0xc781: nop     dword ptr [rax]
0xc788: mov     rax, qword ptr [rip + 0x1ec79]
0xc78f: mov     rdx, rbx
0xc792: sub     rdx, r13
0xc795: add     qword ptr [rax + 0x48], rdx
0xc799: mov     rdi, qword ptr [rax + 0x58]
0xc79d: add     qword ptr [rax + 0x38], rdx
0xc7a1: mov     qword ptr [rax + 0x18], rbx
0xc7a5: lea     rsi, [rdi + r13]
0xc7a9: add     rdi, rbx
0xc7ac: mov     qword ptr [rax + 0x20], rbx
0xc7b0: call    0x48b0
0xc7b5: mov     rbx, qword ptr [rip + 0x1ec4c]
0xc7bc: mov     rax, qword ptr [rbx + 0x38]
0xc7c0: mov     rdi, qword ptr [rbx + 0x58]
0xc7c4: lea     rsi, [rax + 1]
0xc7c8: mov     qword ptr [rbx + 0x50], rsi
0xc7cc: call    0x1d690
0xc7d1: mov     qword ptr [rbx + 0x58], rax
0xc7d5: pop     rbx
0xc7d6: pop     rbp
0xc7d7: pop     r12
0xc7d9: pop     r13
0xc7db: pop     r14
0xc7dd: ret
0xc7de: nop
0xc7e0: endbr64
0xc7e4: test    rdi, rdi
0xc7e7: je      0xc818
0xc7e9: push    rbp
0xc7ea: mov     rbp, rdi
0xc7ed: mov     rdi, qword ptr [rdi + 0x58]
0xc7f1: test    rdi, rdi
0xc7f4: je      0xc7fb
0xc7f6: call    0x18d10
0xc7fb: mov     rdi, qword ptr [rbp + 0x60]
0xc7ff: test    rdi, rdi
0xc802: je      0xc809
0xc804: call    0x18d10
0xc809: mov     rdi, rbp
0xc80c: pop     rbp
0xc80d: jmp     0x18d10
0x18d10:
0xc812: nop     word ptr [rax + rax]

```

```

0xc818: ret
0xc819: nop      dword ptr [rax]
0xc820: endbr64
0xc824: mov      rax, qword ptr [rip + 0x1ebdd]
0xc82b: push     rbp
0xc82c: mov      rdi, qword ptr [rax + 0x68]
0xc830: xor      eax, eax
0xc832: call     0x1aac0
0xc837: mov      rax, qword ptr [rip + 0x1ebca]
0xc83e: mov      rdi, qword ptr [rax]
0xc841: call     0xc5c0
0xc846: mov      rbp, qword ptr [rip + 0x1ebbb]
0xc84d: test     rbp, rbp
0xc850: je       0xc876
0xc852: mov      rdi, qword ptr [rbp + 0x58]
0xc856: test     rdi, rdi
0xc859: je       0xc860
0xc85b: call     0x18d10
0xc860: mov      rdi, qword ptr [rbp + 0x60]
0xc864: test     rdi, rdi
0xc867: je       0xc86e
0xc869: call     0x18d10
0xc86e: mov      rdi, rbp
0xc871: call     0x18d10
0xc876: mov      qword ptr [rip + 0x1eb87], 0
0xc881: pop      rbp
0xc882: ret
0xc883: nop      word ptr cs:[rax + rax]
0xc88e: nop
0xc890: endbr64
0xc894: mov      rax, qword ptr [rip + 0x1eb6d]
0xc89b: movsxd   rcx, edi
0xc89e: mov      edx, esi
0xc8a0: mov      r8, rcx
0xc8a3: mov      qword ptr [rax + 0x28], rcx
0xc8a7: movsxd   rcx, esi
0xc8aa: mov      rdi, qword ptr [rax]
0xc8ad: xor      esi, esi
0xc8af: mov      qword ptr [rax + 0x30], rcx
0xc8b3: mov      ecx, 1
0xc8b8: jmp      0xc5e0
0xc5e0:
0xc8bd: nop      dword ptr [rax]
0xc8c0: endbr64
0xc8c4: push     r12
0xc8c6: mov      edi, 0x70
0xc8cb: push     rbx
0xc8cc: sub      rsp, 8

```

```

0xc8d0: call    0x1d670
0xc8d5: mov     rbx, qword ptr [rip + 0x1eb2c]
0xc8dc: mov     r12, rax
0xc8df: movdqu  xmm0, xmmword ptr [rbx]
0xc8e3: movups  xmmword ptr [rax], xmm0
0xc8e6: movdqu  xmm1, xmmword ptr [rbx + 0x10]
0xc8eb: movups  xmmword ptr [rax + 0x10], xmm1
0xc8ef: movdqu  xmm2, xmmword ptr [rbx + 0x20]
0xc8f4: movups  xmmword ptr [rax + 0x20], xmm2
0xc8f8: movdqu  xmm3, xmmword ptr [rbx + 0x30]
0xc8fd: movups  xmmword ptr [rax + 0x30], xmm3
0xc901: movdqu  xmm4, xmmword ptr [rbx + 0x40]
0xc906: movups  xmmword ptr [rax + 0x40], xmm4
0xc90a: movdqu  xmm5, xmmword ptr [rbx + 0x50]
0xc90f: movups  xmmword ptr [rax + 0x50], xmm5
0xc913: movdqu  xmm6, xmmword ptr [rbx + 0x60]
0xc918: cmp     qword ptr [rax + 0x58], 0
0xc91d: movups  xmmword ptr [rax + 0x60], xmm6
0xc921: je      0xc94a
0xc923: mov     rdi, qword ptr [rax + 0x50]
0xc927: call    0x1d670
0xc92c: mov     rdx, qword ptr [r12 + 0x50]
0xc931: mov     qword ptr [rbx + 0x58], rax
0xc935: mov     rax, qword ptr [rip + 0x1eacc]
0xc93c: mov     rsi, qword ptr [r12 + 0x58]
0xc941: mov     rdi, qword ptr [rax + 0x58]
0xc945: call    0x4bb0
0xc94a: mov     rdi, qword ptr [r12 + 0x60]
0xc94f: test    rdi, rdi
0xc952: je      0xc964
0xc954: mov     rbx, qword ptr [rip + 0x1eaad]
0xc95b: call    0x1d7a0
0xc960: mov     qword ptr [rbx + 0x60], rax
0xc964: add     rsp, 8
0xc968: mov     rax, r12
0xc96b: pop     rbx
0xc96c: pop     r12
0xc96e: ret
0xc96f: nop
0xc970: endbr64
0xc974: test    rdi, rdi
0xc977: je      0xc9f0
0xc979: push    r13
0xc97b: push    r12
0xc97d: push    rbp
0xc97e: push    rbx
0xc97f: mov     rbx, rdi
0xc982: sub     rsp, 8

```

```

0xc986: mov     rbp, qword ptr [rip + 0x1ea7b]
0xc98d: mov     rdi, qword ptr [rbp + 0x58]
0xc991: mov     r13, qword ptr [rbp + 0x28]
0xc995: mov     r12, qword ptr [rbp + 0x30]
0xc999: test    rdi, rdi
0xc99c: je      0xc9a3
0xc99e: call    0x18d10
0xc9a3: mov     rdi, qword ptr [rbp + 0x60]
0xc9a7: test    rdi, rdi
0xc9aa: je      0xc9b1
0xc9ac: call    0x18d10
0xc9b1: mov     rdi, rbp
0xc9b4: call    0x18d10
0xc9b9: movsxd  rax, r13d
0xc9bc: mov     edx, r12d
0xc9bf: movsxd  r12, r12d
0xc9c2: mov     qword ptr [rbx + 0x28], rax
0xc9c6: mov     rdi, qword ptr [rbx]
0xc9c9: mov     r8d, r13d
0xc9cc: mov     ecx, 1
0xc9d1: mov     qword ptr [rbx + 0x30], r12
0xc9d5: xor     esi, esi
0xc9d7: mov     qword ptr [rip + 0x1ea2a], rbx
0xc9de: add     rsp, 8
0xc9e2: pop     rbx
0xc9e3: pop     rbp
0xc9e4: pop     r12
0xc9e6: pop     r13
0xc9e8: jmp     0xc5e0
0xc5e0:
0xc9ed: nop     dword ptr [rax]
0xc9f0: ret
0xc9f1: nop     word ptr cs:[rax + rax]
0xc9fc: nop     dword ptr [rax]
0xca00: endbr64
0xca04: mov     rdx, qword ptr [rip + 0x1e9fd]
0xca0b: mov     rax, qword ptr [rdx + 0x18]
0xca0f: sub     rax, qword ptr [rdx + 0x40]
0xca13: ret
0xca14: nop     word ptr cs:[rax + rax]
0xca1f: nop
0xca20: endbr64
0xca24: mov     rax, qword ptr [rip + 0x1e9dd]
0xca2b: mov     r8d, dword ptr [rax + 8]
0xca2f: mov     dword ptr [rax + 8], edi
0xca32: mov     eax, r8d
0xca35: ret
0xca36: nop     word ptr cs:[rax + rax]

```

```

0xca40: endbr64
0xca44: mov     rax, qword ptr [rip + 0x1e9bd]
0xca4b: cmp     qword ptr [rax + 0x48], 0
0xca50: sete    al
0xca53: movzx   eax, al
0xca56: ret
0xca57: nop     word ptr [rax + rax]
0xca60: endbr64
0xca64: mov     rax, qword ptr [rip + 0x1e99d]
0xca6b: mov     rdx, qword ptr [rax + 0x18]
0xca6f: mov     qword ptr [rax + 0x20], rdx
0xca73: ret
0xca74: nop     word ptr cs:[rax + rax]
0xca7f: nop
0xca80: endbr64
0xca84: sub     rsp, 8
0xca88: xor     eax, eax
0xca8a: mov     edi, 3
0xca8f: call    0xc6e0
0xca94: mov     rax, qword ptr [rip + 0x1e96d]
0xca9b: mov     dword ptr [rax + 0x10], 0x18
0xcaa2: add     rsp, 8
0xcaa6: ret
0xcaa7: nop     word ptr [rax + rax]
0xcab0: endbr64
0xcab4: sub     rsp, 8
0xcab8: xor     eax, eax
0xcaba: mov     edi, 1
0xcabf: call    0xc6e0
0xcac4: mov     rax, qword ptr [rip + 0x1e93d]
0xcacb: mov     dword ptr [rax + 0x10], 0x19
0xcad2: add     rsp, 8
0xcad6: ret
0xcad7: nop     word ptr [rax + rax]
0xcae0: endbr64
0xcae4: mov     rax, qword ptr [rip + 0x1e91d]
0xcaeb: mov     rdx, qword ptr [rax + 0x18]
0xcaef: mov     rcx, qword ptr [rax + 0x20]
0xcaf3: movsxd  rdx, edx
0xcaf6: mov     qword ptr [rax + 0x18], rcx
0xcafa: mov     qword ptr [rax + 0x20], rdx
0xcafe: ret
0xcaff: nop
0xcb00: endbr64
0xcb04: mov     rax, qword ptr [rip + 0x1e8fd]
0xcb0b: mov     rdx, qword ptr [rax + 0x18]
0xcb0f: cmp     rdx, qword ptr [rax + 0x40]
0xcb13: jbe     0xcb24

```



```

0xcb15: sub     rdx, 1
0xcb19: mov     dword ptr [rax + 0x10], 1
0xcb20: mov     qword ptr [rax + 0x18], rdx
0xcb24: ret
0xcb25: nop     word ptr cs:[rax + rax]
0xcb30: endbr64
0xcb34: mov     rax, qword ptr [rip + 0x1e8cd]
0xcb3b: mov     rdx, qword ptr [rax + 0x18]
0xcb3f: cmp     rdx, qword ptr [rax + 0x38]
0xcb43: jae     0xcb54
0xcb45: add     rdx, 1
0xcb49: mov     dword ptr [rax + 0x10], 2
0xcb50: mov     qword ptr [rax + 0x18], rdx
0xcb54: ret
0xcb55: nop     word ptr cs:[rax + rax]
0xcb60: endbr64
0xcb64: mov     rdx, qword ptr [rip + 0x1e89d]
0xcb6b: mov     rax, qword ptr [rdx + 0x18]
0xcb6f: mov     rdi, qword ptr [rdx + 0x40]
0xcb73: cmp     rax, rdi
0xcb76: jbe     0xcc20
0xcb7c: mov     r8, qword ptr [rdx + 0x58]
0xcb80: jmp     0xcba9
0xcba9:
0xcb82: nop     word ptr [rax + rax]
0xcb88: lea     ecx, [rsi - 0x30]
0xcb8b: cmp     cl, 9
0xcb8e: jbe     0xcbbf
0xcb90: sub     esi, 0x24
0xcb93: cmp     sil, 1
0xcb97: jbe     0xcbbf
0xcb99: mov     qword ptr [rdx + 0x18], rax
0xcb9d: mov     dword ptr [rdx + 0x10], 1
0xcba4: cmp     rdi, rax
0xcba7: jae     0xcbef
0xcba9: sub     rax, 1
0xcbad: movzx   esi, byte ptr [r8 + rax]
0xcbb2: mov     ecx, esi
0xcbb4: and     ecx, 0xffffffff
0xcbb7: sub     ecx, 0x41
0xcbb8: cmp     cl, 0x19
0xcbbd: ja      0xcb88
0xcbbf: mov     rax, qword ptr [rdx + 0x18]
0xcbc3: cmp     rdi, rax
0xcbc6: jae     0xcbef
0xcbc8: sub     rax, 1
0xcbcc: movzx   esi, byte ptr [r8 + rax]
0xcbd1: mov     ecx, esi

```

```

0xcbd3: and    ecx, 0xffffffff
0xcbd6: sub    ecx, 0x41
0xcbd9: cmp    cl, 0x19
0xcbbdc: jbe    0xcc00
0xcbbde: lea    ecx, [rsi - 0x30]
0xcbe1: cmp    cl, 9
0xcbe4: jbe    0xcc00
0xcbe6: sub    esi, 0x24
0xcbe9: cmp    sil, 1
0xcbed: jbe    0xcc00
0xcbef: mov    dword ptr [rdx + 0x10], 3
0xcbf6: ret
0xcbf7: nop    word ptr [rax + rax]
0xcc00: mov    qword ptr [rdx + 0x18], rax
0xcc04: mov    dword ptr [rdx + 0x10], 1
0xcc0b: cmp    rdi, rax
0xcc0e: jb     0xcbc8
0xcc10: mov    dword ptr [rdx + 0x10], 3
0xcc17: ret
0xcc18: nop    dword ptr [rax + rax]
0xcc20: ret
0xcc21: nop    word ptr cs:[rax + rax]
0xcc2c: nop    dword ptr [rax]
0xcc30: endbr64
0xcc34: mov    rdx, qword ptr [rip + 0x1e7cd]
0xcc3b: mov    rax, qword ptr [rdx + 0x18]
0xcc3f: mov    rdi, qword ptr [rdx + 0x38]
0xcc43: cmp    rax, rdi
0xcc46: jae    0xcce8
0xcc4c: mov    r8, qword ptr [rdx + 0x58]
0xcc50: jmp    0xcc7d
0xcc7d:
0xcc52: nop    word ptr [rax + rax]
0xcc58: lea    ecx, [rsi - 0x30]
0xcc5b: cmp    cl, 9
0xcc5e: jbe    0xcc8f
0xcc60: sub    esi, 0x24
0xcc63: cmp    sil, 1
0xcc67: jbe    0xcc8f
0xcc69: add    rax, 1
0xcc6d: mov    dword ptr [rdx + 0x10], 2
0xcc74: mov    qword ptr [rdx + 0x18], rax
0xcc78: cmp    rdi, rax
0xcc7b: je     0xccbb
0xcc7d: movzx  esi, byte ptr [r8 + rax]
0xcc82: mov    ecx, esi
0xcc84: and    ecx, 0xffffffff
0xcc87: sub    ecx, 0x41

```

```

0xcc8a: cmp     cl, 0x19
0xcc8d: ja      0xcc58
0xcc8f: mov     rax, qword ptr [rdx + 0x18]
0xcc93: cmp     rdi, rax
0xcc96: jbe     0xccbb
0xcc98: movzx   esi, byte ptr [r8 + rax]
0xcc9d: mov     ecx, esi
0xcc9f: and     ecx, 0xffffffff
0xcca2: sub     ecx, 0x41
0xcca5: cmp     cl, 0x19
0xcca8: jbe     0xcc88
0xccaa: lea     ecx, [rsi - 0x30]
0xccad: cmp     cl, 9
0xccb0: jbe     0xcc88
0xccb2: sub     esi, 0x24
0xccb5: cmp     sil, 1
0xccb9: jbe     0xcc88
0xccbb: mov     dword ptr [rdx + 0x10], 4
0xcc2: ret
0xcc3: nop     dword ptr [rax + rax]
0xcc8: add     rax, 1
0xcccc: mov     dword ptr [rdx + 0x10], 2
0xcd3: mov     qword ptr [rdx + 0x18], rax
0xcd7: cmp     rdi, rax
0xcd8a: jne     0xcc98
0xcdc: mov     dword ptr [rdx + 0x10], 4
0xcce3: ret
0xcce4: nop     dword ptr [rax]
0xcce8: ret
0xcce9: nop     dword ptr [rax]
0xccf0: endbr64
0xccf4: mov     rax, qword ptr [rip + 0x1e70d]
0xccfb: mov     rdx, qword ptr [rax + 0x40]
0xccff: mov     dword ptr [rax + 0x10], 5
0xcd06: mov     qword ptr [rax + 0x18], rdx
0xcd0a: ret
0xcd0b: nop     dword ptr [rax + rax]
0xcd10: endbr64
0xcd14: mov     rax, qword ptr [rip + 0x1e6ed]
0xcd1b: mov     rdx, qword ptr [rax + 0x38]
0xcd1f: mov     dword ptr [rax + 0x10], 6
0xcd26: mov     qword ptr [rax + 0x18], rdx
0xcd2a: ret
0xcd2b: nop     dword ptr [rax + rax]
0xcd30: endbr64
0xcd34: push    rbp
0xcd35: push    rbx
0xcd36: mov     ebx, edi

```

```

0xcd38: sub     rsp, 8
0xcd3c: call    0x4f00
0xcd41: movsxd  rdx, ebx
0xcd44: mov     rax, qword ptr [rax]
0xcd47: test    byte ptr [rax + rdx*2 + 1], 0x40
0xcd4c: je      0xcdab
0xcd4e: mov     rbp, qword ptr [rip + 0x1e6b3]
0xcd55: mov     rsi, qword ptr [rbp + 0x38]
0xcd59: lea     rdx, [rsi + 1]
0xcd5d: cmp     rdx, qword ptr [rbp + 0x50]
0xcd61: jae     0xcdb8
0xcd63: mov     rax, qword ptr [rbp + 0x18]
0xcd67: mov     rcx, qword ptr [rbp + 0x58]
0xcd6b: sub     rdx, rax
0xcd6e: lea     rsi, [rcx + rax]
0xcd72: lea     rdi, [rcx + rax + 1]
0xcd77: call    0x4d40
0xcd7c: mov     rax, qword ptr [rip + 0x1e685]
0xcd83: mov     rdx, qword ptr [rax + 0x58]
0xcd87: mov     rax, qword ptr [rax + 0x18]
0xcd8b: mov     byte ptr [rdx + rax], bl
0xcd8e: mov     rax, qword ptr [rip + 0x1e673]
0xcd95: add     qword ptr [rax + 0x18], 1
0xcd9a: add     qword ptr [rax + 0x38], 1
0xcd9f: add     qword ptr [rax + 0x48], 1
0xcda4: mov     dword ptr [rax + 0x10], 7
0xcdab: add     rsp, 8
0xcdaf: pop     rbx
0xcdb0: pop     rbp
0xcdb1: ret
0xcdb2: nop     word ptr [rax + rax]
0xcdb8: add     rsi, 0x21
0xcdbc: mov     rdi, qword ptr [rbp + 0x58]
0xcdc0: mov     qword ptr [rbp + 0x50], rsi
0xcdc4: call    0x1d690
0xcdc9: mov     qword ptr [rbp + 0x58], rax
0xcdcd: mov     rbp, qword ptr [rip + 0x1e634]
0xcd4d: mov     rax, qword ptr [rbp + 0x38]
0xcd88: lea     rdx, [rax + 1]
0xcd8c: jmp     0xcd63
0xcd63:
0xcdde: nop
0xcde0: endbr64
0xcde4: mov     rax, qword ptr [rip + 0x1e61d]
0xcdeb: mov     rcx, qword ptr [rax + 0x18]
0xcdef: mov     rdx, qword ptr [rax + 0x38]
0xcdf3: cmp     rcx, rdx
0xcdf6: jbe     0xce00

```

```

0xcdf8: ret
0xcdf9: nop      dword ptr [rax]
0xce00: push     rbx
0xce01: mov      rax, qword ptr [rax + 0x58]
0xce05: add      rdx, 1
0xce09: sub      rdx, rcx
0xce0c: lea      rsi, [rax + rcx + 1]
0xce11: lea      rdi, [rax + rcx]
0xce15: call     0x4bb0
0xce1a: mov      rbx, qword ptr [rip + 0x1e5e7]
0xce21: mov      rsi, qword ptr [rbx + 0x38]
0xce25: sub      qword ptr [rbx + 0x48], 1
0xce2a: lea      rax, [rsi - 1]
0xce2e: mov      qword ptr [rbx + 0x38], rax
0xce32: test     al, 0xf
0xce34: je       0xce40
0xce36: mov      dword ptr [rbx + 0x10], 8
0xce3d: pop      rbx
0xce3e: ret
0xce3f: nop
0xce40: mov      qword ptr [rbx + 0x50], rsi
0xce44: mov      rdi, qword ptr [rbx + 0x58]
0xce48: call     0x1d690
0xce4d: mov      qword ptr [rbx + 0x58], rax
0xce51: mov      rbx, qword ptr [rip + 0x1e5b0]
0xce58: mov      dword ptr [rbx + 0x10], 8
0xce5f: pop      rbx
0xce60: ret
0xce61: nop      word ptr cs:[rax + rax]
0xce6c: nop      dword ptr [rax]
0xce70: endbr64
0xce74: mov      rax, qword ptr [rip + 0x1e58d]
0xce7b: mov      rcx, qword ptr [rax + 0x18]
0xce7f: cmp      rcx, qword ptr [rax + 0x40]
0xce83: ja       0xce90
0xce85: ret
0xce86: nop      word ptr cs:[rax + rax]
0xce90: push     rbx
0xce91: mov      rdx, qword ptr [rax + 0x38]
0xce95: mov      rsi, qword ptr [rax + 0x58]
0xce99: add      rdx, 1
0xce9d: lea      rdi, [rsi + rcx - 1]
0xcea2: sub      rdx, rcx
0xcea5: add      rsi, rcx
0xcea8: call     0x4bb0
0xcead: mov      rbx, qword ptr [rip + 0x1e554]
0xceb4: mov      rsi, qword ptr [rbx + 0x38]
0xceb8: sub      qword ptr [rbx + 0x18], 1

```

```

0xcebd: sub     qword ptr [rbx + 0x48], 1
0xcec2: lea     rax, [rsi - 1]
0xcec6: mov     qword ptr [rbx + 0x38], rax
0xceca: test    al, 0xf
0xcecc: je      0xcee0
0xcece: mov     dword ptr [rbx + 0x10], 9
0xcd5: pop     rbx
0xcd6: ret
0xcd7: nop     word ptr [rax + rax]
0xcee0: mov     qword ptr [rbx + 0x50], rsi
0xcee4: mov     rdi, qword ptr [rbx + 0x58]
0xcee8: call    0x1d690
0xceed: mov     qword ptr [rbx + 0x58], rax
0xcef1: mov     rbx, qword ptr [rip + 0x1e510]
0xcef8: mov     dword ptr [rbx + 0x10], 9
0xceff: pop     rbx
0xcf00: ret
0xcf01: nop     word ptr cs:[rax + rax]
0xcf0c: nop     dword ptr [rax]
0xcf10: endbr64
0xcf14: push    rbp
0xcf15: push    rbx
0xcf16: sub     rsp, 8
0xcf1a: mov     r9, qword ptr [rip + 0x1e4e7]
0xcf21: mov     rax, qword ptr [r9 + 0x18]
0xcf25: mov     rbx, qword ptr [r9 + 0x20]
0xcf29: mov     qword ptr [r9 + 0x20], rax
0xcf2d: xor     eax, eax
0xcf2f: call    0xcc30
0xcf34: xor     eax, eax
0xcf36: mov     edi, 3
0xcf3b: mov     rbp, qword ptr [r9 + 0x18]
0xcf3f: call    0xc6e0
0xcf44: mov     rax, qword ptr [rip + 0x1e4bd]
0xcf4b: mov     rdx, qword ptr [rax + 0x18]
0xcf4f: cmp     rdx, rbx
0xcf52: jae     0xcf65
0xcf54: lea     rcx, [rdx + rbx]
0xcf58: sub     rcx, rbp
0xcf5b: cmp     rbx, rbp
0xcf5e: cmovae  rdx, rcx
0xcf62: mov     rbx, rdx
0xcf65: mov     qword ptr [rax + 0x20], rbx
0xcf69: mov     dword ptr [rax + 0x10], 0xa
0xcf70: add     rsp, 8
0xcf74: pop     rbx
0xcf75: pop     rbp
0xcf76: ret

```

```

0xcf77: nop      word ptr [rax + rax]
0xcf80: endbr64
0xcf84: push     rbp
0xcf85: push     rbx
0xcf86: sub      rsp, 8
0xcf8a: mov      rax, qword ptr [rip + 0x1e477]
0xcf91: mov      rbp, qword ptr [rax + 0x18]
0xcf95: mov      rbx, qword ptr [rax + 0x20]
0xcf99: mov      qword ptr [rax + 0x20], rbp
0xcf9d: xor      eax, eax
0xcf9f: call     0xcb60
0xcfa4: xor      eax, eax
0xcfa6: mov      edi, 3
0xcfab: call     0xc6e0
0xcfb0: mov      rax, qword ptr [rip + 0x1e451]
0xcfb7: mov      rdx, qword ptr [rax + 0x18]
0xcfb7: mov      rdx, qword ptr [rax + 0x18]
0xcfb7: cmp      rdx, rbx
0xcfbe: jae      0xcfd1
0xcfc0: lea      rcx, [rdx + rbx]
0xcfc4: sub      rcx, rbp
0xcfc7: cmp      rbx, rbp
0xcfca: cmovae   rdx, rcx
0xcfce: mov      rbx, rdx
0xcfd1: mov      qword ptr [rax + 0x20], rbx
0xcfd5: mov      dword ptr [rax + 0x10], 0xb
0xcfdc: add      rsp, 8
0xcfe0: pop      rbx
0xcfe1: pop      rbp
0xcfe2: ret
0xcfe3: nop      word ptr cs:[rax + rax]
0xcfee: nop
0xcff0: endbr64
0xcff4: push     rbx
0xcff5: mov      rbx, qword ptr [rip + 0x1e40c]
0xcffc: mov      esi, 1
0xd001: mov      qword ptr [rbx + 0x18], 0
0xd009: mov      rdi, qword ptr [rbx + 0x58]
0xd00d: mov      qword ptr [rbx + 0x20], 0
0xd015: mov      qword ptr [rbx + 0x38], 0
0xd01d: mov      qword ptr [rbx + 0x40], 0
0xd025: mov      qword ptr [rbx + 0x48], 0
0xd02d: mov      qword ptr [rbx + 0x50], 1
0xd035: call     0x1d690
0xd03a: mov      qword ptr [rbx + 0x58], rax
0xd03e: mov      rax, qword ptr [rip + 0x1e3c3]
0xd045: mov      rax, qword ptr [rax + 0x58]
0xd049: mov      byte ptr [rax], 0
0xd04c: mov      rax, qword ptr [rip + 0x1e3b5]

```

```

0xd053: mov     dword ptr [rax + 0x10], 0xc
0xd05a: pop     rbx
0xd05b: ret
0xd05c: nop     dword ptr [rax]
0xd060: endbr64
0xd064: push    rbx
0xd065: mov     edi, 0x70
0xd06a: sub     rsp, 0x10
0xd06e: mov     rax, qword ptr fs:[0x28]
0xd077: mov     qword ptr [rsp + 8], rax
0xd07c: xor     eax, eax
0xd07e: call    0x1d670
0xd083: mov     qword ptr [rax + 8], 1
0xd08b: mov     qword ptr [rax + 0x58], 0
0xd093: mov     qword ptr [rax + 0x60], 0
0xd09b: mov     qword ptr [rax + 0x28], 0
0xd0a3: mov     qword ptr [rax + 0x30], 0
0xd0ab: mov     qword ptr [rip + 0x1e356], rax
0xd0b2: xor     eax, eax
0xd0b4: call    0xcff0
0xd0b9: mov     rbx, qword ptr [rip + 0x1e348]
0xd0c0: xor     eax, eax
0xd0c2: call    0xc590
0xd0c7: lea     rdi, [rip + 0x135e4]
0xd0ce: mov     qword ptr [rbx], rax
0xd0d1: call    0x17c50
0xd0d6: mov     ecx, 1
0xd0db: mov     rsi, rsp
0xd0de: mov     edx, 1
0xd0e3: lea     rdi, [rip + 0x13e42]
0xd0ea: call    0x17320
0xd0ef: mov     rdi, qword ptr [rsp]
0xd0f3: lea     rax, [rip + 0x13e24]
0xd0fa: mov     rbx, qword ptr [rip + 0x1e307]
0xd101: test    rdi, rdi
0xd104: cmov    rdi, rax
0xd108: xor     eax, eax
0xd10a: call    0x19940
0xd10f: lea     rdi, [rip + 0x1cb32]
0xd116: mov     qword ptr [rbx + 0x68], rax
0xd11a: mov     eax, dword ptr [rip + 0x1cb40]
0xd120: test    eax, eax
0xd122: lea     rax, [rip + 0x1cb07]
0xd129: cmov    rdi, rax
0xd12d: call    0x17c50
0xd132: lea     rdi, [rip + 0x1d0c7]
0xd139: mov     edx, 6
0xd13e: lea     rsi, [rip + 0x1d0db]

```



```

0xd145: call    0x17f70
0xd14a: xor      eax, eax
0xd14c: call    0x1a310
0xd151: mov      rax, qword ptr [rip + 0x1e2b0]
0xd158: mov      rdi, qword ptr [rax + 0x68]
0xd15c: xor      eax, eax
0xd15e: call    0x1a900
0xd163: nop      dword ptr [rax + rax]
0xd168: xor      eax, eax
0xd16a: call    0x1ab40
0xd16f: test     rax, rax
0xd172: jne      0xd168
0xd174: mov      edi, 0x200
0xd179: call    0x1a620
0xd17e: mov      rax, qword ptr [rsp + 8]
0xd183: sub      rax, qword ptr fs:[0x28]
0xd18c: jne      0xd194
0xd18e: add      rsp, 0x10
0xd192: pop      rbx
0xd193: ret
0xd194: call    0x49d0
0xd199: nop      dword ptr [rax]
0xd1a0: endbr64
0xd1a4: sub      rsp, 8
0xd1a8: mov      rax, qword ptr [rip + 0x1e259]
0xd1af: or       edi, 2
0xd1b2: mov      rdx, qword ptr [rax + 0x40]
0xd1b6: mov      dword ptr [rax + 0x10], 6
0xd1bd: mov      qword ptr [rax + 0x20], rdx
0xd1c1: mov      rdx, qword ptr [rax + 0x38]
0xd1c5: mov      qword ptr [rax + 0x18], rdx
0xd1c9: xor      eax, eax
0xd1cb: call    0xc6e0
0xd1d0: mov      rax, qword ptr [rip + 0x1e231]
0xd1d7: mov      rdx, qword ptr [rax + 0x18]
0xd1db: mov      dword ptr [rax + 0x10], 0xd
0xd1e2: mov      qword ptr [rax + 0x20], rdx
0xd1e6: add      rsp, 8
0xd1ea: ret
0xd1eb: nop      dword ptr [rax + rax]
0xd1f0: endbr64
0xd1f4: mov      rax, qword ptr [rip + 0x1e20d]
0xd1fb: push     rbx
0xd1fc: mov      edi, 3
0xd201: mov      rdx, qword ptr [rax + 0x18]
0xd205: mov      rsi, qword ptr [rax + 0x20]
0xd209: mov      dword ptr [rax + 0x10], 5
0xd210: mov      rcx, qword ptr [rax + 0x40]

```

```

0xd214: mov     qword ptr [rax + 0x20], rdx
0xd218: mov     rbx, rsi
0xd21b: mov     qword ptr [rax + 0x18], rcx
0xd21f: sub     rbx, rdx
0xd222: cmp     rsi, rdx
0xd225: cmovbe  rbx, rcx
0xd229: xor     eax, eax
0xd22b: call    0xc6e0
0xd230: mov     rdx, qword ptr [rip + 0x1e1d1]
0xd237: mov     rax, qword ptr [rdx + 0x38]
0xd23b: mov     dword ptr [rdx + 0x10], 0xe
0xd242: cmp     rax, rbx
0xd245: cmova   rax, rbx
0xd249: mov     qword ptr [rdx + 0x20], rax
0xd24d: pop     rbx
0xd24e: ret
0xd24f: nop
0xd250: endbr64
0xd254: mov     rax, qword ptr [rip + 0x1e1ad]
0xd25b: push    rbx
0xd25c: mov     edi, 3
0xd261: mov     rdx, qword ptr [rax + 0x18]
0xd265: mov     rbx, qword ptr [rax + 0x20]
0xd269: mov     dword ptr [rax + 0x10], 6
0xd270: mov     qword ptr [rax + 0x20], rdx
0xd274: mov     rdx, qword ptr [rax + 0x38]
0xd278: mov     qword ptr [rax + 0x18], rdx
0xd27c: xor     eax, eax
0xd27e: call    0xc6e0
0xd283: mov     rdx, qword ptr [rip + 0x1e17e]
0xd28a: mov     rax, qword ptr [rdx + 0x38]
0xd28e: mov     dword ptr [rdx + 0x10], 0xf
0xd295: cmp     rax, rbx
0xd298: cmova   rax, rbx
0xd29c: mov     qword ptr [rdx + 0x20], rax
0xd2a0: pop     rbx
0xd2a1: ret
0xd2a2: nop     word ptr cs:[rax + rax]
0xd2ad: nop     dword ptr [rax]
0xd2b0: endbr64
0xd2b4: mov     rax, qword ptr [rip + 0x1e14d]
0xd2bb: mov     rdx, qword ptr [rax + 0x58]
0xd2bf: mov     rax, qword ptr [rax + 0x18]
0xd2c3: cmp     byte ptr [rdx + rax], 0x20
0xd2c7: je      0xd2d0
0xd2c9: ret
0xd2ca: nop     word ptr [rax + rax]
0xd2d0: sub     rsp, 8

```

```

0xd2d4: nop      dword ptr [rax]
0xd2d8: xor       eax, eax
0xd2da: call      0xcde0
0xd2df: mov       rax, qword ptr [rip + 0x1e122]
0xd2e6: mov       rdx, qword ptr [rax + 0x58]
0xd2ea: mov       rcx, qword ptr [rax + 0x18]
0xd2ee: cmp       byte ptr [rdx + rcx], 0x20
0xd2f2: je        0xd2d8
0xd2f4: cmp       qword ptr [rax + 0x48], 0
0xd2f9: jne       0xd31d
0xd2fb: jmp       0xd324
0xd324:
0xd2fd: nop      dword ptr [rax]
0xd300: xor       eax, eax
0xd302: call      0xce70
0xd307: mov       rax, qword ptr [rip + 0x1e0fa]
0xd30e: cmp       qword ptr [rax + 0x48], 0
0xd313: je        0xd324
0xd315: mov       rdx, qword ptr [rax + 0x58]
0xd319: mov       rcx, qword ptr [rax + 0x18]
0xd31d: cmp       byte ptr [rdx + rcx - 1], 0x20
0xd322: je        0xd300
0xd324: mov       dword ptr [rax + 0x10], 0x11
0xd32b: add       rsp, 8
0xd32f: ret
0xd330: endbr64
0xd334: mov       rax, qword ptr [rip + 0x1e0cd]
0xd33b: mov       rdx, qword ptr [rax + 0x58]
0xd33f: mov       rax, qword ptr [rax + 0x18]
0xd343: cmp       byte ptr [rdx + rax], 0x20
0xd347: je        0xd350
0xd349: ret
0xd34a: nop      word ptr [rax + rax]
0xd350: sub       rsp, 8
0xd354: xor       eax, eax
0xd356: call      0xd2b0
0xd35b: mov       edi, 0x20
0xd360: call      0xcd30
0xd365: mov       rax, qword ptr [rip + 0x1e09c]
0xd36c: mov       dword ptr [rax + 0x10], 0x10
0xd373: add       rsp, 8
0xd377: ret
0xd378: nop      dword ptr [rax + rax]
0xd380: endbr64
0xd384: push     rbp
0xd385: push     rbx
0xd386: sub       rsp, 8
0xd38a: mov       rbp, qword ptr [rip + 0x1e077]

```

```

0xd391: mov     rbx, qword ptr [rbp + 0x18]
0xd395: cmp     rbx, qword ptr [rbp + 0x38]
0xd399: jb      0xd3a8
0xd39b: add     rsp, 8
0xd39f: pop     rbx
0xd3a0: pop     rbp
0xd3a1: ret
0xd3a2: nop     word ptr [rax + rax]
0xd3a8: xor     eax, eax
0xd3aa: call    0xcc30
0xd3af: cmp     rbx, qword ptr [rbp + 0x18]
0xd3b3: jae     0xd3e4
0xd3b5: call    0x4ef0
0xd3ba: nop     word ptr [rax + rax]
0xd3c0: mov     rdx, qword ptr [rbp + 0x58]
0xd3c4: mov     rcx, qword ptr [rax]
0xd3c7: add     rdx, rbx
0xd3ca: add     rbx, 1
0xd3ce: movsx   rsi, byte ptr [rdx]
0xd3d2: mov     ecx, dword ptr [rcx + rsi*4]
0xd3d5: mov     byte ptr [rdx], cl
0xd3d7: mov     rbp, qword ptr [rip + 0x1e02a]
0xd3de: cmp     qword ptr [rbp + 0x18], rbx
0xd3e2: ja      0xd3c0
0xd3e4: mov     dword ptr [rbp + 0x10], 0x12
0xd3eb: add     rsp, 8
0xd3ef: pop     rbx
0xd3f0: pop     rbp
0xd3f1: ret
0xd3f2: nop     word ptr cs:[rax + rax]
0xd3fd: nop     dword ptr [rax]
0xd400: endbr64
0xd404: push    rbp
0xd405: push    rbx
0xd406: sub     rsp, 8
0xd40a: mov     rbp, qword ptr [rip + 0x1dff7]
0xd411: mov     rbx, qword ptr [rbp + 0x18]
0xd415: cmp     rbx, qword ptr [rbp + 0x38]
0xd419: jb      0xd428
0xd41b: add     rsp, 8
0xd41f: pop     rbx
0xd420: pop     rbp
0xd421: ret
0xd422: nop     word ptr [rax + rax]
0xd428: xor     eax, eax
0xd42a: call    0xcc30
0xd42f: cmp     rbx, qword ptr [rbp + 0x18]
0xd433: jae     0xd464

```

```

0xd435: call    0x47c0
0xd43a: nop      word ptr [rax + rax]
0xd440: mov      rdx, qword ptr [rbp + 0x58]
0xd444: mov      rcx, qword ptr [rax]
0xd447: add      rdx, rbx
0xd44a: add      rbx, 1
0xd44e: movsx    rsi, byte ptr [rdx]
0xd452: mov      ecx, dword ptr [rcx + rsi*4]
0xd455: mov      byte ptr [rdx], cl
0xd457: mov      rbp, qword ptr [rip + 0x1dfaa]
0xd45e: cmp      qword ptr [rbp + 0x18], rbx
0xd462: ja       0xd440
0xd464: mov      dword ptr [rbp + 0x10], 0x13
0xd46b: add      rsp, 8
0xd46f: pop      rbx
0xd470: pop      rbp
0xd471: ret
0xd472: nop      word ptr cs:[rax + rax]
0xd47d: nop      dword ptr [rax]
0xd480: endbr64
0xd484: push     r14
0xd486: push     r13
0xd488: push     r12
0xd48a: push     rbp
0xd48b: mov      rbp, qword ptr [rip + 0x1df76]
0xd492: push     rbx
0xd493: mov      rbx, qword ptr [rbp + 0x18]
0xd497: cmp      rbx, qword ptr [rbp + 0x38]
0xd49b: jb       0xd4b0
0xd49d: pop      rbx
0xd49e: pop      rbp
0xd49f: pop      r12
0xd4a1: pop      r13
0xd4a3: pop      r14
0xd4a5: ret
0xd4a6: nop      word ptr cs:[rax + rax]
0xd4b0: xor      eax, eax
0xd4b2: call     0xcc30
0xd4b7: cmp      rbx, qword ptr [rbp + 0x18]
0xd4bb: jae      0xd528
0xd4bd: call     0x4f00
0xd4c2: mov      r14d, 1
0xd4c8: mov      r13, rax
0xd4cb: jmp      0xd4f3
0xd4f3:
0xd4cd: nop      dword ptr [rax]
0xd4d0: call     0x47c0
0xd4d5: xor      r14d, r14d

```

```

0xd4d8: mov     rax, qword ptr [rax]
0xd4db: mov     eax, dword ptr [rax + r12*4]
0xd4df: mov     byte ptr [rbp], al
0xd4e2: mov     rbp, qword ptr [rip + 0x1df1f]
0xd4e9: add     rbx, 1
0xd4ed: cmp     qword ptr [rbp + 0x18], rbx
0xd4f1: jbe     0xd528
0xd4f3: mov     rax, qword ptr [rbp + 0x58]
0xd4f7: add     rax, rbx
0xd4fa: movsx   r12, byte ptr [rax]
0xd4fe: mov     rbp, rax
0xd501: mov     rax, qword ptr [r13]
0xd505: test    byte ptr [rax + r12*2], 8
0xd50a: je      0xd4e2
0xd50c: test    r14d, r14d
0xd50f: jne     0xd4d0
0xd511: call    0x4ef0
0xd516: mov     rax, qword ptr [rax]
0xd519: mov     eax, dword ptr [rax + r12*4]
0xd51d: mov     byte ptr [rbp], al
0xd520: jmp     0xd4e2
0xd4e2:
0xd522: nop     word ptr [rax + rax]
0xd528: mov     dword ptr [rbp + 0x10], 0x14
0xd52f: pop     rbx
0xd530: pop     rbp
0xd531: pop     r12
0xd533: pop     r13
0xd535: pop     r14
0xd537: ret
0xd538: nop     dword ptr [rax + rax]
0xd540: endbr64
0xd544: push    r12
0xd546: mov     r12, rdi
0xd549: push    rbp
0xd54a: push    rbx
0xd54b: call    0x49b0
0xd550: mov     rbp, qword ptr [rip + 0x1deb1]
0xd557: mov     rbx, rax
0xd55a: mov     rsi, qword ptr [rbp + 0x40]
0xd55e: sub     rax, rsi
0xd561: add     qword ptr [rbp + 0x18], rax
0xd565: add     qword ptr [rbp + 0x20], rax
0xd569: mov     rax, qword ptr [rbp + 0x48]
0xd56d: lea     rdx, [rax + 1]
0xd571: lea     r8, [rbx + rdx]
0xd575: cmp     r8, qword ptr [rbp + 0x50]
0xd579: ja      0xd5d8

```

```

0xd57b: mov     rdi, qword ptr [rbp + 0x58]
0xd57f: add     rsi, rdi
0xd582: add     rdi, rbx
0xd585: call    0x4d40
0xd58a: mov     rax, qword ptr [rip + 0x1de77]
0xd591: mov     rdx, rbx
0xd594: mov     rsi, r12
0xd597: mov     rdi, qword ptr [rax + 0x58]
0xd59b: call    0x4bb0
0xd5a0: mov     rax, qword ptr [rip + 0x1de61]
0xd5a7: mov     rsi, rbx
0xd5aa: mov     rdi, qword ptr [rax + 0x58]
0xd5ae: call    0x184d0
0xd5b3: mov     rax, qword ptr [rip + 0x1de4e]
0xd5ba: mov     qword ptr [rax + 0x40], rbx
0xd5be: add     rbx, qword ptr [rax + 0x48]
0xd5c2: mov     qword ptr [rax + 0x38], rbx
0xd5c6: mov     dword ptr [rax + 0x10], 0x15
0xd5cd: pop     rbx
0xd5ce: pop     rbp
0xd5cf: pop     r12
0xd5d1: ret
0xd5d2: nop     word ptr [rax + rax]
0xd5d8: mov     qword ptr [rbp + 0x50], r8
0xd5dc: mov     rdi, qword ptr [rbp + 0x58]
0xd5e0: mov     rsi, r8
0xd5e3: call    0x1d690
0xd5e8: mov     qword ptr [rbp + 0x58], rax
0xd5ec: mov     rbp, qword ptr [rip + 0x1de15]
0xd5f3: mov     rax, qword ptr [rbp + 0x48]
0xd5f7: mov     rsi, qword ptr [rbp + 0x40]
0xd5fb: lea     rdx, [rax + 1]
0xd5ff: jmp     0xd57b
0xd57b:
0xd604: nop     word ptr cs:[rax + rax]
0xd60f: nop
0xd610: endbr64
0xd614: test    rdi, rdi
0xd617: je      0xd6f0
0xd61d: push    r12
0xd61f: push    rbp
0xd620: mov     rbp, rdi
0xd623: push    rbx
0xd624: call    0x49b0
0xd629: mov     r12, qword ptr [rip + 0x1ddd8]
0xd630: mov     rbx, rax
0xd633: mov     rax, qword ptr [r12 + 0x38]
0xd638: lea     rdx, [rax + 1]

```

```

0xd63c: mov     rax, qword ptr [r12 + 0x50]
0xd641: lea     rcx, [rbx + rdx]
0xd645: cmp     rcx, rax
0xd648: ja      0xd6c0
0xd64a: mov     rax, qword ptr [r12 + 0x18]
0xd64f: mov     rdi, qword ptr [r12 + 0x58]
0xd654: sub     rdx, rax
0xd657: lea     rsi, [rdi + rax]
0xd65b: add     rax, rbx
0xd65e: add     rdi, rax
0xd661: call    0x4d40
0xd666: mov     rax, qword ptr [rip + 0x1dd9b]
0xd66d: mov     rdx, rbx
0xd670: mov     rsi, rbp
0xd673: mov     rdi, qword ptr [rax + 0x18]
0xd677: add     rdi, qword ptr [rax + 0x58]
0xd67b: call    0x4bb0
0xd680: mov     rax, qword ptr [rip + 0x1dd81]
0xd687: mov     rsi, rbx
0xd68a: mov     rdi, qword ptr [rax + 0x18]
0xd68e: add     rdi, qword ptr [rax + 0x58]
0xd692: call    0x184d0
0xd697: mov     rax, qword ptr [rip + 0x1dd6a]
0xd69e: add     qword ptr [rax + 0x18], rbx
0xd6a2: add     qword ptr [rax + 0x38], rbx
0xd6a6: add     qword ptr [rax + 0x48], rbx
0xd6aa: mov     dword ptr [rax + 0x10], 0x16
0xd6b1: pop     rbx
0xd6b2: pop     rbp
0xd6b3: pop     r12
0xd6b5: ret
0xd6b6: nop     word ptr cs:[rax + rax]
0xd6c0: lea     rsi, [rax + rbx + 0x21]
0xd6c5: mov     rdi, qword ptr [r12 + 0x58]
0xd6ca: mov     qword ptr [r12 + 0x50], rsi
0xd6cf: call    0x1d690
0xd6d4: mov     qword ptr [r12 + 0x58], rax
0xd6d9: mov     r12, qword ptr [rip + 0x1dd28]
0xd6e0: mov     rax, qword ptr [r12 + 0x38]
0xd6e5: lea     rdx, [rax + 1]
0xd6e9: jmp     0xd64a
0xd64a:
0xd6ee: nop
0xd6f0: ret
0xd6f1: nop     word ptr cs:[rax + rax]
0xd6fc: nop     dword ptr [rax]
0xd700: endbr64
0xd704: mov     rax, qword ptr [rip + 0x1dcfd]

```



```

0xd70b: mov     rdi, qword ptr [rax + 0x60]
0xd70f: test    rdi, rdi
0xd712: je      0xd730
0xd714: sub     rsp, 8
0xd718: call    0xd610
0xd71d: mov     rax, qword ptr [rip + 0x1dce4]
0xd724: mov     dword ptr [rax + 0x10], 0x1a
0xd72b: add     rsp, 8
0xd72f: ret
0xd730: ret
0xd731: nop     word ptr cs:[rax + rax]
0xd73c: nop     dword ptr [rax]
0xd740: endbr64
0xd744: mov     rax, qword ptr [rip + 0x1dcbd]
0xd74b: push    r12
0xd74d: push    rbp
0xd74e: push    rbx
0xd74f: mov     rbx, qword ptr [rax + 0x40]
0xd753: mov     r12, qword ptr [rax + 0x28]
0xd757: mov     eax, dword ptr [rax + 0xc]
0xd75a: lea     rbp, [rbx + 3]
0xd75e: test    eax, eax
0xd760: je      0xd7f0
0xd766: mov     edx, dword ptr [rip + 0x1caa4]
0xd76c: mov     esi, dword ptr [rip + 0x1ca9a]
0xd772: mov     edi, dword ptr [rip + 0x1ca9c]
0xd778: call    0xa570
0xd77d: mov     rcx, qword ptr [rip + 0x1dc84]
0xd784: cmp     r12, rbp
0xd787: jnb     0xd810
0xd78d: mov     rsi, qword ptr [rcx + 0x18]
0xd791: mov     rdx, qword ptr [rcx + 0x40]
0xd795: xor     ebx, ebx
0xd797: mov     rdi, qword ptr [rcx + 0x28]
0xd79b: mov     rax, rdi
0xd79e: sub     rax, rdx
0xd7a1: mov     edx, 1
0xd7a6: mov     r8, rax
0xd7a9: shr     r8, 2
0xd7ad: cmp     rax, 3
0xd7b1: cmova   edx, r8d
0xd7b5: xor     eax, eax
0xd7b7: cmp     rdi, rsi
0xd7ba: ja      0xd7d3
0xd7bc: mov     rax, rsi
0xd7bf: sub     rax, rdi
0xd7c2: movsxd  rdi, edx
0xd7c5: lea     r8d, [rdx + rax]

```

```

0xd7c9: xor     edx, edx
0xd7cb: div     rdi
0xd7ce: mov     eax, r8d
0xd7d1: sub     eax, edx
0xd7d3: mov     edx, esi
0xd7d5: mov     rdi, qword ptr [rcx]
0xd7d8: xor     esi, esi
0xd7da: sub     edx, eax
0xd7dc: call    0xc680
0xd7e1: cmp     r12, rbp
0xd7e4: jb      0xd840
0xd7e6: pop     rbx
0xd7e7: pop     rbp
0xd7e8: pop     r12
0xd7ea: ret
0xd7eb: nop     dword ptr [rax + rax]
0xd7f0: mov     edx, dword ptr [rip + 0x1ca0e]
0xd7f6: mov     esi, dword ptr [rip + 0x1ca04]
0xd7fc: mov     edi, dword ptr [rip + 0x1ca06]
0xd802: jmp     0xd778
0xd778:
0xd807: nop     word ptr [rax + rax]
0xd810: mov     rbx, qword ptr [rcx + 0x40]
0xd814: mov     rsi, qword ptr [rcx + 0x18]
0xd818: mov     qword ptr [rcx + 0x40], 0
0xd820: xor     edx, edx
0xd822: add     qword ptr [rcx + 0x58], rbx
0xd826: sub     rsi, rbx
0xd829: sub     qword ptr [rcx + 0x38], rbx
0xd82d: sub     qword ptr [rcx + 0x20], rbx
0xd831: movsxd  rbx, ebx
0xd834: mov     qword ptr [rcx + 0x18], rsi
0xd838: jmp     0xd797
0xd797:
0xd83d: nop     dword ptr [rax]
0xd840: mov     rax, qword ptr [rip + 0x1dbc1]
0xd847: movsxd  rbx, ebx
0xd84a: mov     qword ptr [rax + 0x40], rbx
0xd84e: sub     qword ptr [rax + 0x58], rbx
0xd852: add     qword ptr [rax + 0x38], rbx
0xd856: add     qword ptr [rax + 0x18], rbx
0xd85a: add     qword ptr [rax + 0x20], rbx
0xd85e: pop     rbx
0xd85f: pop     rbp
0xd860: pop     r12
0xd862: ret
0xd863: nop     word ptr cs:[rax + rax]
0xd86e: nop

```

```

0xd870: endbr64
0xd874: push    r15
0xd876: push    r14
0xd878: push    r13
0xd87a: push    r12
0xd87c: push    rbp
0xd87d: push    rbx
0xd87e: sub     rsp, 0x28
0xd882: mov     rax, qword ptr fs:[0x28]
0xd88b: mov     qword ptr [rsp + 0x18], rax
0xd890: mov     rax, qword ptr [rip + 0x1db71]
0xd897: lea     r14, [rsp + 0x17]
0xd89c: mov     rdi, r14
0xd89f: mov     rbx, qword ptr [rax + 0x40]
0xd8a3: mov     r13, qword ptr [rax + 0x28]
0xd8a7: call    0xa6a0
0xd8ac: mov     rax, qword ptr [rip + 0x1db55]
0xd8b3: add     rbx, 3
0xd8b7: mov     ecx, dword ptr [rax + 0xc]
0xd8ba: test    ecx, ecx
0xd8bc: je      0xda20
0xd8c2: mov     edx, dword ptr [rip + 0x1c948]
0xd8c8: mov     esi, dword ptr [rip + 0x1c93e]
0xd8ce: mov     edi, dword ptr [rip + 0x1c940]
0xd8d4: call    0xa570
0xd8d9: mov     rdx, qword ptr [rip + 0x1db28]
0xd8e0: cmp     r13, rbx
0xd8e3: jb      0xda90
0xd8e9: mov     qword ptr [rsp + 8], 0
0xd8f2: mov     rax, qword ptr [rdx + 0x18]
0xd8f6: mov     rcx, qword ptr [rdx + 0x40]
0xd8fa: mov     rdi, qword ptr [rdx + 0x28]
0xd8fe: mov     r15, rdi
0xd901: sub     r15, rcx
0xd904: cmp     r15, 3
0xd908: jbe     0xda80
0xd90e: shr     r15, 2
0xd912: movsxd  rcx, r15d
0xd915: cmp     rdi, rax
0xd918: ja      0xda70
0xd91e: mov     edx, eax
0xd920: sub     rax, rdi
0xd923: sub     edx, edi
0xd925: add     r15d, edx
0xd928: xor     edx, edx
0xd92a: div     rcx
0xd92d: sub     r15d, edx
0xd930: mov     r12d, r15d

```

```

0xd933: call    0x1d670
0xd938: mov     esi, 0x20
0xd93d: mov     rbp, rax
0xd940: mov     rax, qword ptr [rip + 0x1dac1]
0xd947: mov     rdi, rbp
0xd94a: mov     rdx, qword ptr [rax + 0x28]
0xd94e: call    0x4a50
0xd953: mov     rax, qword ptr [rip + 0x1daae]
0xd95a: mov     edx, dword ptr [rax + 8]
0xd95d: test    edx, edx
0xd95f: je      0xda40
0xd965: mov     rdx, qword ptr [rax + 0x38]
0xd969: sub     rdx, qword ptr [rax + 0x40]
0xd96d: mov     rdi, rbp
0xd970: sub     rdx, r12
0xd973: mov     rsi, qword ptr [rax + 0x28]
0xd977: sub     rsi, qword ptr [rax + 0x40]
0xd97b: cmp     rdx, rsi
0xd97e: cmova   rdx, rsi
0xd982: mov     rsi, qword ptr [rax + 0x40]
0xd986: add     rsi, r12
0xd989: add     rsi, qword ptr [rax + 0x58]
0xd98d: call    0x4bb0
0xd992: mov     rax, qword ptr [rip + 0x1da6f]
0xd999: xor     edx, edx
0xd99b: xor     esi, esi
0xd99d: mov     rdi, qword ptr [rax]
0xd9a0: call    0xc680
0xd9a5: cmp     r13, rbx
0xd9a8: jae     0xdaf0
0xd9ae: mov     rax, qword ptr [rip + 0x1da53]
0xd9b5: mov     rsi, rbp
0xd9b8: mov     rdi, qword ptr [rax]
0xd9bb: mov     rdx, qword ptr [rax + 0x28]
0xd9bf: sub     edx, dword ptr [rax + 0x40]
0xd9c2: call    0xc600
0xd9c7: mov     rax, qword ptr [rip + 0x1da3a]
0xd9ce: xor     esi, esi
0xd9d0: mov     edx, dword ptr [rax + 0x18]
0xd9d3: mov     rdi, qword ptr [rax]
0xd9d6: sub     edx, r15d
0xd9d9: call    0xc680
0xd9de: cmp     r13, rbx
0xd9e1: jnb     0xdac8
0xd9e7: mov     rdi, rbp
0xd9ea: call    0x18d10
0xd9ef: mov     rdi, r14
0xd9f2: call    0xa6b0

```

```

0xd9f7: mov     rax, qword ptr [rsp + 0x18]
0xd9fc: sub     rax, qword ptr fs:[0x28]
0xda05: jne     0xdb0b
0xda0b: add     rsp, 0x28
0xda0f: pop     rbx
0xda10: pop     rbp
0xda11: pop     r12
0xda13: pop     r13
0xda15: pop     r14
0xda17: pop     r15
0xda19: ret
0xda1a: nop     word ptr [rax + rax]
0xda20: mov     edx, dword ptr [rip + 0x1c7de]
0xda26: mov     esi, dword ptr [rip + 0x1c7d4]
0xda2c: mov     edi, dword ptr [rip + 0x1c7d6]
0xda32: jmp     0xd8d4
0xd8d4
0xda37: nop     word ptr [rax + rax]
0xda40: mov     rsi, qword ptr [rax + 0x40]
0xda44: mov     rdx, qword ptr [rax + 0x38]
0xda48: mov     rdi, rbp
0xda4b: mov     rax, qword ptr [rax + 0x28]
0xda4f: sub     rdx, rsi
0xda52: sub     rdx, r12
0xda55: sub     rax, rsi
0xda58: mov     esi, 0x2a
0xda5d: cmp     rdx, rax
0xda60: cmova   rdx, rax
0xda64: call    0x4a50
0xda69: jmp     0xd992
0xd992
0xda6e: nop
0xda70: xor     r12d, r12d
0xda73: xor     r15d, r15d
0xda76: jmp     0xd933
0xd933
0xda7b: nop     dword ptr [rax + rax]
0xda80: mov     ecx, 1
0xda85: mov     r15d, 1
0xda8b: jmp     0xd915
0xd915
0xda90: mov     r12, qword ptr [rdx + 0x40]
0xda94: mov     rax, qword ptr [rdx + 0x18]
0xda98: mov     qword ptr [rdx + 0x40], 0
0xdaa0: add     qword ptr [rdx + 0x58], r12
0xdaa4: movsxd rcx, r12d
0xdaa7: sub     rax, r12
0xdaaa: sub     qword ptr [rdx + 0x38], r12

```

```

0xdaae: mov     qword ptr [rsp + 8], rcx
0xdab3: xor     ecx, ecx
0xdab5: sub     qword ptr [rdx + 0x20], r12
0xdab9: mov     qword ptr [rdx + 0x18], rax
0xdabd: jmp     0xd8fa
0xd8fa
0xdac2: nop     word ptr [rax + rax]
0xdac8: mov     rax, qword ptr [rip + 0x1d939]
0xdacf: movsxd  r12, dword ptr [rsp + 8]
0xdad4: sub     qword ptr [rax + 0x58], r12
0xdad8: add     qword ptr [rax + 0x38], r12
0xdadc: add     qword ptr [rax + 0x18], r12
0xdae0: add     qword ptr [rax + 0x20], r12
0xdae4: mov     qword ptr [rax + 0x40], r12
0xdae8: jmp     0xd9e7
0xd9e7
0xdaed: nop     dword ptr [rax]
0xdaf0: mov     rax, qword ptr [rip + 0x1d911]
0xdaf7: mov     rsi, qword ptr [rax + 0x58]
0xdafb: mov     edx, dword ptr [rax + 0x40]
0xdafe: mov     rdi, qword ptr [rax]
0xdb01: call    0xc600
0xdb06: jmp     0xd9ae
0xd9ae
0xdb0b: call    0x49d0
0xdb10: endbr64
0xdb14: mov     rax, qword ptr [rip + 0x1d8ed]
0xdb1b: push    rbx
0xdb1c: mov     rbx, rdi
0xdb1f: mov     rdi, qword ptr [rdi]
0xdb22: mov     rsi, qword ptr [rax + 0x48]
0xdb26: add     rsi, 1
0xdb2a: call    0x1d690
0xdb2f: mov     qword ptr [rbx], rax
0xdb32: mov     rdi, rax
0xdb35: mov     rax, qword ptr [rip + 0x1d8cc]
0xdb3c: mov     rcx, qword ptr [rax + 0x48]
0xdb40: mov     rsi, qword ptr [rax + 0x40]
0xdb44: add     rsi, qword ptr [rax + 0x58]
0xdb48: lea     rdx, [rcx + 1]
0xdb4c: call    0x4bb0
0xdb51: mov     rax, qword ptr [rip + 0x1d8b0]
0xdb58: pop     rbx
0xdb59: mov     eax, dword ptr [rax + 0x48]
0xdb5c: ret
0xdb5d: nop     dword ptr [rax]
0xdb60: endbr64
0xdb64: push    rbp

```

```

0xdb65: xor     eax, eax
0xdb67: mov     rbp, rdi
0xdb6a: call    0xcff0
0xdb6f: test    rbp, rbp
0xdb72: lea     rax, [rip + 0x133bf]
0xdb79: cmov     rbp, rax
0xdb7d: mov     rdi, rbp
0xdb80: call    0xd540
0xdb85: xor     eax, eax
0xdb87: call    0xd870
0xdb8c: xor     eax, eax
0xdb8e: pop     rbp
0xdb8f: jmp     0xd740
0xd740:
0xdb94: nop     word ptr cs:[rax + rax]
0xdb9f: nop
0xdba0: endbr64
0xdba4: mov     rax, qword ptr [rip + 0x1d85d]
0xdbab: mov     dword ptr [rax + 0xc], edi
0xdbae: ret
0xdbaf: nop
0xdbb0: endbr64
0xdbb4: push    r12
0xdbb6: push    rbp
0xdbb7: push    rbx
0xdbb8: cmp     edi, 1
0xdbbb: je      0xdbd0
0xdbbd: cmp     edi, 2
0xdbc0: je      0xdc90
0xdbc6: test    edi, edi
0xdbc8: je      0xdc40
0xdbca: pop     rbx
0xdbc b: pop     rbp
0xdbc c: pop     r12
0xdbce: ret
0xdbcf: nop
0xdbd0: mov     ecx, dword ptr [rip + 0x1d83e]
0xdbd6: test    ecx, ecx
0xdbd8: je      0xdd38
0xdbde: xor     eax, eax
0xdbe0: call    0x1ab10
0xdbe5: mov     rbp, rax
0xdbe8: test    rax, rax
0xdbeb: je      0xdbca
0xdbed: mov     r12, qword ptr [rbp]
0xdbf1: mov     rbx, qword ptr [rip + 0x1d810]
0xdbf8: mov     rdi, r12
0xdbfb: call    0x49b0

```

```

0xdc00: mov     rdi, qword ptr [rbx + 0x40]
0xdc04: mov     qword ptr [rbx + 0x48], rax
0xdc08: lea     rsi, [rax + rdi]
0xdc0c: mov     qword ptr [rbx + 0x38], rsi
0xdc10: mov     qword ptr [rbx + 0x18], rsi
0xdc14: add     rsi, 1
0xdc18: cmp     rsi, qword ptr [rbx + 0x50]
0xdc1c: ja      0xdd70
0xdc22: add     rdi, qword ptr [rbx + 0x58]
0xdc26: mov     rsi, r12
0xdc29: call    0x48b0
0xdc2e: pop     rbx
0xdc2f: xor     eax, eax
0xdc31: pop     rbp
0xdc32: pop     r12
0xdc34: jmp     0xd870
0xd870
0xdc39: nop     dword ptr [rax]
0xdc40: mov     eax, dword ptr [rip + 0x1d7ce]
0xdc46: test    eax, eax
0xdc48: jne     0xdd18
0xdc4e: xor     eax, eax
0xdc50: call    0x1ab10
0xdc55: test    rax, rax
0xdc58: je      0xdd98
0xdc5e: mov     rdx, qword ptr [rip + 0x1d7a3]
0xdc65: mov     rsi, qword ptr [rax]
0xdc68: mov     rbp, qword ptr [rdx + 0x40]
0xdc6c: add     rbp, qword ptr [rdx + 0x58]
0xdc70: mov     rdi, rbp
0xdc73: call    0x4b30
0xdc78: test    eax, eax
0xdc7a: jne     0xdd58
0xdc80: pop     rbx
0xdc81: xor     eax, eax
0xdc83: pop     rbp
0xdc84: pop     r12
0xdc86: jmp     0x1ab40
0x1ab40
0xdc8b: nop     dword ptr [rax + rax]
0xdc90: mov     edx, dword ptr [rip + 0x1d77e]
0xdc96: test    edx, edx
0xdc98: je      0xdcf8
0xdc9a: xor     eax, eax
0xdc9c: call    0x1ab40
0xdca1: mov     rbp, rax
0xdca4: test    rax, rax
0xdca7: jne     0xdbed

```



```

0xdcad: mov     rax, qword ptr [rip + 0x1d754]
0xdcb4: mov     edi, 2
0xdcb9: mov     rdx, qword ptr [rax + 0x40]
0xdcbd: mov     dword ptr [rax + 0x10], 6
0xdcc4: mov     qword ptr [rax + 0x20], rdx
0xdcc8: mov     rdx, qword ptr [rax + 0x38]
0xdccc: mov     qword ptr [rax + 0x18], rdx
0xdcd0: xor     eax, eax
0xdcd2: call    0xc6e0
0xdcd7: mov     rax, qword ptr [rip + 0x1d72a]
0xdcde: mov     rdx, qword ptr [rax + 0x18]
0xdce2: mov     dword ptr [rax + 0x10], 0xd
0xdce9: mov     qword ptr [rax + 0x20], rdx
0xdc2e: jmp     0xdc2e
0xdcf2: nop     word ptr [rax + rax]
0xdcf8: mov     dword ptr [rip + 0x1d712], 1
0xdd02: xor     eax, eax
0xdd04: call    0x1a570
0xdd09: mov     dword ptr [rip + 0x1d701], eax
0xdd0f: jmp     0xdc9a
0xdc9a:
0xdd11: nop     dword ptr [rax]
0xdd18: mov     edi, dword ptr [rip + 0x1d6f2]
0xdd1e: xor     eax, eax
0xdd20: call    0x1ac30
0xdd25: mov     dword ptr [rip + 0x1d6e5], 0
0xdd2f: jmp     0xdc4e
0xdc4e:
0xdd34: nop     dword ptr [rax]
0xdd38: mov     dword ptr [rip + 0x1d6d2], 1
0xdd42: xor     eax, eax
0xdd44: call    0x1a570
0xdd49: mov     dword ptr [rip + 0x1d6c1], eax
0xdd4f: jmp     0xdbde
0xdbde:
0xdd54: nop     dword ptr [rax]
0xdd58: xor     eax, eax
0xdd5a: mov     rdi, rbp
0xdd5d: call    0x1a380
0xdd62: xor     eax, eax
0xdd64: call    0x1ab40
0xdd69: jmp     0xdc80
0xdc80:
0xdd6e: nop
0xdd70: mov     qword ptr [rbx + 0x50], rsi
0xdd74: mov     rdi, qword ptr [rbx + 0x58]
0xdd78: call    0x1d690

```

```

0xdd7d: mov     qword ptr [rbx + 0x58], rax
0xdd81: mov     rbx, qword ptr [rip + 0x1d680]
0xdd88: mov     r12, qword ptr [rbp]
0xdd8c: mov     rdi, qword ptr [rbx + 0x40]
0xdd90: jmp     0xdc22
0xdc22:
0xdd95: nop     dword ptr [rax]
0xdd98: mov     rax, qword ptr [rip + 0x1d669]
0xdd9f: mov     rdi, qword ptr [rax + 0x40]
0xdda3: add     rdi, qword ptr [rax + 0x58]
0xdda7: xor     eax, eax
0xdda9: call    0x1a380
0xddae: jmp     0xdc80
0xdc80:
0xddb3: nop     word ptr cs:[rax + rax]
0xddbd: nop     dword ptr [rax]
0xddc0: endbr64
0ddc4: mov     eax, dword ptr [rip + 0x1be96]
0ddca: push    rbp
0ddcb: mov     rbp, rdi
0ddce: lea     rdi, [rip + 0x1be73]
0ddd5: test    eax, eax
0ddd7: lea     rax, [rip + 0x1be52]
0dddde: cmove   rdi, rax
0dde2: call    0x17c50
0dde7: mov     edx, 9
0ddeec: lea     rsi, [rip + 0x1c4ad]
0ddf3: lea     rdi, [rip + 0x1c466]
0dddfa: call    0x17f70
0ddfff: mov     rdi, rbp
0xde02: call    0x1d7a0
0xde07: mov     rdi, rax
0xde0a: mov     rbp, rax
0xde0d: mov     qword ptr [rip + 0x1d794], rax
0xde14: call    0x49b0
0xde19: mov     rdi, rbp
0xde1c: mov     rsi, rax
0xde1f: call    0x184d0
0xde24: xor     eax, eax
0xde26: call    0xc590
0xde2b: lea     rdi, [rip + 0x1d5ee]
0xde32: pop     rbp
0xde33: mov     qword ptr [rip + 0x1d78e], rax
0xde3a: jmp     0x4a30
0x4a30:
0xde3f: nop
0xde40: endbr64
0xde44: mov     rdi, qword ptr [rip + 0x1d77d]

```

```

0xde4b: jmp      0xc5c0
0xc5c0
0xde50: endbr64
0xde54: push     rbp
0xde55: mov      rbp, rdi
0xde58: push     rbx
0xde59: mov      rbx, rsi
0xde5c: sub      rsp, 8
0xde60: mov      rdi, qword ptr [rip + 0x1d749]
0xde67: test     rdi, rdi
0xde6a: je       0xde71
0xde6c: call     0x18d10
0xde71: mov      rdi, rbp
0xde74: call     0x1d670
0xde79: mov      rdi, qword ptr [rip + 0x1d748]
0xde80: mov      edx, ebx
0xde82: xor      esi, esi
0xde84: mov      qword ptr [rip + 0x1d725], rax
0xde8b: add      rsp, 8
0xde8f: mov      r8d, ebp
0xde92: mov      ecx, 1
0xde97: pop      rbx
0xde98: pop      rbp
0xde99: jmp      0xc5e0
0xc5e0
0xde9e: nop
0xdea0: endbr64
0xdea4: push     rbp
0xdea5: mov      rbp, rsp
0xdea8: push     r15
0xdeaa: push     r14
0xdeac: push     r13
0xdeae: push     r12
0xdeb0: push     rbx
0xdeb1: sub      rsp, 0x68
0xdeb5: mov      rax, qword ptr fs:[0x28]
0xdeb6: mov      qword ptr [rbp - 0x38], rax
0xdec2: xor      eax, eax
0xdec4: lea      rax, [rbp - 0x61]
0xdec8: mov      rdi, rax
0xdecb: mov      qword ptr [rbp - 0x80], rax
0xdecf: call     0xa6a0
0xded4: mov      rax, qword ptr [rip + 0x1d6ed]
0xdedb: mov      esi, 0x20
0xdee0: mov      rdi, qword ptr [rip + 0x1d6c9]
0xdee7: movsxd   rdx, dword ptr [rax + 0xc]
0xdeeb: call     0x4a50
0xdef0: mov      rdi, qword ptr [rip + 0x1d6c9]

```

```

0xdef7: call    0x49b0
0xdefc: lea      r14, [rax + 1]
0xdf00: mov      rdi, r14
0xdf03: call    0x1d670
0xdf08: mov      rcx, qword ptr [rip + 0x1d6b1]
0xdf0f: mov      r13, rax
0xdf12: movzx    eax, byte ptr [rcx]
0xdf15: test     al, al
0xdf17: je       0xe430
0xdf1d: xor      esi, esi
0xdf1f: xor      ebx, ebx
0xdf21: lea      r15, [rip + 0x13188]
0xdf28: xor      edx, edx
0xdf2a: mov      dword ptr [rbp - 0x74], esi
0xdf2d: jmp      0xdf48
0xdf48
0xdf2f: nop
0xdf30: cmp      al, 9
0xdf32: je       0xdf80
0xdf34: add      dword ptr [rbp - 0x74], 1
0xdf38: mov      byte ptr [r13 + r12], al
0xdf3d: movsxd   rdx, ebx
0xdf40: movzx    eax, byte ptr [rcx + rdx]
0xdf44: test     al, al
0xdf46: je       0xdfc0
0xdf48: movsxd   r12, dword ptr [rbp - 0x74]
0xdf4c: add      ebx, 1
0xdf4f: cmp      al, 0x5c
0xdf51: jne      0xdf30
0xdf53: lea      rdx, [rcx + rdx + 1]
0xdf58: movzx    eax, byte ptr [rdx]
0xdf5b: test     al, al
0xdf5d: je       0xe320
0xdf63: sub      eax, 0x5c
0xdf66: cmp      al, 0x17
0xdf68: ja       0xe0d8
0xdf6e: movzx    eax, al
0xdf71: movsxd   rax, dword ptr [r15 + rax*4]
0xdf75: add      rax, r15
0xdf78: jmp      rax
rax
->->->                                indirect JMP
0x8d38: mov      eax, dword ptr [rsp + 0x24]
0x8d3c: add      eax, 1
0x8d3f: mov      byte ptr [rbx], 0
0x8d42: jmp      0x8c4b
0x8c4b
0x8d47: nop      word ptr [rax + rax]

```

```

0x8d50: mov     rdi, r15
0x8d53: call    0xf510
0x8d58: cmp     eax, 4
0x8d5b: je      0x8c88
0x8d61: mov     rdi, qword ptr [rsp + 0x50]
0x8d66: test    rdi, rdi
0x8d69: je      0x8d70
0x8d6b: call    0x18d10
0x8d70: mov     qword ptr [r12], 0
0x8d78: mov     eax, 0xffffffff
0x8d7d: jmp     0x8c4b
0x8c4b:
0x8d82: nop     word ptr [rax + rax]
0x8d88: mov     rdi, r15
0x8d8b: call    0xf510
0x8d90: cmp     eax, 3
0x8d93: je      0x8c88
0x8d99: jmp     0x8d61
0x8d61:
0x8d9b: nop     dword ptr [rax + rax]
0x8da0: mov     rdi, qword ptr [r15 + 0x18]
0x8da4: call    0x49b0
0x8da9: lea     rdi, [rax + 3]
0x8dad: call    0x1d670
0x8db2: mov     r8, qword ptr [r15 + 0x18]
0x8db6: mov     qword ptr [rsp + 0x50], rax
0x8dbb: mov     rdi, rax
0x8dbe: jmp     0x8caf
0x8caf:
0x8dc3: nop     dword ptr [rax + rax]
0x8dc8: mov     rdi, r15
0x8dcb: call    0xf4b0
0x8dd0: mov     edi, eax
0x8dd2: mov     r14d, eax
0x8dd5: call    0x4970
0x8dda: test    rax, rax
0x8ddd: je      0x8eab
0x8de3: mov     rdi, qword ptr [rax]
0x8de6: call    0x1d7a0
0x8deb: mov     qword ptr [rsp + 0x50], rax
0x8df0: jmp     0x8cc9
0x8cc9:
0x8df5: nop     dword ptr [rax]
0x8df8: mov     edi, 0x10
0x8dfd: call    0x1d670
0x8e02: mov     rdi, r15
0x8e05: mov     qword ptr [rsp + 0x50], rax
0x8e0a: call    0xf4f0

```

```

0x8e0f: mov     rdi, qword ptr [rsp + 0x50]
0x8e14: mov     esi, 1
0x8e19: lea     rcx, [rip + 0x1771c]
0x8e20: and     eax, 0xfff
0x8e25: mov     rdx, -1
0x8e2c: mov     r8d, eax
0x8e2f: xor     eax, eax
0x8e31: call    0x4f10
0x8e36: jmp     0x8cc9
0x8cc9
0x8e3b: nop     dword ptr [rax + rax]
0x8e40: xor     eax, eax
0x8e42: cmp     dl, 0x69
0x8e45: mov     edx, dword ptr [rip + 0x21aad]
0x8e4b: setne   al
0x8e4e: add     eax, 1
0x8e51: mov     dword ptr [rsp + 0x24], eax
0x8e55: test    edx, edx
0x8e57: je      0x9198
0x8e5d: mov     eax, dword ptr [r15 + 0x70]
0x8e61: test    eax, eax
0x8e63: je      0x9198
0x8e69: mov     edi, 0x15
0x8e6e: call    0x1d670
0x8e73: movdqa  xmm0, xmmword ptr [rip + 0x17d35]
0x8e7b: mov     qword ptr [rsp + 0x50], rax
0x8e80: mov     byte ptr [rax + 0x10], 0
0x8e84: movups  xmmword ptr [rax], xmm0
0x8e87: jmp     0x8cc9
0x8cc9
0x8e8c: nop     dword ptr [rax]
0x8e90: mov     rdi, r15
0x8e93: call    0xf4d0
0x8e98: mov     edi, eax
0x8e9a: mov     r14d, eax
0x8e9d: call    0x4a10
0x8ea2: test    rax, rax
0x8ea5: jne     0x8de3
0x8eab: mov     edi, 0x10
0x8eb0: call    0x1d670
0x8eb5: mov     r8d, r14d
0x8eb8: mov     esi, 1
0x8ebd: lea     rcx, [rip + 0x17678]
0x8ec4: mov     qword ptr [rsp + 0x50], rax
0x8ec9: mov     rdi, rax
0x8ecc: mov     rdx, -1
0x8ed3: xor     eax, eax
0x8ed5: call    0x4f10

```

```

0x8eda: jmp      0x8cc9
0x8cc9
0x8edf: nop
0x8ee0: mov     rdi, r15
0x8ee3: call    0xf510
0x8ee8: cmp     eax, 2
0x8eeb: je      0x8c88
0x8ef1: jmp     0x8d61
0x8d61
0x8ef6: nop     word ptr cs:[rax + rax]
0x8f00: mov     rdi, r15
0x8f03: call    0xf510
0x8f08: test    eax, eax
0x8f0a: je      0x8c88
0x8f10: jmp     0x8d61
0x8d61
0x8f15: nop     dword ptr [rax]
0x8f18: mov     rdi, qword ptr [r15 + 0x18]
0x8f1c: mov     esi, 0x2f
0x8f21: lea     r14, [rip + 0x17604]
0x8f28: call    0x4a20
0x8f2d: mov     edi, 8
0x8f32: cmp     byte ptr [rax + 1], 0
0x8f36: je      0x8c9f
0x8f3c: lea     r14, [rax + 1]
0x8f40: jmp     0x8c93
0x8c93
0x8f45: mov     esi, dword ptr [rip + 0x219ad]
0x8f4b: test    esi, esi
0x8f4d: jne     0x8c28
0x8f53: cmp     byte ptr [r14 + 2], 0x7b
0x8f58: jne     0x8c38
0x8f5e: add     r14, 3
0x8f62: mov     esi, 0x7d
0x8f67: mov     rdi, r14
0x8f6a: call    0x4a00
0x8f6f: mov     rbp, rax
0x8f72: test    rax, rax
0x8f75: je      0x8c38
0x8f7b: mov     byte ptr [rax], 0
0x8f7e: mov     edx, 2
0x8f83: lea     rsi, [rip + 0x175a8]
0x8f8a: mov     rdi, r14
0x8f8d: call    0x8070
0x8f92: mov     byte ptr [rbp], 0x7d
0x8f96: cmp     al, 0x79
0x8f98: je      0x8cc9
0x8f9e: jmp     0x8d61

```

```

0x8d61
0x8fa3: nop        dword ptr [rax + rax]
0x8fa8: mov         ecx, dword ptr [rip + 0x2194a]
0x8fae: test        ecx, ecx
0x8fb0: jne         0x8c28
0x8fb6: cmp         byte ptr [r14 + 2], 0x7b
0x8fbb: jne         0x8c38
0x8fc1: add         r14, 3
0x8fc5: mov         esi, 0x2c
0x8fca: mov         rdi, r14
0x8fcd: call        0x4a00
0x8fd2: mov         rbp, rax
0x8fd5: test        rax, rax
0x8fd8: je          0x8c38
0x8fde: mov         byte ptr [rax], 0
0x8fe1: mov         rax, qword ptr [rsp + 0x18]
0x8fe6: lea         r15, [rsp + 0x48]
0x8feb: mov         rcx, qword ptr [rsp + 0x10]
0x8ff0: mov         rdx, qword ptr [rsp + 8]
0x8ff5: mov         rsi, r15
0x8ff8: mov         dword ptr [rip + 0x218f6], 1
0x9002: mov         r8, qword ptr [rax + 8]
0x9006: mov         qword ptr [rax + 8], r14
0x900a: mov         rdi, rax
0x900d: mov         r14, rax
0x9010: xor         eax, eax
0x9012: mov         qword ptr [rsp + 0x28], r8
0x9017: call        0x8b50
0x901c: mov         r8, qword ptr [rsp + 0x28]
0x9021: mov         dword ptr [rip + 0x218cd], 0
0x902b: mov         qword ptr [r14 + 8], r8
0x902f: test        eax, eax
0x9031: jle         0x9290
0x9037: mov         rax, qword ptr [rsp + 0x18]
0x903c: mov         rdi, qword ptr [rax]
0x903f: call        0x49b0
0x9044: mov         r14, qword ptr [rsp + 0x48]
0x9049: mov         qword ptr [rsp + 0x28], rax
0x904e: mov         rdi, r14
0x9051: call        0x49b0
0x9056: mov         rdx, qword ptr [rsp + 0x28]
0x905b: lea         rdi, [rdx + rax + 0x11]
0x9060: call        0x1d670
0x9065: mov         r9, r14
0x9068: mov         esi, 1
0x906d: lea         rcx, [rip + 0x174c1]
0x9074: mov         r10, rax
0x9077: mov         rax, qword ptr [rsp + 0x18]

```



```

0x907c: mov     rdx, -1
0x9083: mov     rdi, r10
0x9086: mov     qword ptr [rsp + 0x28], r10
0x908b: mov     r8, qword ptr [rax]
0x908e: xor     eax, eax
0x9090: call    0x4f10
0x9095: mov     rdi, r14
0x9098: lea     r14, [rbp + 1]
0x909c: call    0x18d10
0x90a1: mov     byte ptr [rbp], 0x2c
0x90a5: mov     esi, 0x7d
0x90aa: mov     rdi, r14
0x90ad: mov     qword ptr [rsp + 0x48], 0
0x90b6: call    0x4a00
0x90bb: mov     rbp, rax
0x90be: test    rax, rax
0x90c1: je      0x8c38
0x90c7: mov     byte ptr [rax], 0
0x90ca: mov     rdi, qword ptr [rsp + 0x28]
0x90cf: call    0x49b0
0x90d4: mov     r10, qword ptr [rsp + 0x28]
0x90d9: cmp     rax, 0x32
0x90dd: jbe     0x90e4
0x90df: mov     byte ptr [r10 + 0x32], 0
0x90e4: mov     rax, qword ptr [rsp + 0x18]
0x90e9: mov     rcx, qword ptr [rsp + 0x10]
0x90ee: mov     rsi, r15
0x90f1: mov     qword ptr [rsp + 0x30], r10
0x90f6: mov     dword ptr [rip + 0x217f8], 1
0x9100: mov     rdx, qword ptr [rsp + 8]
0x9105: mov     r8, qword ptr [rax + 8]
0x9109: mov     qword ptr [rax + 8], r14
0x910d: mov     r15, rax
0x9110: mov     rdi, rax
0x9113: xor     eax, eax
0x9115: mov     qword ptr [rsp + 0x28], r8
0x911a: call    0x8b50
0x911f: mov     r8, qword ptr [rsp + 0x28]
0x9124: mov     r10, qword ptr [rsp + 0x30]
0x9129: mov     dword ptr [rip + 0x217c5], 0
0x9133: test    eax, eax
0x9135: mov     qword ptr [r15 + 8], r8
0x9139: jle     0x9266
0x913f: mov     rax, qword ptr [rsp + 0x18]
0x9144: mov     r15, qword ptr [rsp + 0x48]
0x9149: mov     rdi, r10
0x914c: lea     rsi, [rsp + 0x50]
0x9151: mov     qword ptr [rsp + 0x28], r10

```

```

0x9156: mov     rcx, qword ptr [rax + 0x28]
0x915a: mov     rdx, r15
0x915d: call    0x83e0
0x9162: mov     rdi, qword ptr [rsp + 0x28]
0x9167: mov     r14, rax
0x916a: call    0x18d10
0x916f: mov     rdi, r15
0x9172: call    0x18d10
0x9177: mov     byte ptr [rbp], 0x7d
0x917b: mov     qword ptr [rsp + 0x48], 0
0x9184: test    r14, r14
0x9187: jne     0x8cc9
0x918d: jmp     0x8d61
0x8d61
0x9192: nop     word ptr [rax + rax]
0x9198: mov     qword ptr [rsp + 0x50], 0
0x91a1: mov     rdi, r15
0x91a4: xor     r14d, r14d
0x91a7: call    0x100c0
0x91ac: mov     qword ptr [rsp + 0x28], rbx
0x91b1: mov     qword ptr [rsp + 0x30], rbp
0x91b6: mov     qword ptr [rsp + 0x38], r12
0x91bb: jmp     0x9232
0x9232
0x91bd: nop     dword ptr [rax]
0x91c0: cdqe
0x91c2: lea     rcx, [rax + rax*4]
0x91c6: mov     rax, qword ptr [r15 + 0x38]
0x91ca: shl     rcx, 4
0x91ce: mov     rdi, qword ptr [rax + rcx]
0x91d2: mov     r12, rcx
0x91d5: call    0x49b0
0x91da: mov     rdi, qword ptr [rsp + 0x50]
0x91df: lea     rbp, [rax + r14]
0x91e3: lea     rsi, [rbp + 4]
0x91e7: lea     rbx, [rbp + 3]
0x91eb: call    0x1d690
0x91f0: mov     qword ptr [rsp + 0x50], rax
0x91f5: mov     byte ptr [rax + r14], 0x22
0x91fa: mov     rax, qword ptr [r15 + 0x38]
0x91fe: mov     rsi, qword ptr [rax + r12]
0x9202: mov     rax, qword ptr [rsp + 0x50]
0x9207: lea     rdi, [rax + r14 + 1]
0x920c: mov     r14, rbx
0x920f: call    0x48b0
0x9214: mov     rax, qword ptr [rsp + 0x50]
0x9219: mov     byte ptr [rax + rbp + 1], 0x22
0x921e: mov     rax, qword ptr [rsp + 0x50]

```

```

0x9223: mov     byte ptr [rax + rbp + 2], 0x20
0x9228: mov     rax, qword ptr [rsp + 0x50]
0x922d: mov     byte ptr [rax + rbp + 3], 0
0x9232: mov     rdi, r15
0x9235: call    0x100e0
0x923a: cmp     eax, -1
0x923d: jne     0x91c0
0x923f: mov     rbx, qword ptr [rsp + 0x28]
0x9244: mov     rbp, qword ptr [rsp + 0x30]
0x9249: mov     r12, qword ptr [rsp + 0x38]
0x924e: test    r14, r14
0x9251: jne     0x8cc9
0x9257: jmp     0x8d61
0x8d61
0x925c: mov     eax, 1
0x9261: jmp     0x8d3f
0x8d3f
0x9266: mov     byte ptr [rbp], 0x7d
0x926a: mov     rdi, r10
0x926d: mov     dword ptr [rsp + 8], eax
0x9271: call    0x18d10
0x9276: mov     eax, dword ptr [rsp + 8]
0x927a: test    eax, eax
0x927c: je      0x8c38
0x9282: jmp     0x8d61
0x8d61
0x9287: nop     word ptr [rax + rax]
0x9290: mov     byte ptr [rbp], 0x2c
0x9294: je      0x8c38
0x929a: jmp     0x8d61
0x8d61
0x929f: call    0x49d0
0x92a4: nop     word ptr cs:[rax + rax]
0x92af: nop
0x92b0: endbr64
0x92b4: push    r13
0x92b6: mov     r13, rdx
0x92b9: push    r12
0x92bb: mov     r12, rsi
0x92be: push    rbp
0x92bf: mov     rbp, rdi
0x92c2: call    0x47e0
0x92c7: test    rax, rax
0x92ca: je      0x92d8
0x92cc: pop     rbp
0x92cd: pop     r12
0x92cf: pop     r13
0x92d1: ret

```

```

0x92d2: nop      word ptr [rax + rax]
0x92d8: test     r12, r12
0x92db: je       0x9300
0x92dd: mov      rdi, r12
0x92e0: call     0x47e0
0x92e5: mov      rsi, rax
0x92e8: test     rax, rax
0x92eb: je       0x9300
0x92ed: mov      rdi, rbp
0x92f0: pop      rbp
0x92f1: pop      r12
0x92f3: pop      r13
0x92f5: jmp      0x18380
0x18380
0x92fa: nop      word ptr [rax + rax]
0x9300: mov      rsi, r13
0x9303: mov      rdi, rbp
0x9306: pop      rbp
0x9307: pop      r12
0x9309: pop      r13
0x930b: jmp      0x18380
0x18380
0x9310: endbr64
0x9314: push     r15
0x9316: push     r14
0x9318: push     r13
0x931a: push     r12
0x931c: push     rbp
0x931d: push     rbx
0x931e: mov      ebx, edi
0x9320: sub      rsp, 0xb8
0x9327: mov      qword ptr [rsp + 0x18], rsi
0x932c: mov      rax, qword ptr fs:[0x28]
0x9335: mov      qword ptr [rsp + 0xa8], rax
0x933d: xor      eax, eax
0x933f: mov      dword ptr [rsi], 0
0x9345: cmp      edi, 0x7ff
0x934b: jg       0x95d8
0x9351: lea      rax, [rsp + 0x20]
0x9356: lea      r12, [rsp + 0x50]
0x935b: mov      qword ptr [rsp + 8], rax
0x9360: lea      r13, [rip + 0x2089e]
0x9367: jmp      0x94df
0x94df
0x936c: nop      dword ptr [rax]
0x9370: mov      rdi, r12
0x9373: call     0xbd80
0x9378: test     rax, rax

```

```

0x937b: je      0x9508
0x9381: cmp     byte ptr [rax], 0
0x9384: je      0x94d0
0x938a: mov     edx, 0x50
0x938f: mov     rsi, rax
0x9392: mov     rdi, r12
0x9395: call    0x4cf0
0x939a: mov     dword ptr [rsp + 4], 0
0x93a2: mov     edi, 1
0x93a7: mov     esi, 0x30
0x93ac: call    0x1d740
0x93b1: mov     rdi, qword ptr [rsp + 0x20]
0x93b6: mov     rbp, rax
0x93b9: test    rdi, rdi
0x93bc: je      0x9560
0x93c2: call    0x1d7a0
0x93c7: mov     edi, 8
0x93cc: mov     qword ptr [rbp], rax
0x93d0: call    0x19100
0x93d5: mov     rdi, qword ptr [rsp + 0x30]
0x93da: mov     qword ptr [rbp + 0x28], rax
0x93de: test    rdi, rdi
0x93e1: je      0x93ec
0x93e3: call    0x1d7a0
0x93e8: mov     qword ptr [rbp + 0x10], rax
0x93ec: mov     rdi, qword ptr [rsp + 0x28]
0x93f1: test    rdi, rdi
0x93f4: je      0x9474
0x93f6: call    0x1d7a0
0x93fb: mov     qword ptr [rbp + 8], rax
0x93ff: mov     r14, qword ptr [rsp + 0x38]
0x9404: mov     eax, 1
0x9409: test    r14, r14
0x940c: je      0x9421
0x940e: call    0x4ef0
0x9413: movsx   rdx, byte ptr [r14]
0x9417: mov     rax, qword ptr [rax]
0x941a: cmp     dword ptr [rax + rdx*4], 0x79
0x941e: sete    al
0x9421: mov     byte ptr [rbp + 0x18], al
0x9424: mov     rdx, qword ptr [rsp + 0x40]
0x9429: test    rdx, rdx
0x942c: je      0x944c
0x942e: mov     qword ptr [rsp + 0x10], rdx
0x9433: call    0x4ef0
0x9438: mov     rdx, qword ptr [rsp + 0x10]
0x943d: mov     rax, qword ptr [rax]
0x9440: movsx   rdx, byte ptr [rdx]

```

```

0x9444: cmp     dword ptr [rax + rdx*4], 0x79
0x9448: sete    byte ptr [rbp + 0x19]
0x944c: mov     rdx, qword ptr [rsp + 0x48]
0x9451: test    rdx, rdx
0x9454: je      0x9474
0x9456: mov     qword ptr [rsp + 0x10], rdx
0x945b: call    0x4ef0
0x9460: mov     rdx, qword ptr [rsp + 0x10]
0x9465: mov     rax, qword ptr [rax]
0x9468: movsx   rdx, byte ptr [rdx]
0x946c: cmp     dword ptr [rax + rdx*4], 0x79
0x9470: sete    byte ptr [rbp + 0x1a]
0x9474: call    0x4f00
0x9479: mov     r15, qword ptr [rbp]
0x947d: mov     rax, qword ptr [rax]
0x9480: movsx   rdx, byte ptr [r15]
0x9484: test    byte ptr [rax + rdx*2 + 1], 2
0x9489: jne     0x9518
0x948f: mov     rdi, r12
0x9492: call    0x1d7a0
0x9497: cmp     byte ptr [rbp + 0x1b], 0
0x949b: mov     qword ptr [rbp + 0x20], rax
0x949f: jne     0x94ac
0x94a1: cmp     qword ptr [rbp + 8], 0
0x94a6: je      0x9590
0x94ac: mov     eax, dword ptr [rsp + 4]
0x94b0: test    eax, eax
0x94b2: je      0x94c5
0x94b4: mov     rdi, r12
0x94b7: call    0x9f00
0x94bc: test    rax, rax
0x94bf: je      0x95a0
0x94c5: mov     rsi, rbp
0x94c8: mov     rdi, r12
0x94cb: call    0xaf70
0x94d0: add     ebx, 1
0x94d3: cmp     ebx, 0x800
0x94d9: je      0x9580
0x94df: mov     rsi, qword ptr [rsp + 8]
0x94e4: xor     ecx, ecx
0x94e6: mov     edx, 6
0x94eb: mov     rdi, r12
0x94ee: call    0x17320
0x94f3: movzx   eax, byte ptr [rsp + 0x50]
0x94f8: test    al, al
0x94fa: je      0x95d8
0x9500: cmp     al, 0x5e
0x9502: jne     0x9370

```

```

0x9508: mov     dword ptr [rsp + 4], 1
0x9510: jmp     0x93a2
0x93a2:
0x9515: nop     dword ptr [rax]
0x9518: lea     r14, [rip + 0x1fb01]
0x951f: jmp     0x9535
0x9535:
0x9521: nop     dword ptr [rax]
0x9528: add     r14, 0x23
0x952c: cmp     r13, r14
0x952f: je      0x948f
0x9535: mov     rsi, r14
0x9538: mov     rdi, r15
0x953b: call    0x4b30
0x9540: test    eax, eax
0x9542: jne     0x9528
0x9544: mov     rdi, r15
0x9547: call    0x18d10
0x954c: mov     qword ptr [rbp], r14
0x9550: mov     byte ptr [rbp + 0x1b], 1
0x9554: jmp     0x948f
0x948f:
0x9559: nop     dword ptr [rax]
0x9560: mov     rdi, rax
0x9563: add     ebx, 1
0x9566: call    0x18d10
0x956b: cmp     ebx, 0x800
0x9571: jne     0x94df
0x9577: nop     word ptr [rax + rax]
0x9580: mov     eax, 0x800
0x9585: jmp     0x95da
0x95da:
0x9587: nop     word ptr [rax + rax]
0x9590: cmp     qword ptr [rbp + 0x10], 0
0x9595: jne     0x94ac
0x959b: jmp     0x94d0
0x94d0:
0x95a0: mov     rcx, qword ptr [rip + 0x21341]
0x95a7: mov     rdi, qword ptr [rip + 0x212d2]
0x95ae: mov     r8, r12
0x95b1: xor     eax, eax
0x95b3: lea     rdx, [rip + 0x1720e]
0x95ba: mov     esi, 1
0x95bf: call    0x4e60
0x95c4: mov     rax, qword ptr [rsp + 0x18]
0x95c9: add     dword ptr [rax], 1
0x95cc: jmp     0x94d0
0x94d0:

```

```

0x95d1: nop      dword ptr [rax]
0x95d8: mov      eax, ebx
0x95da: mov      rdx, qword ptr [rsp + 0xa8]
0x95e2: sub      rdx, qword ptr fs:[0x28]
0x95eb: jne      0x95ff
0x95ed: add      rsp, 0xb8
0x95f4: pop      rbx
0x95f5: pop      rbp
0x95f6: pop      r12
0x95f8: pop      r13
0x95fa: pop      r14
0x95fc: pop      r15
0x95fe: ret
0x95ff: call     0x49d0
0x9604: nop      word ptr cs:[rax + rax]
0x960f: nop
0x9610: endbr64
0x9614: sub      rsp, 8
0x9618: xor      edi, edi
0x961a: call     0x9b00
0x961f: xor      eax, eax
0x9621: call     0xa6c0
0x9626: mov      rdi, qword ptr [rip + 0x2129b]
0x962d: add      rsp, 8
0x9631: jmp      0xbb40
0xbb40:
0x9636: nop      word ptr cs:[rax + rax]
0x9640: endbr64
0x9644: push     rbp
0x9645: lea      rbp, [rip + 0x205b9]
0x964c: push     rbx
0x964d: mov      rdi, rbp
0x9650: sub      rsp, 0x48
0x9654: mov      rax, qword ptr fs:[0x28]
0x965d: mov      qword ptr [rsp + 0x38], rax
0x9662: xor      eax, eax
0x9664: call     0x49b0
0x9669: mov      rdi, qword ptr [rip + 0x21240]
0x9670: mov      rbx, rax
0x9673: call     0x11100
0x9678: mov      edx, 0x33
0x967d: mov      rsi, rsp
0x9680: sub      edx, ebx
0x9682: mov      rdi, rax
0x9685: call     0x182b0
0x968a: mov      rsi, rbp
0x968d: mov      rdi, rax
0x9690: call     0x4df0

```



```

0x9695: mov     rdi, rax
0x9698: call    0xd540
0x969d: mov     rax, qword ptr [rsp + 0x38]
0x96a2: sub     rax, qword ptr fs:[0x28]
0x96ab: jne     0x96b4
0x96ad: add     rsp, 0x48
0x96b1: pop     rbx
0x96b2: pop     rbp
0x96b3: ret
0x96b4: call    0x49d0
0x96b9: nop     dword ptr [rax]
0x96c0: endbr64
0x96c4: sub     rsp, 8
0x96c8: mov     r8d, 1
0x96ce: mov     rdx, qword ptr [rip + 0x211db]
0x96d5: mov     rdi, qword ptr [rip + 0x211cc]
0x96dc: mov     rcx, -1
0x96e3: mov     esi, 0x22
0x96e8: call    0x13e30
0x96ed: mov     r8d, 1
0x96f3: mov     rdx, qword ptr [rip + 0x211ae]
0x96fa: mov     rdi, qword ptr [rip + 0x211af]
0x9701: mov     rcx, -1
0x9708: mov     esi, 0x22
0x970d: add     rsp, 8
0x9711: jmp     0x13e30
0x13e30:
0x9716: nop     word ptr cs:[rax + rax]
0x9720: endbr64
0x9724: sub     rsp, 8
0x9728: mov     rdx, qword ptr [rip + 0x211b9]
0x972f: mov     edi, 1
0x9734: xor     eax, eax
0x9736: lea     rsi, [rip + 0x170b3]
0x973d: call    0x4d20
0x9742: lea     rdi, [rip + 0x170cf]
0x9749: call    0x48d0
0x974e: lea     rdi, [rip + 0x170eb]
0x9755: call    0x48d0
0x975a: lea     rdi, [rip + 0x16dde]
0x9761: call    0x48d0
0x9766: lea     rdi, [rip + 0x170fb]
0x976d: call    0x48d0
0x9772: lea     rdi, [rip + 0x17117]
0x9779: call    0x48d0
0x977e: lea     rdi, [rip + 0x1713b]
0x9785: add     rsp, 8
0x9789: jmp     0x48d0

```

```

0x48d0
0x978e: nop
0x9790: cmp     dword ptr [rip + 0x21ba9], 1
0x9797: push    rbx
0x9798: mov     ebx, edi
0x979a: je      0x9830
0x97a0: cmp     ebx, 1
0x97a3: je      0x984a
0x97a9: mov     rdi, qword ptr [rip + 0x20598]
0x97b0: test    rdi, rdi
0x97b3: je      0x97c6
0x97b5: lea     rdx, [rip + 0x2b4]
0x97bc: mov     esi, 1
0x97c1: call    0x4820
0x97c6: movzx   edx, byte ptr [rip + 0x21b83]
0x97cd: mov     dword ptr [rip + 0x21b75], 0
0x97d7: mov     dword ptr [rip + 0x21b67], 0
0x97e1: mov     eax, edx
0x97e3: and     eax, 0x7f
0x97e6: and     edx, 0x40
0x97e9: mov     byte ptr [rip + 0x21b61], al
0x97ef: jne     0x9810
0x97f1: mov     dword ptr [rip + 0x21b45], 1
0x97fb: and     eax, 0x7f
0x97fe: shl     ebx, 7
0x9801: or      eax, ebx
0x9803: mov     byte ptr [rip + 0x21b47], al
0x9809: pop     rbx
0x980a: ret
0x980b: nop     dword ptr [rax + rax]
0x9810: mov     dword ptr [rip + 0x21b2a], 0
0x981a: mov     edi, 1
0x981f: call    0x9870
0x9824: movzx   eax, byte ptr [rip + 0x21b25]
0x982b: jmp     0x97f1
0x97f1
0x982d: nop     dword ptr [rax]
0x9830: movzx   eax, byte ptr [rip + 0x21b19]
0x9837: shr     al, 7
0x983a: movzx   eax, al
0x983d: cmp     eax, edi
0x983f: je      0x9809
0x9841: cmp     ebx, 1
0x9844: jne     0x97a9
0x984a: mov     rdi, qword ptr [rip + 0x20517]
0x9851: test    rdi, rdi
0x9854: je      0x9824
0x9856: lea     rdx, [rip + 0x213]

```

```

0x985d: mov     esi, 1
0x9862: call    0x4820
0x9867: movzx   eax, byte ptr [rip + 0x21ae2]
0x986e: jmp     0x97f1
0x97f1
0x9870: cmp     dword ptr [rip + 0x21acd], 1
0x9877: push    rbx
0x9878: mov     ebx, edi
0x987a: je      0x98e0
0x987c: cmp     ebx, 1
0x987f: je      0x98f6
0x9881: mov     rdi, qword ptr [rip + 0x204c0]
0x9888: test    rdi, rdi
0x988b: je      0x989e
0x988d: lea     rdx, [rip + 0x1dc]
0x9894: mov     esi, 1
0x9899: call    0x4820
0x989e: movzx   eax, byte ptr [rip + 0x21aab]
0x98a5: mov     dword ptr [rip + 0x21a9d], 0
0x98af: mov     dword ptr [rip + 0x21a8f], 0
0x98b9: and     eax, 0xffffffffbf
0x98bc: mov     byte ptr [rip + 0x21a8e], al
0x98c2: test    al, al
0x98c4: js      0x9920
0x98c6: mov     dword ptr [rip + 0x21a74], 1
0x98d0: and     eax, 0xffffffffbf
0x98d3: shl     ebx, 6
0x98d6: or      eax, ebx
0x98d8: mov     byte ptr [rip + 0x21a72], al
0x98de: pop     rbx
0x98df: ret
0x98e0: movzx   eax, byte ptr [rip + 0x21a69]
0x98e7: sar     eax, 6
0x98ea: and     eax, 1
0x98ed: cmp     eax, edi
0x98ef: je      0x98de
0x98f1: cmp     ebx, 1
0x98f4: jne     0x9881
0x98f6: mov     rdi, qword ptr [rip + 0x2048b]
0x98fd: test    rdi, rdi
0x9900: je      0x9934
0x9902: lea     rdx, [rip + 0x167]
0x9909: mov     esi, 1
0x990e: call    0x4820
0x9913: movzx   eax, byte ptr [rip + 0x21a36]
0x991a: jmp     0x98c6
0x98c6
0x991c: nop     dword ptr [rax]

```

```

0x9920: mov     dword ptr [rip + 0x21a16], 0
0x992a: mov     edi, 1
0x992f: call    0x9790
0x9934: movzx   eax, byte ptr [rip + 0x21a15]
0x993b: jmp     0x98c6
0x98c6
0x993d: nop     dword ptr [rax]
0x9940: push    r12
0x9942: mov     edx, 5
0x9947: lea     rsi, [rip + 0x17293]
0x994e: mov     r12d, 1
0x9954: push    rbp
0x9955: mov     rbp, rdi
0x9958: sub     rsp, 8
0x995c: call    0x4890
0x9961: test    eax, eax
0x9963: je      0x99d3
0x9965: mov     edx, 4
0x996a: lea     rsi, [rip + 0x1725f]
0x9971: mov     rdi, rbp
0x9974: call    0x4890
0x9979: test    eax, eax
0x997b: je      0x99d3
0x997d: mov     edx, 9
0x9982: lea     rsi, [rip + 0x1724c]
0x9989: mov     rdi, rbp
0x998c: call    0x4890
0x9991: test    eax, eax
0x9993: je      0x99d3
0x9995: lea     rsi, [rip + 0x17243]
0x999c: mov     rdi, rbp
0x999f: call    0x4b30
0x99a4: test    eax, eax
0x99a6: je      0x99d3
0x99a8: lea     rsi, [rip + 0x17238]
0x99af: mov     rdi, rbp
0x99b2: call    0x4b30
0x99b7: test    eax, eax
0x99b9: je      0x99d3
0x99bb: lea     rsi, [rip + 0x1722b]
0x99c2: mov     rdi, rbp
0x99c5: xor     r12d, r12d
0x99c8: call    0x4b30
0x99cd: test    eax, eax
0x99cf: sete    r12b
0x99d3: add     rsp, 8
0x99d7: mov     eax, r12d
0x99da: pop     rbp

```

```

0x99db: pop      r12
0x99dd: ret
0x99de: nop
0x99e0: push     r13
0x99e2: push     r12
0x99e4: mov      r12, qword ptr [rip + 0x20f8d]
0x99eb: push     rbp
0x99ec: test     r12, r12
0x99ef: je       0x9a58
0x99f1: mov      r13, rdi
0x99f4: jmp      0x9a13
0x9a13:
0x99f6: nop      word ptr cs:[rax + rax]
0x9a00: js       0x9a30
0x9a02: mov      r12, qword ptr [r12 + 8]
0x9a07: mov      qword ptr [rip + 0x20f6a], r12
0x9a0e: test     r12, r12
0x9a11: je       0x9a58
0x9a13: mov      rbp, qword ptr [r12]
0x9a17: mov      rdi, r13
0x9a1a: mov      rsi, rbp
0x9a1d: call     0x4b30
0x9a22: test     eax, eax
0x9a24: jne      0x9a00
0x9a26: mov      rax, r12
0x9a29: pop      rbp
0x9a2a: pop      r12
0x9a2c: pop      r13
0x9a2e: ret
0x9a2f: nop
0x9a30: mov      rdi, r13
0x9a33: call     0x49b0
0x9a38: mov      rsi, rbp
0x9a3b: mov      rdi, r13
0x9a3e: mov      rdx, rax
0x9a41: call     0x4890
0x9a46: pop      rbp
0x9a47: neg      eax
0x9a49: sbb      r12, r12
0x9a4c: mov      rax, r12
0x9a4f: pop      r12
0x9a51: pop      r13
0x9a53: ret
0x9a54: nop      dword ptr [rax]
0x9a58: mov      r12, -1
0x9a5f: pop      rbp
0x9a60: mov      rax, r12
0x9a63: pop      r12

```

```

0x9a65: pop      r13
0x9a67: ret
0x9a68: nop      dword ptr [rax + rax]
0x9a70: endbr64
0x9a74: movsxd   rdx, dword ptr [rip + 0x2148d]
0x9a7b: push     r12
0x9a7d: push     rbp
0x9a7e: mov      ebp, edi
0x9a80: push     rbx
0x9a81: cmp      edx, 0x400
0x9a87: je       0x9ab0
0x9a89: lea      eax, [rdx + 1]
0x9a8c: lea      r12, [rip + 0x2148d]
0x9a93: mov      byte ptr [r12 + rdx], bpl
0x9a97: pop      rbx
0x9a98: mov      dword ptr [rip + 0x2146a], eax
0x9a9e: pop      rbp
0x9a9f: mov      eax, 1
0x9aa4: pop      r12
0x9aa6: ret
0x9aa7: nop      word ptr [rax + rax]
0x9ab0: xor      ebx, ebx
0x9ab2: lea      r12, [rip + 0x21467]
0x9ab9: jmp      0x9acc
0x9acc:
0x9abb: nop      dword ptr [rax + rax]
0x9ac0: mov      edx, dword ptr [rip + 0x21442]
0x9ac6: add      ebx, eax
0x9ac8: cmp      edx, ebx
0x9aca: jle     0x9ae5
0x9acc: sub      edx, ebx
0x9ace: movsxd   rsi, ebx
0x9ad1: mov      edi, 1
0x9ad6: add      rsi, r12
0x9ad9: movsxd   rdx, edx
0x9adc: call    0x18da0
0x9ae1: test     eax, eax
0x9ae3: jns     0x9ac0
0x9ae5: mov      eax, 1
0x9aea: xor      edx, edx
0x9aec: jmp     0x9a93
0x9a93:
0x9aee: nop
0x9af0: endbr64
0x9af4: mov      dword ptr [rip + 0x21862], edi
0x9afa: ret
0x9afb: nop      dword ptr [rax + rax]
0x9b00: endbr64

```

```

0x9b04: push    r12
0x9b06: push    rbp
0x9b07: push    rbx
0x9b08: mov     ebx, edi
0x9b0a: sub     rsp, 0x50
0x9b0e: mov     rax, qword ptr fs:[0x28]
0x9b17: mov     qword ptr [rsp + 0x48], rax
0x9b1c: xor     eax, eax
0x9b1e: cmp     edi, 1
0x9b21: je      0x9b60
0x9b23: lea     rdx, [rip + 0x21896]
0x9b2a: mov     esi, 1
0x9b2f: mov     edi, 1
0x9b34: call    0x4d70
0x9b39: mov     dword ptr [rip + 0x20e19], ebx
0x9b3f: mov     rax, qword ptr [rsp + 0x48]
0x9b44: sub     rax, qword ptr fs:[0x28]
0x9b4d: jne     0x9c7e
0x9b53: add     rsp, 0x50
0x9b57: pop     rbx
0x9b58: pop     rbp
0x9b59: pop     r12
0x9b5b: ret
0x9b5c: nop     dword ptr [rax]
0x9b60: mov     rax, qword ptr [rip + 0x21889]
0x9b67: mov     rdx, qword ptr [rip + 0x21852]
0x9b6e: lea     r12, [rip + 0x2180b]
0x9b75: mov     esi, 1
0x9b7a: movdqa  xmm0, xmmword ptr [rip + 0x2183e]
0x9b82: movdqa  xmm1, xmmword ptr [rip + 0x21846]
0x9b8a: mov     edi, 1
0x9b8f: mov     qword ptr [rip + 0x2181a], rax
0x9b96: mov     eax, dword ptr [rip + 0x2185c]
0x9b9c: movaps  xmmword ptr [rip + 0x217dd], xmm0
0x9ba3: movdqa  xmm2, xmmword ptr [rip + 0x21835]
0x9bab: mov     dword ptr [rip + 0x21807], eax
0x9bb1: movabs  rax, 0xffffffffffffffa3c
0x9bbb: and     rax, rdx
0x9bbe: mov     rdx, r12
0x9bc1: movaps  xmmword ptr [rip + 0x217c8], xmm1
0x9bc8: mov     qword ptr [rip + 0x217b1], rax
0x9bcf: mov     eax, dword ptr [rip + 0x217b7]
0x9bd5: movaps  xmmword ptr [rip + 0x217c4], xmm2
0x9bdc: and     eax, 0xffffffff5
0x9bdf: mov     byte ptr [rip + 0x217bb], 0
0x9be6: mov     dword ptr [rip + 0x217a1], 7
0x9bf0: or      al, 0x81
0x9bf2: mov     dword ptr [rip + 0x21794], eax

```

```

0x9bf8: mov     eax, 0x100
0x9bfd: mov     word ptr [rip + 0x21792], ax
0x9c04: mov     qword ptr [rip + 0x2178a], 0x1a0000
0x9c0f: call    0x4d70
0x9c14: mov     rdi, r12
0x9c17: mov     r12, rsp
0x9c1a: call    0x4d10
0x9c1f: mov     edx, 1
0x9c24: mov     esi, 0x540a
0x9c29: mov     edi, 1
0x9c2e: mov     dword ptr [rip + 0x20cfc], eax
0x9c34: xor     eax, eax
0x9c36: call    0x4a70
0x9c3b: mov     esi, 1
0x9c40: mov     edi, 1
0x9c45: call    0x4db0
0x9c4a: mov     ebp, dword ptr [rip + 0x200d4]
0x9c50: mov     rsi, r12
0x9c53: mov     edi, 1
0x9c58: call    0x4d60
0x9c5d: mov     rdx, r12
0x9c60: mov     esi, 1
0x9c65: mov     edi, 1
0x9c6a: mov     byte ptr [rsp + 0x11], bpl
0x9c6f: mov     byte ptr [rsp + 0x12], 0
0x9c74: call    0x4d70
0x9c79: jmp     0x9b39
0x9b39
0x9c7e: call    0x49d0
0x9c83: nop     word ptr cs:[rax + rax]
0x9c8e: nop
0x9c90: endbr64
0x9c94: mov     eax, dword ptr [rip + 0x20cbe]
0x9c9a: ret
0x9c9b: nop     dword ptr [rax + rax]
0x9ca0: endbr64
0x9ca4: push    r12
0x9ca6: push    rbx
0x9ca7: mov     ebx, edi
0x9ca9: mov     edi, 1
0x9cae: sub     rsp, 0x58
0x9cb2: mov     rax, qword ptr fs:[0x28]
0x9cbb: mov     qword ptr [rsp + 0x48], rax
0x9cc0: xor     eax, eax
0x9cc2: mov     r12, rsp
0x9cc5: mov     rsi, r12
0x9cc8: call    0x4d60
0x9ccd: mov     rdx, r12

```



```

0x9cd0: mov     esi, 1
0x9cd5: mov     edi, 1
0x9cda: mov     byte ptr [rsp + 0x11], bl
0x9cde: mov     byte ptr [rsp + 0x12], 0
0x9ce3: call    0x4d70
0x9ce8: mov     dword ptr [rip + 0x20036], ebx
0x9cee: mov     rax, qword ptr [rsp + 0x48]
0x9cf3: sub     rax, qword ptr fs:[0x28]
0x9cfc: jne     0x9d06
0x9cfe: add     rsp, 0x58
0x9d02: pop     rbx
0x9d03: pop     r12
0x9d05: ret
0x9d06: call    0x49d0
0x9d0b: nop     dword ptr [rax + rax]
0x9d10: endbr64
0x9d14: mov     edx, dword ptr [rip + 0x211ee]
0x9d1a: test    edx, edx
0x9d1c: jle     0x9d66
0x9d1e: push    rbp
0x9d1f: xor     ebp, ebp
0x9d21: push    rbx
0x9d22: lea     rbx, [rip + 0x211f7]
0x9d29: sub     rsp, 8
0x9d2d: jmp     0x9d3c
0x9d3c:
0x9d2f: nop
0x9d30: mov     edx, dword ptr [rip + 0x211d2]
0x9d36: add     ebp, eax
0x9d38: cmp     edx, ebp
0x9d3a: jle     0x9d55
0x9d3c: sub     edx, ebp
0x9d3e: movsxd  rsi, ebp
0x9d41: mov     edi, 1
0x9d46: add     rsi, rbx
0x9d49: movsxd  rdx, edx
0x9d4c: call    0x18da0
0x9d51: test    eax, eax
0x9d53: jns     0x9d30
0x9d55: mov     dword ptr [rip + 0x211a9], 0
0x9d5f: add     rsp, 8
0x9d63: pop     rbx
0x9d64: pop     rbp
0x9d65: ret
0x9d66: mov     dword ptr [rip + 0x21198], 0
0x9d70: ret
0x9d71: nop     word ptr cs:[rax + rax]
0x9d7c: nop     dword ptr [rax]

```

```

0x9d80: endbr64
0x9d84: push    rbp
0x9d85: lea      rdx, [rip - 0x31c]
0x9d8c: push    rbx
0x9d8d: sub      rsp, 8
0x9d91: mov      esi, dword ptr [rip + 0x20c05]
0x9d97: mov      rdi, qword ptr [rip + 0x2004a]
0x9d9e: call     0x4820
0x9da3: mov      edx, dword ptr [rip + 0x2115f]
0x9da9: test     edx, edx
0x9dab: jle      0x9de5
0x9dad: xor      ebp, ebp
0x9daf: lea      rbx, [rip + 0x2116a]
0x9db6: jmp      0x9dcc
0x9dcc:
0x9db8: nop      dword ptr [rax + rax]
0x9dc0: mov      edx, dword ptr [rip + 0x21142]
0x9dc6: add      ebp, eax
0x9dc8: cmp      edx, ebp
0x9dca: jle      0x9de5
0x9dcc: sub      edx, ebp
0x9dce: movsxd   rsi, ebp
0x9dd1: mov      edi, 1
0x9dd6: add      rsi, rbx
0x9dd9: movsxd   rdx, edx
0x9ddc: call     0x18da0
0x9de1: test     eax, eax
0x9de3: jns      0x9dc0
0x9de5: mov      dword ptr [rip + 0x21119], 0
0x9def: add      rsp, 8
0x9df3: pop      rbx
0x9df4: pop      rbp
0x9df5: ret
0x9df6: nop      word ptr cs:[rax + rax]
0x9e00: endbr64
0x9e04: push    rbp
0x9e05: lea      rdx, [rip - 0x39c]
0x9e0c: push    rbx
0x9e0d: sub      rsp, 8
0x9e11: mov      eax, dword ptr [rip + 0x20b85]
0x9e17: mov      rdi, qword ptr [rip + 0x2010a]
0x9e1e: lea      esi, [rax - 1]
0x9e21: call     0x4820
0x9e26: mov      edx, dword ptr [rip + 0x210dc]
0x9e2c: test     edx, edx
0x9e2e: jle      0x9e65
0x9e30: xor      ebp, ebp
0x9e32: lea      rbx, [rip + 0x210e7]

```

```

0x9e39: jmp      0x9e4c
0x9e4c:
0x9e3b: nop      dword ptr [rax + rax]
0x9e40: mov      edx, dword ptr [rip + 0x210c2]
0x9e46: add      ebp, eax
0x9e48: cmp      edx, ebp
0x9e4a: jle      0x9e65
0x9e4c: sub      edx, ebp
0x9e4e: movsxd   rsi, ebp
0x9e51: mov      edi, 1
0x9e56: add      rsi, rbx
0x9e59: movsxd   rdx, edx
0x9e5c: call     0x18da0
0x9e61: test     eax, eax
0x9e63: jns      0x9e40
0x9e65: mov      dword ptr [rip + 0x21099], 0
0x9e6f: add      rsp, 8
0x9e73: pop      rbx
0x9e74: pop      rbp
0x9e75: ret
0x9e76: nop      word ptr cs:[rax + rax]
0x9e80: endbr64
0x9e84: push     rbp
0x9e85: lea      rdx, [rip - 0x41c]
0x9e8c: push     rbx
0x9e8d: sub      rsp, 8
0x9e91: mov      eax, dword ptr [rip + 0x20b05]
0x9e97: mov      rdi, qword ptr [rip + 0x200aa]
0x9e9e: lea      esi, [rax - 1]
0x9ea1: call     0x4820
0x9ea6: mov      edx, dword ptr [rip + 0x2105c]
0x9eac: test     edx, edx
0x9eae: jle      0x9ee5
0x9eb0: xor      ebp, ebp
0x9eb2: lea      rbx, [rip + 0x21067]
0x9eb9: jmp      0x9ecc
0x9ecc:
0x9ebb: nop      dword ptr [rax + rax]
0x9ec0: mov      edx, dword ptr [rip + 0x21042]
0x9ec6: add      ebp, eax
0x9ec8: cmp      edx, ebp
0x9eca: jle      0x9ee5
0x9ecc: sub      edx, ebp
0x9ece: movsxd   rsi, ebp
0x9ed1: mov      edi, 1
0x9ed6: add      rsi, rbx
0x9ed9: movsxd   rdx, edx
0x9edc: call     0x18da0

```

```

0x9ee1: test    eax, eax
0x9ee3: jns      0x9ec0
0x9ee5: mov     dword ptr [rip + 0x21019], 0
0x9eef: add     rsp, 8
0x9ef3: pop     rbx
0x9ef4: pop     rbp
0x9ef5: ret
0x9ef6: nop     word ptr cs:[rax + rax]
0x9f00: endbr64
0x9f04: mov     edx, dword ptr [rip + 0x214f2]
0x9f0a: movzx   eax, byte ptr [rdi]
0x9f0d: push    r14
0x9f0f: push    r13
0x9f11: mov     r13, rdi
0x9f14: push    r12
0x9f16: push    rbp
0x9f17: push    rbx
0x9f18: test    edx, edx
0x9f1a: jne     0x9fc8
0x9f20: cmp     al, 0x5e
0x9f22: jne     0x9fb8
0x9f28: lea     rbx, [r13 + 1]
0x9f2c: mov     rbp, r13
0x9f2f: lea     r14, [rip + 0x1fd6a]
0x9f36: jmp     0x9f56
0x9f56:
0x9f38: nop     dword ptr [rax + rax]
0x9f40: add     rbp, 1
0x9f44: mov     byte ptr [rbx - 1], al
0x9f47: mov     rdx, rbx
0x9f4a: add     rbx, 1
0x9f4e: movzx   eax, byte ptr [rbp]
0x9f52: test    al, al
0x9f54: je      0x9fa6
0x9f56: cmp     al, 0x5e
0x9f58: jne     0x9f40
0x9f5a: movzx   r12d, byte ptr [rbp + 1]
0x9f5f: test    r12b, r12b
0x9f62: je      0x9fb8
0x9f64: call    0x47c0
0x9f69: mov     r8, rax
0x9f6c: movzx   eax, r12b
0x9f70: mov     rdx, qword ptr [r8]
0x9f73: mov     eax, dword ptr [rdx + rax*4]
0x9f76: cmp     eax, 0x47
0x9f79: je      0x9fb8
0x9f7b: cmp     eax, 0x5a
0x9f7e: je      0x9fb8

```

```

0x9f80: and    r12d, 0x7f
0x9f84: add    rbp, 2
0x9f88: mov     rdx, rbx
0x9f8b: add     rbx, 1
0x9f8f: sub     r12d, 0x20
0x9f93: movsxd   r12, r12d
0x9f96: movzx    eax, byte ptr [r14 + r12]
0x9f9b: mov     byte ptr [rbx - 2], al
0x9f9e: movzx    eax, byte ptr [rbp]
0x9fa2: test    al, al
0x9fa4: jne     0x9f56
0x9fa6: mov     byte ptr [rdx], 0
0x9fa9: mov     rax, r13
0x9fac: pop     rbx
0x9fad: pop     rbp
0x9fae: pop     r12
0x9fb0: pop     r13
0x9fb2: pop     r14
0x9fb4: ret
0x9fb5: nop     dword ptr [rax]
0x9fb8: pop     rbx
0x9fb9: xor     eax, eax
0x9fbb: pop     rbp
0x9fbc: pop     r12
0x9fbe: pop     r13
0x9fc0: pop     r14
0x9fc2: ret
0x9fc3: nop     dword ptr [rax + rax]
0x9fc8: test    al, al
0x9fca: jne     0x9f28
0x9fd0: mov     rdx, rdi
0x9fd3: jmp     0x9fa6
0x9fa6:
0x9fd5: nop     word ptr cs:[rax + rax]
0x9fe0: endbr64
0x9fe4: push    r15
0x9fe6: push    r14
0x9fe8: mov     r14, rdi
0x9feb: push    r13
0x9fed: lea     r13, [rip + 0x16c12]
0x9ff4: push    r12
0x9ff6: lea     r12, [rip + 0x209c3]
0x9ffd: push    rbp
0x9ffe: lea     rbp, [rip + 0x16bef]
0xa005: push    rbx
0xa006: mov     rbx, rdi
0xa009: sub     rsp, 0x18
0xa00d: mov     rax, qword ptr fs:[0x28]

```

```

0xa016: mov     qword ptr [rsp + 8], rax
0xa01b: xor      eax, eax
0xa01d: mov     byte ptr [rip + 0x2099c], 0
0xa024: movzx   r15d, byte ptr [rdi]
0xa028: test    r15b, r15b
0xa02b: jne     0xa0c2
0xa031: jmp     0xa0f0
0xa0f0:
0xa036: nop     word ptr cs:[rax + rax]
0xa040: cmp     r15b, 0x20
0xa044: je      0xa148
0xa04a: cmp     r15b, 0x7f
0xa04e: je      0xa160
0xa054: cmp     r15b, 0xff
0xa058: je      0xa180
0xa05e: call    0x4f00
0xa063: mov     r8, rax
0xa066: movzx   eax, r15b
0xa06a: mov     rsi, qword ptr [r8]
0xa06d: test    byte ptr [rsi + rax*2], 2
0xa071: je      0xa120
0xa077: lea     edx, [r15 + 0x40]
0xa07b: lea     rsi, [rsp + 5]
0xa080: mov     rdi, r12
0xa083: mov     byte ptr [rsp + 5], 0x5e
0xa088: mov     byte ptr [rsp + 6], dl
0xa08c: mov     edx, 0x12c
0xa091: mov     byte ptr [rsp + 7], 0
0xa096: call    0x47d0
0xa09b: add     r14, 1
0xa09f: cmp     byte ptr [r14], 0
0xa0a3: je      0xa0f0
0xa0a5: cmp     rbx, r14
0xa0a8: je      0xa0be
0xa0aa: mov     edx, 0x12c
0xa0af: lea     rsi, [rip + 0x1668f]
0xa0b6: mov     rdi, r12
0xa0b9: call    0x47d0
0xa0be: movzx   r15d, byte ptr [r14]
0xa0c2: cmp     r15b, 0x1b
0xa0c6: jne     0xa040
0xa0cc: mov     edx, 0x12c
0xa0d1: mov     rsi, rbp
0xa0d4: mov     rdi, r12
0xa0d7: add     r14, 1
0xa0db: call    0x47d0
0xa0e0: cmp     byte ptr [r14], 0
0xa0e4: jne     0xa0a5

```

```

0xa0e6: nop      word ptr cs:[rax + rax]
0xa0f0: mov      rax, qword ptr [rsp + 8]
0xa0f5: sub      rax, qword ptr fs:[0x28]
0xa0fe: jne      0xa199
0xa104: add      rsp, 0x18
0xa108: mov      rax, r12
0xa10b: pop      rbx
0xa10c: pop      rbp
0xa10d: pop      r12
0xa10f: pop      r13
0xa111: pop      r14
0xa113: pop      r15
0xa115: ret
0xa116: nop      word ptr cs:[rax + rax]
0xa120: lea      rsi, [rsp + 5]
0xa125: mov      edx, 0x12c
0xa12a: mov      rdi, r12
0xa12d: mov      byte ptr [rsp + 5], r15b
0xa132: mov      byte ptr [rsp + 6], 0
0xa137: call     0x47d0
0xa13c: jmp      0xa09b
0xa09b
0xa141: nop      dword ptr [rax]
0xa148: mov      edx, 0x12c
0xa14d: mov      rsi, r13
0xa150: mov      rdi, r12
0xa153: call     0x47d0
0xa158: jmp      0xa09b
0xa09b
0xa15d: nop      dword ptr [rax]
0xa160: mov      edx, 0x12c
0xa165: lea      rsi, [rip + 0x16a8f]
0xa16c: mov      rdi, r12
0xa16f: call     0x47d0
0xa174: jmp      0xa09b
0xa09b
0xa179: nop      dword ptr [rax]
0xa180: mov      edx, 0x12c
0xa185: lea      rsi, [rip + 0x16a79]
0xa18c: mov      rdi, r12
0xa18f: call     0x47d0
0xa194: jmp      0xa09b
0xa09b
0xa199: call     0x49d0
0xa19e: nop
0xa1a0: endbr64
0xa1a4: mov      eax, dword ptr [rip + 0x211ae]
0xa1aa: mov      r8, rdi

```

```

0xa1ad: mov     edi, dword ptr [rip + 0x207e5]
0xa1b3: push    r13
0xa1b5: push    r12
0xa1b7: xor     r12d, r12d
0xa1ba: lea     edx, [rax + rsi]
0xa1bd: push    rbx
0xa1be: mov     dword ptr [rip + 0x21194], edx
0xa1c4: cmp     edi, eax
0xa1c6: jle     0xa21c
0xa1c8: mov     ecx, dword ptr [rip + 0x21186]
0xa1ce: cmp     ecx, dword ptr [rip + 0x207c8]
0xa1d4: jge     0xa21c
0xa1d6: mov     r12d, edi
0xa1d9: sub     r12d, eax
0xa1dc: cmp     edx, edi
0xa1de: cmovle  r12d, esi
0xa1e2: imul    edi, ecx
0xa1e5: mov     rsi, r8
0xa1e8: movsxd  r13, r12d
0xa1eb: lea     ebx, [rdi + rax]
0xa1ee: mov     rdi, qword ptr [rip + 0x21143]
0xa1f5: mov     rdx, r13
0xa1f8: movsxd  rbx, ebx
0xa1fb: add     rdi, rbx
0xa1fe: call    0x4bb0
0xa203: add     rbx, qword ptr [rip + 0x21126]
0xa20a: movzx   esi, byte ptr [rip + 0x21140]
0xa211: mov     rdx, r13
0xa214: mov     rdi, rbx
0xa217: call    0x4a50
0xa21c: mov     eax, r12d
0xa21f: pop     rbx
0xa220: pop     r12
0xa222: pop     r13
0xa224: ret
0xa225: nop     word ptr cs:[rax + rax]
0xa230: endbr64
0xa234: push    r13
0xa236: push    r12
0xa238: xor     r12d, r12d
0xa23b: push    rbx
0xa23c: sub     rsp, 0x10
0xa240: mov     rax, qword ptr fs:[0x28]
0xa249: mov     qword ptr [rsp + 8], rax
0xa24e: xor     eax, eax
0xa250: mov     eax, dword ptr [rip + 0x21102]
0xa256: mov     byte ptr [rsp + 7], dil
0xa25b: mov     edi, dword ptr [rip + 0x20737]

```



```

0xa261: lea     edx, [rax + 1]
0xa264: mov     dword ptr [rip + 0x210ee], edx
0xa26a: cmp     eax, edi
0xa26c: jge     0xa2c2
0xa26e: mov     ecx, dword ptr [rip + 0x210e0]
0xa274: cmp     ecx, dword ptr [rip + 0x20722]
0xa27a: jge     0xa2c2
0xa27c: cmp     edx, edi
0xa27e: jle     0xa2e0
0xa280: mov     r12d, edi
0xa283: sub     r12d, eax
0xa286: movsxd  r13, r12d
0xa289: imul    edi, ecx
0xa28c: lea     rsi, [rsp + 7]
0xa291: mov     rdx, r13
0xa294: lea     ebx, [rdi + rax]
0xa297: mov     rdi, qword ptr [rip + 0x2109a]
0xa29e: movsxd  rbx, ebx
0xa2a1: add     rdi, rbx
0xa2a4: call    0x4bb0
0xa2a9: add     rbx, qword ptr [rip + 0x21080]
0xa2b0: movzx   esi, byte ptr [rip + 0x2109a]
0xa2b7: mov     rdx, r13
0xa2ba: mov     rdi, rbx
0xa2bd: call    0x4a50
0xa2c2: mov     rax, qword ptr [rsp + 8]
0xa2c7: sub     rax, qword ptr fs:[0x28]
0xa2d0: jne     0xa2ee
0xa2d2: add     rsp, 0x10
0xa2d6: mov     eax, r12d
0xa2d9: pop     rbx
0xa2da: pop     r12
0xa2dc: pop     r13
0xa2de: ret
0xa2df: nop
0xa2e0: mov     r13d, 1
0xa2e6: mov     r12d, 1
0xa2ec: jmp     0xa289
0xa289
0xa2ee: call    0x49d0
0xa2f3: nop     word ptr cs:[rax + rax]
0xa2fe: nop
0xa300: endbr64
0xa304: push    rbp
0xa305: lea     rdx, [rip - 0x89c]
0xa30c: push    rbx
0xa30d: sub     rsp, 8
0xa311: mov     esi, dword ptr [rip + 0x20685]

```

```

0xa317: mov     rdi, qword ptr [rip + 0x1faca]
0xa31e: call    0x4820
0xa323: mov     edx, dword ptr [rip + 0x20bdf]
0xa329: test    edx, edx
0xa32b: jle     0xa365
0xa32d: xor     ebp, ebp
0xa32f: lea     rbx, [rip + 0x20bea]
0xa336: jmp     0xa34c
0xa34c:
0xa338: nop     dword ptr [rax + rax]
0xa340: mov     edx, dword ptr [rip + 0x20bc2]
0xa346: add     ebp, eax
0xa348: cmp     edx, ebp
0xa34a: jle     0xa365
0xa34c: sub     edx, ebp
0xa34e: movsxd  rsi, ebp
0xa351: mov     edi, 1
0xa356: add     rsi, rbx
0xa359: movsxd  rdx, edx
0xa35c: call    0x18da0
0xa361: test    eax, eax
0xa363: jns     0xa340
0xa365: mov     edx, dword ptr [rip + 0x20631]
0xa36b: xor     esi, esi
0xa36d: imul    edx, dword ptr [rip + 0x20624]
0xa374: mov     dword ptr [rip + 0x20b8a], 0
0xa37e: mov     rdi, qword ptr [rip + 0x20fb3]
0xa385: movsxd  rdx, edx
0xa388: call    0x4a50
0xa38d: mov     edx, dword ptr [rip + 0x20609]
0xa393: imul    edx, dword ptr [rip + 0x205fe]
0xa39a: xor     esi, esi
0xa39c: mov     rdi, qword ptr [rip + 0x20f8d]
0xa3a3: movsxd  rdx, edx
0xa3a6: call    0x4a50
0xa3ab: mov     edx, dword ptr [rip + 0x205eb]
0xa3b1: imul    edx, dword ptr [rip + 0x205e0]
0xa3b8: xor     esi, esi
0xa3ba: mov     rdi, qword ptr [rip + 0x20f67]
0xa3c1: movsxd  rdx, edx
0xa3c4: call    0x4a50
0xa3c9: mov     edx, dword ptr [rip + 0x205cd]
0xa3cf: imul    edx, dword ptr [rip + 0x205c2]
0xa3d6: xor     esi, esi
0xa3d8: mov     rdi, qword ptr [rip + 0x20f41]
0xa3df: movsxd  rdx, edx
0xa3e2: call    0x4a50
0xa3e7: mov     dword ptr [rip + 0x20f67], 0

```

```

0xa3f1: mov     dword ptr [rip + 0x20f59], 0
0xa3fb: add     rsp, 8
0xa3ff: pop     rbx
0xa400: pop     rbp
0xa401: ret
0xa402: nop     word ptr cs:[rax + rax]
0xa40d: nop     dword ptr [rax]
0xa410: endbr64
0xa414: sub     rsp, 8
0xa418: mov     edx, dword ptr [rip + 0x2057e]
0xa41e: imul    edx, dword ptr [rip + 0x20573]
0xa425: mov     esi, 0x20
0xa42a: mov     rdi, qword ptr [rip + 0x20f07]
0xa431: movsxd  rdx, edx
0xa434: call    0x4a50
0xa439: mov     edx, dword ptr [rip + 0x2055d]
0xa43f: imul    edx, dword ptr [rip + 0x20552]
0xa446: movzx   esi, byte ptr [rip + 0x20f04]
0xa44d: mov     rdi, qword ptr [rip + 0x20edc]
0xa454: movsxd  rdx, edx
0xa457: call    0x4a50
0xa45c: mov     edx, dword ptr [rip + 0x2053a]
0xa462: imul    edx, dword ptr [rip + 0x2052f]
0xa469: xor     esi, esi
0xa46b: mov     rdi, qword ptr [rip + 0x20eb6]
0xa472: add     rsp, 8
0xa476: movsxd  rdx, edx
0xa479: jmp     0x4a50
0x4a50
0xa47e: nop
0xa480: endbr64
0xa484: mov     edx, dword ptr [rip + 0x20512]
0xa48a: imul    edx, dword ptr [rip + 0x20507]
0xa491: xor     esi, esi
0xa493: mov     rdi, qword ptr [rip + 0x20e8e]
0xa49a: movsxd  rdx, edx
0xa49d: jmp     0x4a50
0x4a50
0xa4a2: nop     word ptr cs:[rax + rax]
0xa4ad: nop     dword ptr [rax]
0xa4b0: endbr64
0xa4b4: mov     dword ptr [rip + 0x20e9a], edi
0xa4ba: mov     dword ptr [rip + 0x20e98], esi
0xa4c0: ret
0xa4c1: nop     word ptr cs:[rax + rax]
0xa4cc: nop     dword ptr [rax]
0xa4d0: endbr64
0xa4d4: mov     eax, dword ptr [rip + 0x20e7a]

```

```

0xa4da: mov     dword ptr [rdi], eax
0xa4dc: mov     eax, dword ptr [rip + 0x20e76]
0xa4e2: mov     dword ptr [rsi], eax
0xa4e4: ret
0xa4e5: nop     word ptr cs:[rax + rax]
0xa4f0: endbr64
0xa4f4: movzx   eax, byte ptr [rip + 0x20e56]
0xa4fb: and     edi, 7
0xa4fe: and     eax, 0xffffffff8
0xa501: or      eax, edi
0xa503: mov     byte ptr [rip + 0x20e48], al
0xa509: ret
0xa50a: nop     word ptr [rax + rax]
0xa510: endbr64
0xa514: movzx   eax, byte ptr [rip + 0x20e36]
0xa51b: shl     edi, 3
0xa51e: and     edi, 0x38
0xa521: and     eax, 0xffffffffc7
0xa524: or      eax, edi
0xa526: mov     byte ptr [rip + 0x20e25], al
0xa52c: ret
0xa52d: nop     dword ptr [rax]
0xa530: endbr64
0xa534: movzx   eax, byte ptr [rip + 0x20e16]
0xa53b: shl     edi, 6
0xa53e: and     edi, 0x40
0xa541: and     eax, 0xffffffffbf
0xa544: or      eax, edi
0xa546: mov     byte ptr [rip + 0x20e05], al
0xa54c: ret
0xa54d: nop     dword ptr [rax]
0xa550: endbr64
0xa554: movzx   eax, byte ptr [rip + 0x20df6]
0xa55b: shl     edi, 7
0xa55e: and     eax, 0x7f
0xa561: or      eax, edi
0xa563: mov     byte ptr [rip + 0x20de8], al
0xa569: ret
0xa56a: nop     word ptr [rax + rax]
0xa570: endbr64
0xa574: shl     edi, 6
0xa577: and     esi, 7
0xa57a: shl     edx, 3
0xa57d: mov     eax, edi
0xa57f: movzx   edi, byte ptr [rip + 0x20dcb]
0xa586: and     edx, 0x38
0xa589: and     eax, 0x40
0xa58c: and     edi, 0xffffffffb8

```

```

0xa58f: or      edi, eax
0xa591: or      edi, esi
0xa593: and     edi, 0xffffffffc7
0xa596: or      edi, edx
0xa598: mov     byte ptr [rip + 0x20db2], dil
0xa59f: ret
0xa5a0: endbr64
0xa5a4: push    rbp
0xa5a5: push    rbx
0xa5a6: sub     rsp, 8
0xa5aa: movsxd  rax, dword ptr [rip + 0x20957]
0xa5b1: cmp     eax, 0x400
0xa5b6: je      0xa610
0xa5b8: lea     edx, [rax + 1]
0xa5bb: lea     rbx, [rip + 0x2095e]
0xa5c2: mov     dword ptr [rip + 0x20940], edx
0xa5c8: mov     byte ptr [rbx + rax], 7
0xa5cc: test    edx, edx
0xa5ce: jle     0xa5fd
0xa5d0: xor     ebp, ebp
0xa5d2: jmp     0xa5e4
0xa5e4:
0xa5d4: nop     dword ptr [rax]
0xa5d8: mov     edx, dword ptr [rip + 0x2092a]
0xa5de: add     ebp, eax
0xa5e0: cmp     edx, ebp
0xa5e2: jle     0xa5fd
0xa5e4: sub     edx, ebp
0xa5e6: movsxd  rsi, ebp
0xa5e9: mov     edi, 1
0xa5ee: add     rsi, rbx
0xa5f1: movsxd  rdx, edx
0xa5f4: call    0x18da0
0xa5f9: test    eax, eax
0xa5fb: jns     0xa5d8
0xa5fd: mov     dword ptr [rip + 0x20901], 0
0xa607: add     rsp, 8
0xa60b: pop     rbx
0xa60c: pop     rbp
0xa60d: ret
0xa60e: nop
0xa610: xor     ebp, ebp
0xa612: lea     rbx, [rip + 0x20907]
0xa619: jmp     0xa62c
0xa62c:
0xa61b: nop     dword ptr [rax + rax]
0xa620: add     ebp, eax
0xa622: mov     eax, dword ptr [rip + 0x208e0]

```

```

0xa628: cmp     eax, ebp
0xa62a: jle     0xa645
0xa62c: sub     eax, ebp
0xa62e: movsxd  rsi, ebp
0xa631: mov     edi, 1
0xa636: add     rsi, rbx
0xa639: movsxd  rdx, eax
0xa63c: call    0x18da0
0xa641: test    eax, eax
0xa643: jns     0xa620
0xa645: mov     dword ptr [rip + 0x208b9], 1
0xa64f: mov     edx, 1
0xa654: mov     byte ptr [rip + 0x208c5], 7
0xa65b: jmp     0xa5d0
0xa5d0
0xa660: endbr64
0xa664: test    edi, edi
0xa666: je      0xa688
0xa668: mov     rdi, qword ptr [rip + 0x1f759]
0xa66f: test    rdi, rdi
0xa672: je      0xa694
0xa674: lea     rdx, [rip - 0xc0b]
0xa67b: mov     esi, 1
0xa680: jmp     0x4820
0x4820
0xa685: nop     dword ptr [rax]
0xa688: mov     rdi, qword ptr [rip + 0x1f719]
0xa68f: test    rdi, rdi
0xa692: jne     0xa674
0xa694: ret
0xa695: nop     word ptr cs:[rax + rax]
0xa6a0: endbr64
0xa6a4: movzx   eax, byte ptr [rip + 0x20ca6]
0xa6ab: mov     byte ptr [rdi], al
0xa6ad: ret
0xa6ae: nop
0xa6b0: endbr64
0xa6b4: movzx   eax, byte ptr [rdi]
0xa6b7: mov     byte ptr [rip + 0x20c94], al
0xa6bd: ret
0xa6be: nop
0xa6c0: endbr64
0xa6c4: push    rbp
0xa6c5: push    rbx
0xa6c6: sub     rsp, 8
0xa6ca: cmp     dword ptr [rip + 0x1f58f], 1
0xa6d1: je      0xa720
0xa6d3: mov     rdi, qword ptr [rip + 0x1f66e]

```

```

0xa6da: test    rdi, rdi
0xa6dd: je      0xa6f0
0xa6df: lea     rdx, [rip - 0xc76]
0xa6e6: mov     esi, 1
0xa6eb: call    0x4820
0xa6f0: mov     dword ptr [rip + 0x20c52], 0
0xa6fa: mov     dword ptr [rip + 0x20c44], 0
0xa704: mov     dword ptr [rip + 0x20c36], 0
0xa70e: mov     dword ptr [rip + 0x20c28], 0
0xa718: add     rsp, 8
0xa71c: pop     rbx
0xa71d: pop     rbp
0xa71e: ret
0xa71f: nop
0xa720: mov     edx, dword ptr [rip + 0x207e2]
0xa726: cmp     edx, 0x3fb
0xa72c: jg      0xa750
0xa72e: movsxd  rax, edx
0xa731: lea     rcx, [rip + 0x207e8]
0xa738: add     edx, 4
0xa73b: add     rax, rcx
0xa73e: mov     ecx, dword ptr [rip + 0x1f5db]
0xa744: mov     dword ptr [rip + 0x207be], edx
0xa74a: mov     dword ptr [rax], ecx
0xa74c: jmp     0xa6d3
0xa6d3:
0xa74e: nop
0xa750: xor     ebx, ebx
0xa752: lea     rbp, [rip + 0x207c7]
0xa759: jmp     0xa76c
0xa76c:
0xa75b: nop     dword ptr [rax + rax]
0xa760: mov     edx, dword ptr [rip + 0x207a2]
0xa766: add     ebx, eax
0xa768: cmp     edx, ebx
0xa76a: jle     0xa785
0xa76c: sub     edx, ebx
0xa76e: movsxd  rsi, ebx
0xa771: mov     edi, 1
0xa776: add     rsi, rbp
0xa779: movsxd  rdx, edx
0xa77c: call    0x18da0
0xa781: test    eax, eax
0xa783: jns     0xa760
0xa785: mov     edx, 4
0xa78a: lea     rax, [rip + 0x2078f]
0xa791: jmp     0xa73e
0xa73e:

```

```

0xa793: nop      word ptr cs:[rax + rax]
0xa79e: nop
0xa7a0: push     r12
0xa7a2: mov      r12d, edi
0xa7a5: push     rbp
0xa7a6: mov      ebp, esi
0xa7a8: sub      rsp, 8
0xa7ac: mov      eax, dword ptr [rip + 0x1f71e]
0xa7b2: test     eax, eax
0xa7b4: je       0xa800
0xa7b6: mov      eax, ebp
0xa7b8: mov      esi, dword ptr [rip + 0x201da]
0xa7be: mov      edx, dword ptr [rip + 0x201d8]
0xa7c4: or       eax, r12d
0xa7c7: mov      rdi, qword ptr [rip + 0x1f63a]
0xa7ce: js       0xa7d4
0xa7d0: cmp      ebp, esi
0xa7d2: jl       0xa810
0xa7d4: sub      edx, 1
0xa7d7: sub      esi, 1
0xa7da: call     0x4960
0xa7df: add      rsp, 8
0xa7e3: mov      esi, 1
0xa7e8: lea      rdx, [rip - 0xd7f]
0xa7ef: pop      rbp
0xa7f0: mov      rdi, rax
0xa7f3: pop      r12
0xa7f5: jmp      0x4820
0x4820
0xa7fa: nop      word ptr [rax + rax]
0xa800: xor      eax, eax
0xa802: call     0xa6c0
0xa807: jmp      0xa7b6
0xa7b6
0xa809: nop      dword ptr [rax]
0xa810: cmp      r12d, edx
0xa813: jge      0xa7d4
0xa815: mov      edx, r12d
0xa818: mov      esi, ebp
0xa81a: jmp      0xa7da
0xa7da
0xa81c: nop      dword ptr [rax]
0xa820: endbr64
0xa824: push     r15
0xa826: push     r14
0xa828: push     r13
0xa82a: push     r12
0xa82c: push     rbp

```



```

0xa82d: push    rbx
0xa82e: sub     rsp, 0x38
0xa832: mov     ebx, dword ptr [rip + 0x20160]
0xa838: imul    ebx, dword ptr [rip + 0x2015d]
0xa83f: mov     rax, qword ptr fs:[0x28]
0xa848: mov     qword ptr [rsp + 0x28], rax
0xa84d: xor     eax, eax
0xa84f: mov     rdi, qword ptr [rip + 0x1f552]
0xa856: cmp     dword ptr [rip + 0x20aff], 1
0xa85d: sbb     ebx, 0
0xa860: test    rdi, rdi
0xa863: je      0xa876
0xa865: lea     rdx, [rip - 0xdfc]
0xa86c: mov     esi, 1
0xa871: call    0x4820
0xa876: test    ebx, ebx
0xa878: jle     0xab70
0xa87e: movsxd  rax, ebx
0xa881: mov     dword ptr [rsp + 4], 0xffffffff
0xa889: xor     ebp, ebp
0xa88b: mov     r13d, 0xffffffff
0xa891: mov     qword ptr [rsp + 8], rax
0xa896: lea     r12, [rip + 0x20683]
0xa89d: jmp     0xa9ce
0xa9ce:
0xa8a2: nop     word ptr [rax + rax]
0xa8a8: cdq
0xa8a9: idiv    dword ptr [rip + 0x200e9]
0xa8af: mov     r14d, edx
0xa8b2: cmp     edx, r13d
0xa8b5: jne     0xaa20
0xa8bb: cmp     eax, dword ptr [rsp + 4]
0xa8bf: jne     0xaa20
0xa8c5: mov     rax, qword ptr [rip + 0x20a64]
0xa8cc: movzx   ebx, byte ptr [rax + rbp]
0xa8d0: mov     edi, ebx
0xa8d2: mov     r14d, ebx
0xa8d5: sar     edi, 6
0xa8d8: and     r14d, 7
0xa8dc: and     edi, 1
0xa8df: call    0x9870
0xa8e4: mov     eax, ebx
0xa8e6: and     eax, 7
0xa8e9: cmp     dword ptr [rip + 0x20a5c], 1
0xa8f0: je      0xaa40
0xa8f6: cmp     dword ptr [rip + 0x1f363], 1
0xa8fd: je      0xaa5f
0xa903: xor     edi, edi

```

```

0xa905: cmp     eax, 7
0xa908: setne   dil
0xa90c: call    0x9790
0xa911: movzx   eax, byte ptr [rip + 0x20a38]
0xa918: mov     dword ptr [rip + 0x20a2a], 1
0xa922: and     eax, 0xffffffff8
0xa925: or      eax, r14d
0xa928: mov     byte ptr [rip + 0x20a22], al
0xa92e: sar     ebx, 3
0xa931: and     ebx, 7
0xa934: cmp     dword ptr [rip + 0x20a0d], 1
0xa93b: je      0xaab0
0xa941: cmp     dword ptr [rip + 0x1f318], 1
0xa948: je      0xaad8
0xa94e: xor     edi, edi
0xa950: test    ebx, ebx
0xa952: setne   dil
0xa956: call    0x9790
0xa95b: mov     edx, dword ptr [rip + 0x205a7]
0xa961: movzx   eax, byte ptr [rip + 0x209e8]
0xa968: shl     ebx, 3
0xa96b: mov     dword ptr [rip + 0x209d3], 1
0xa975: and     eax, 0xffffffffc7
0xa978: or      ebx, eax
0xa97a: mov     byte ptr [rip + 0x209d0], bl
0xa980: mov     rax, qword ptr [rip + 0x209b1]
0xa987: movzx   ebx, byte ptr [rax + rbp]
0xa98b: cmp     edx, 0x400
0xa991: je      0xab30
0xa997: lea     eax, [rdx + 1]
0xa99a: movsxd  rdx, edx
0xa99d: add     r13d, 1
0xa9a1: cmp     dword ptr [rip + 0x1fff0], r13d
0xa9a8: mov     dword ptr [rip + 0x2055a], eax
0xa9ae: mov     byte ptr [r12 + rdx], bl
0xa9b2: jne     0xa9bf
0xa9b4: mov     dword ptr [rsp + 4], 0xffffffff
0xa9bc: xor     r13d, r13d
0xa9bf: add     rbp, 1
0xa9c3: cmp     rbp, qword ptr [rsp + 8]
0xa9c8: je      0xab70
0xa9ce: mov     rdx, qword ptr [rip + 0x20953]
0xa9d5: mov     rcx, qword ptr [rip + 0x2095c]
0xa9dc: mov     eax, ebp
0xa9de: movzx   ebx, byte ptr [rdx + rbp]
0xa9e2: cmp     byte ptr [rcx + rbp], bl
0xa9e5: jne     0xa8a8
0xa9eb: mov     rdx, qword ptr [rip + 0x2092e]

```

```

0xa9f2: mov     rcx, qword ptr [rip + 0x20937]
0xa9f9: movzx   edi, byte ptr [rdx + rbp]
0xa9fd: cmp     byte ptr [rcx + rbp], dil
0xaa01: je      0xa9bf
0xaa03: cdq
0xaa04: idiv    dword ptr [rip + 0x1ff8e]
0xaa0a: mov     r14d, edx
0xaa0d: cmp     edx, r13d
0xaa10: je      0xa8bb
0xaa16: nop     word ptr cs:[rax + rax]
0xaa20: mov     esi, r14d
0xaa23: mov     edi, eax
0xaa25: mov     dword ptr [rsp + 4], eax
0xaa29: mov     r13d, r14d
0xaa2c: call    0xa7a0
0xaa31: jmp     0xa8c5
0xa8c5
0xaa36: nop     word ptr cs:[rax + rax]
0xaa40: movzx   edx, byte ptr [rip + 0x20909]
0xaa47: and     edx, 7
0xaa4a: cmp     eax, edx
0xaa4c: je      0xa92e
0xaa52: cmp     dword ptr [rip + 0x1f207], 1
0xaa59: jne     0xa903
0xaa5f: mov     eax, dword ptr [rip + 0x164b4]
0xaa65: mov     edx, dword ptr [rip + 0x2049d]
0xaa6b: mov     dword ptr [rsp + 0x10], eax
0xaa6f: movzx   eax, byte ptr [rip + 0x164a7]
0xaa76: mov     byte ptr [rsp + 0x14], al
0xaa7a: lea     eax, [r14 + 0x30]
0xaa7e: mov     byte ptr [rsp + 0x13], al
0xaa82: cmp     edx, 0x3fa
0xaa88: jg      0xac18
0xaa8e: movsxd  rax, edx
0xaa91: add     edx, 5
0xaa94: add     rax, r12
0xaa97: mov     ecx, dword ptr [rsp + 0x10]
0xaa9b: mov     dword ptr [rip + 0x20467], edx
0xaaa1: mov     dword ptr [rax], ecx
0xaaa3: movzx   ecx, byte ptr [rsp + 0x14]
0xaaa8: mov     byte ptr [rax + 4], cl
0xaaab: jmp     0xa911
0xa911
0xaab0: movzx   eax, byte ptr [rip + 0x20899]
0xaab7: mov     edx, dword ptr [rip + 0x2044b]
0xaabd: sar     eax, 3
0xaac0: and     eax, 7
0xaac3: cmp     ebx, eax

```

```

0xaac5: je      0xa980
0xaacb: cmp     dword ptr [rip + 0x1f18e], 1
0xaad2: jne     0xa94e
0xaad8: mov     eax, dword ptr [rip + 0x16436]
0xaade: mov     edx, dword ptr [rip + 0x20424]
0xaae4: mov     dword ptr [rsp + 0x10], eax
0xaae8: movzx   eax, byte ptr [rip + 0x16429]
0xaaef: mov     byte ptr [rsp + 0x14], al
0xaaf3: lea     eax, [rbx + 0x30]
0xaaf6: mov     byte ptr [rsp + 0x13], al
0xaafa: cmp     edx, 0x3fa
0xab00: jg      0xac58
0xab06: movsxd  rax, edx
0xab09: add     edx, 5
0xab0c: add     rax, r12
0xab0f: mov     ecx, dword ptr [rsp + 0x10]
0xab13: mov     dword ptr [rip + 0x203ef], edx
0xab19: mov     dword ptr [rax], ecx
0xab1b: movzx   ecx, byte ptr [rsp + 0x14]
0xab20: mov     byte ptr [rax + 4], cl
0xab23: jmp     0xa961
0xa961
0xab28: nop     dword ptr [rax + rax]
0xab30: xor     r14d, r14d
0xab33: jmp     0xab46
0xab46
0xab35: nop     dword ptr [rax]
0xab38: mov     edx, dword ptr [rip + 0x203ca]
0xab3e: add     r14d, eax
0xab41: cmp     edx, r14d
0xab44: jle     0xab60
0xab46: sub     edx, r14d
0xab49: movsxd  rsi, r14d
0xab4c: mov     edi, 1
0xab51: add     rsi, r12
0xab54: movsxd  rdx, edx
0xab57: call    0x18da0
0xab5c: test    eax, eax
0xab5e: jns     0xab38
0xab60: mov     eax, 1
0xab65: xor     edx, edx
0xab67: jmp     0xa99a
0xa99a
0xab6c: nop     dword ptr [rax]
0xab70: mov     edi, dword ptr [rip + 0x207de]
0xab76: mov     esi, dword ptr [rip + 0x207dc]
0xab7c: call    0xa7a0
0xab81: mov     rdi, qword ptr [rip + 0x1f240]

```

```

0xab88: test    rdi, rdi
0xab8b: je       0xab9e
0xab8d: lea     rdx, [rip - 0x1124]
0xab94: mov     esi, 1
0xab99: call    0x4820
0xab9e: mov     edx, dword ptr [rip + 0x20364]
0xaba4: test    edx, edx
0xaba6: jne     0xac98
0xabac: mov     edx, dword ptr [rip + 0x1fde6]
0xabb2: imul    edx, dword ptr [rip + 0x1fde3]
0xabb9: mov     rsi, qword ptr [rip + 0x20778]
0xabc0: mov     rdi, qword ptr [rip + 0x20761]
0xabc7: movsxd  rdx, edx
0xabca: call    0x4bb0
0xabcf: mov     edx, dword ptr [rip + 0x1fdc3]
0xabd5: imul    edx, dword ptr [rip + 0x1fdc0]
0xabdc: mov     rdi, qword ptr [rip + 0x2073d]
0xabe3: mov     rsi, qword ptr [rip + 0x20746]
0xabea: movsxd  rdx, edx
0xabed: mov     rax, qword ptr [rsp + 0x28]
0xabf2: sub     rax, qword ptr fs:[0x28]
0xabfb: jne     0xace4
0xac01: add     rsp, 0x38
0xac05: pop     rbx
0xac06: pop     rbp
0xac07: pop     r12
0xac09: pop     r13
0xac0b: pop     r14
0xac0d: pop     r15
0xac0f: jmp     0x4bb0
0x4bb0:
0xac14: nop     dword ptr [rax]
0xac18: xor     r15d, r15d
0xac1b: jmp     0xac2e
0xac2e:
0xac1d: nop     dword ptr [rax]
0xac20: mov     edx, dword ptr [rip + 0x202e2]
0xac26: add     r15d, eax
0xac29: cmp     edx, r15d
0xac2c: jle     0xac48
0xac2e: sub     edx, r15d
0xac31: movsxd  rsi, r15d
0xac34: mov     edi, 1
0xac39: add     rsi, r12
0xac3c: movsxd  rdx, edx
0xac3f: call    0x18da0
0xac44: test    eax, eax
0xac46: jns     0xac20

```

```

0xac48: mov     edx, 5
0xac4d: mov     rax, r12
0xac50: jmp     0xaa97
0xaa97:
0xac55: nop     dword ptr [rax]
0xac58: xor     r14d, r14d
0xac5b: jmp     0xac6e
0xac6e:
0xac5d: nop     dword ptr [rax]
0xac60: mov     edx, dword ptr [rip + 0x202a2]
0xac66: add     r14d, eax
0xac69: cmp     edx, r14d
0xac6c: jle     0xac88
0xac6e: sub     edx, r14d
0xac71: movsxd  rsi, r14d
0xac74: mov     edi, 1
0xac79: add     rsi, r12
0xac7c: movsxd  rdx, edx
0xac7f: call    0x18da0
0xac84: test    eax, eax
0xac86: jns     0xac60
0xac88: mov     edx, 5
0xac8d: mov     rax, r12
0xac90: jmp     0xab0f
0xab0f:
0xac95: nop     dword ptr [rax]
0xac98: mov     ebx, 0
0xac9d: lea     rbp, [rip + 0x2027c]
0xaca4: jg      0xacbc
0xaca6: jmp     0xacd5
0xacd5:
0xaca8: nop     dword ptr [rax + rax]
0xacb0: mov     edx, dword ptr [rip + 0x20252]
0xacb6: add     ebx, eax
0xacb8: cmp     edx, ebx
0xacba: jle     0xacd5
0xacbc: sub     edx, ebx
0xacbe: movsxd  rsi, ebx
0xacc1: mov     edi, 1
0xacc6: add     rsi, rbp
0xacc9: movsxd  rdx, edx
0xacc: call    0x18da0
0xacd1: test    eax, eax
0xacd3: jns     0xacb0
0xacd5: mov     dword ptr [rip + 0x20229], 0
0xacdf: jmp     0xabac
0xabac:
0xace4: call    0x49d0

```

```

0xace9: nop      dword ptr [rax]
0xacf0: endbr64
0xacf4: push     r12
0xacf6: xor      eax, eax
0xacf8: mov      r12, rdi
0xacfb: push     rbx
0xacfc: mov      ebx, esi
0xacfe: sub      rsp, 8
0xad02: call     0xa820
0xad07: mov      rdx, qword ptr [rip + 0x1fc1a]
0xad0e: test     rdx, rdx
0xad11: je       0xad17
0xad13: xor      eax, eax
0xad15: call     rdx
0xad17: movsxd   rdx, ebx
0xad1a: mov      rsi, r12
0xad1d: xor      edi, edi
0xad1f: call     0x18d30
0xad24: mov      rdx, qword ptr [rip + 0x1fbf5]
0xad2b: mov      r12d, eax
0xad2e: test     rdx, rdx
0xad31: je       0xad37
0xad33: xor      eax, eax
0xad35: call     rdx
0xad37: add      rsp, 8
0xad3b: mov      eax, r12d
0xad3e: pop      rbx
0xad3f: pop      r12
0xad41: ret
0xad42: nop      word ptr cs:[rax + rax]
0xad4d: nop      dword ptr [rax]
0xad50: push     r12
0xad52: mov      edi, 1
0xad57: push     rbp
0xad58: lea      rbp, [rip + 0x1fda1]
0xad5f: push     rbx
0xad60: sub      rsp, 0x50
0xad64: mov      rax, qword ptr fs:[0x28]
0xad6d: mov      qword ptr [rsp + 0x48], rax
0xad72: xor      eax, eax
0xad74: mov      r12, rsp
0xad77: mov      rsi, r12
0xad7a: call     0x4d60
0xad7f: mov      edx, 0xff
0xad84: mov      esi, 1
0xad89: mov      edi, 1
0xad8e: mov      word ptr [rsp + 0x11], dx
0xad93: mov      rdx, r12

```

```

0xad96: call    0x4d70
0xad9b: mov     edi, 1
0xada0: mov     dword ptr [rip + 0x1ef7a], 0xffffffff
0xadaa: call    0x18b70
0xadaf: mov     dword ptr [rip + 0x20147], 0
0xadb9: nop     dword ptr [rax]
0xadc0: xor     eax, eax
0xadc2: call    0xa820
0xadc7: mov     rdx, qword ptr [rip + 0x1fb5a]
0xadce: test    rdx, rdx
0xadd1: je      0xadd7
0xadd3: xor     eax, eax
0xadd5: call    rdx
0xadd7: mov     edx, 0x400
0xaddc: mov     rsi, rbp
0xaddf: xor     edi, edi
0xade1: call    0x18d30
0xade6: mov     rdx, qword ptr [rip + 0x1fb33]
0xaded: mov     ebx, eax
0xade7: test    rdx, rdx
0xadf2: je      0xadf8
0xadf4: xor     eax, eax
0xadf6: call    rdx
0xadf8: mov     dword ptr [rip + 0x20106], ebx
0xadfe: test    ebx, ebx
0xae00: js      0xadc0
0xae02: xor     edi, edi
0xae04: call    0x18b70
0xae09: mov     rsi, r12
0xae0c: mov     edi, 1
0xae11: call    0x4d60
0xae16: mov     eax, 7
0xae1b: mov     rdx, r12
0xae1e: mov     esi, 1
0xae23: mov     edi, 1
0xae28: mov     word ptr [rsp + 0x11], ax
0xae2d: call    0x4d70
0xae32: mov     eax, dword ptr [rip + 0x200cc]
0xae38: mov     dword ptr [rip + 0x1eee2], 7
0xae42: test    eax, eax
0xae44: je      0xae9f
0xae46: movsxd  rsi, dword ptr [rip + 0x200b3]
0xae4d: movsx   r8d, byte ptr [rbp + rsi]
0xae53: mov     rdx, rsi
0xae56: mov     ecx, r8d
0xae59: test    r8b, r8b
0xae5c: js      0xae90
0xae5e: sub     eax, 1

```



```

0xae61: add     edx, 1
0xae64: mov     dword ptr [rip + 0x2009a], eax
0xae6a: mov     dword ptr [rip + 0x20090], edx
0xae70: mov     rax, qword ptr [rsp + 0x48]
0xae75: sub     rax, qword ptr fs:[0x28]
0xae7e: jne     0xaea7
0xae80: add     rsp, 0x50
0xae84: mov     eax, r8d
0xae87: pop     rbx
0xae88: pop     rbp
0xae89: pop     r12
0xae8b: ret
0xae8c: nop     dword ptr [rax]
0xae90: and     ecx, 0x7f
0xae93: mov     r8d, 0x1b
0xae99: mov     byte ptr [rbp + rsi], cl
0xae9d: jmp     0xae70
0xae70:
0xae9f: mov     r8d, 0xffffffff
0xaea5: jmp     0xae70
0xae70:
0xaea7: call    0x49d0
0xaeac: nop     dword ptr [rax]
0xaeb0: endbr64
0xaeb4: xor     eax, eax
0xaeb6: sub     rsp, 8
0xaeba: call    0x18ab0
0xaebf: mov     eax, dword ptr [rip + 0x2003f]
0xaec5: test    eax, eax
0xaec7: jne     0xaed8
0xaec9: xor     eax, eax
0xaecb: add     rsp, 8
0xaecf: jmp     0xad50
0xad50:
0xaed4: nop     dword ptr [rax]
0xaed8: movsxd  rsi, dword ptr [rip + 0x20021]
0xaedf: lea     rdi, [rip + 0x1fc1a]
0xae6: movsx  r8d, byte ptr [rdi + rsi]
0xaeeb: mov     rdx, rsi
0xaeee: mov     ecx, r8d
0xae71: test    r8b, r8b
0xae74: js      0xaf10
0xae76: sub     eax, 1
0xae79: add     edx, 1
0xae7c: mov     dword ptr [rip + 0x20002], eax
0xaf02: mov     dword ptr [rip + 0x1fff8], edx
0xaf08: mov     eax, r8d
0xaf0b: add     rsp, 8

```

```

0xaf0f: ret
0xaf10: and     ecx, 0x7f
0xaf13: mov     r8d, 0x1b
0xaf19: mov     byte ptr [rdi + rsi], cl
0xaf1c: jmp     0xaf08
0xaf08:
0xaf1e: nop
0xaf20: endbr64
0xaf24: push    r13
0xaf26: mov     r13, rsi
0xaf29: push    r12
0xaf2b: mov     r12, rdx
0xaf2e: push    rbp
0xaf2f: mov     rbp, rdi
0xaf32: mov     edi, 0x18
0xaf37: push    rbx
0xaf38: sub     rsp, 8
0xaf3c: call    0x1d670
0xaf41: mov     rdi, r13
0xaf44: mov     rbx, rax
0xaf47: call    0x1d7a0
0xaf4c: mov     qword ptr [rbx + 0x10], r12
0xaf50: mov     qword ptr [rbx], rax
0xaf53: mov     rax, qword ptr [rbp]
0xaf57: mov     qword ptr [rbx + 8], rax
0xaf5b: mov     qword ptr [rbp], rbx
0xaf5f: add     rsp, 8
0xaf63: pop     rbx
0xaf64: pop     rbp
0xaf65: pop     r12
0xaf67: pop     r13
0xaf69: ret
0xaf6a: nop     word ptr [rax + rax]
0xaf70: endbr64
0xaf74: cmp     byte ptr [rdi], 0
0xaf77: je      0xb030
0xaf7d: push    r13
0xaf7f: push    r12
0xaf81: mov     r12, rsi
0xaf84: push    rbp
0xaf85: mov     rbp, rdi
0xaf88: push    rbx
0xaf89: sub     rsp, 8
0xaf8d: mov     r13, qword ptr [rip + 0x1fa0c]
0xaf94: test    r13, r13
0xaf97: je      0xb010
0xaf99: mov     rbx, qword ptr [r13]
0xaf9d: mov     rsi, qword ptr [rbx]

```

```

0xafa0: call    0x4b30
0xafa5: test    eax, eax
0xafa7: jle     0xb010
0xafa9: nop     dword ptr [rax]
0xafb0: lea     r13, [rbx + 8]
0xafb4: mov     rbx, qword ptr [rbx + 8]
0xafb8: mov     qword ptr [rip + 0x1f9e1], r13
0xafbf: test    rbx, rbx
0xafc2: je      0xafd3
0xafc4: mov     rsi, qword ptr [rbx]
0xafc7: mov     rdi, rbp
0xafca: call    0x4b30
0xafcf: test    eax, eax
0xafd1: jg      0xafb0
0xafd3: mov     edi, 0x18
0xafd8: call    0x1d670
0xafdd: mov     rdi, rbp
0xafe0: mov     rbx, rax
0xafe3: call    0x1d7a0
0xafe8: mov     qword ptr [rbx + 0x10], r12
0xafec: mov     qword ptr [rbx], rax
0xafef: mov     rax, qword ptr [r13]
0xaff3: mov     qword ptr [rbx + 8], rax
0xaff7: mov     qword ptr [r13], rbx
0xaffb: add     rsp, 8
0xafff: pop     rbx
0xb000: pop     rbp
0xb001: pop     r12
0xb003: pop     r13
0xb005: ret
0xb006: nop     word ptr cs:[rax + rax]
0xb010: mov     rbx, qword ptr [rip + 0x1f969]
0xb017: lea     r13, [rip + 0x1f962]
0xb01e: mov     qword ptr [rip + 0x1f97b], r13
0xb025: test    rbx, rbx
0xb028: jne     0xafc4
0xb02a: jmp     0xafd3
0xafd3:
0xb02c: nop     dword ptr [rax]
0xb030: ret
0xb031: nop     word ptr cs:[rax + rax]
0xb03c: nop     dword ptr [rax]
0xb040: endbr64
0xb044: mov     rax, qword ptr [rip + 0x1f935]
0xb04b: mov     qword ptr [rip + 0x1f926], rax
0xb052: ret
0xb053: nop     word ptr cs:[rax + rax]
0xb05e: nop

```

```

0xb060: endbr64
0xb064: cmp     qword ptr [rip + 0x1f90c], 0
0xb06c: je      0xb073
0xb06e: jmp     0x99e0
0x99e0
0xb073: xor     eax, eax
0xb075: ret
0xb076: nop     word ptr cs:[rax + rax]
0xb080: endbr64
0xb084: mov     eax, dword ptr [rip + 0x1f912]
0xb08a: push    r12
0xb08c: mov     r12, rdi
0xb08f: push    rbp
0xb090: push    rbx
0xb091: sub     eax, 1
0xb094: movzx   ebx, byte ptr [rip + 0x202b6]
0xb09b: mov     dword ptr [rip + 0x202b3], eax
0xb0a1: mov     dword ptr [rip + 0x202ad], 0
0xb0ab: mov     eax, ebx
0xb0ad: and     eax, 0xffffffffc0
0xb0b0: or      eax, 0x38
0xb0b3: mov     byte ptr [rip + 0x20298], al
0xb0b9: mov     eax, dword ptr [rip + 0x1f8d9]
0xb0bf: lea     edi, [rax + 1]
0xb0c2: movsxd  rdi, edi
0xb0c5: call    0x1d670
0xb0ca: movsxd  rdx, dword ptr [rip + 0x1f8c7]
0xb0d1: mov     esi, 0x20
0xb0d6: mov     rdi, rax
0xb0d9: mov     rbp, rax
0xb0dc: call    0x4a50
0xb0e1: movsxd  rax, dword ptr [rip + 0x1f8b0]
0xb0e8: mov     rdi, rbp
0xb0eb: mov     byte ptr [rbp + rax], 0
0xb0f0: mov     esi, dword ptr [rip + 0x1f8a2]
0xb0f6: call    0xa1a0
0xb0fb: mov     rdi, rbp
0xb0fe: lea     rbp, [rip + 0x1f8bb]
0xb105: call    0x18d10
0xb10a: mov     eax, dword ptr [rip + 0x1f88c]
0xb110: mov     rdi, r12
0xb113: mov     dword ptr [rip + 0x2023b], 0
0xb11d: sub     eax, 1
0xb120: mov     dword ptr [rip + 0x2022e], eax
0xb126: call    0x9fe0
0xb12b: mov     esi, 0x13
0xb130: lea     rdi, [rip + 0x15ad5]
0xb137: call    0xa1a0

```

```

0xb13c: mov     rdi, rbp
0xb13f: call    0x49b0
0xb144: mov     rdi, rbp
0xb147: mov     esi, eax
0xb149: call    0xa1a0
0xb14e: mov     esi, 1
0xb153: lea     rdi, [rip + 0x155eb]
0xb15a: call    0xa1a0
0xb15f: xor     eax, eax
0xb161: call    0xa820
0xb166: mov     byte ptr [rip + 0x201e5], bl
0xb16c: pop     rbx
0xb16d: pop     rbp
0xb16e: pop     r12
0xb170: ret
0xb171: nop     word ptr cs:[rax + rax]
0xb17c: nop     dword ptr [rax]
0xb180: endbr64
0xb184: push    r15
0xb186: push    r14
0xb188: mov     r14, rdi
0xb18b: push    r13
0xb18d: push    r12
0xb18f: push    rbp
0xb190: lea     rbp, [rip + 0x1f969]
0xb197: push    rbx
0xb198: sub     rsp, 0x28
0xb19c: jmp     0xb1ae
0xb1ae:
0xb19e: nop
0xb1a0: xor     eax, eax
0xb1a2: call    0xad50
0xb1a7: mov     ebx, eax
0xb1a9: cmp     ebx, -1
0xb1ac: jne     0xb1ef
0xb1ae: xor     eax, eax
0xb1b0: call    0x18ab0
0xb1b5: mov     eax, dword ptr [rip + 0x1fd49]
0xb1bb: test    eax, eax
0xb1bd: je      0xb1a0
0xb1bf: movsxd  rsi, dword ptr [rip + 0x1fd3a]
0xb1c6: movsx   ebx, byte ptr [rbp + rsi]
0xb1cb: mov     rdx, rsi
0xb1ce: mov     ecx, ebx
0xb1d0: test    bl, bl
0xb1d2: js      0xb4a0
0xb1d8: sub     eax, 1
0xb1db: add     edx, 1

```

```

0xb1de: mov     dword ptr [rip + 0x1fd20], eax
0xb1e4: mov     dword ptr [rip + 0x1fd16], edx
0xb1ea: cmp     ebx, -1
0xb1ed: je      0xb1ae
0xb1ef: test    r14, r14
0xb1f2: je      0xb1fb
0xb1f4: mov     dword ptr [r14], 1
0xb1fb: mov     eax, dword ptr [rip + 0x201fb]
0xb201: test    ebx, ebx
0xb203: jne     0xb445
0xb209: mov     ebx, 0xff
0xb20e: test    eax, eax
0xb210: jne     0xb240
0xb212: xor     r12d, r12d
0xb215: mov     ebp, 0x1fe
0xb21a: mov     ebx, 0xff
0xb21f: call    0x4f00
0xb224: mov     rax, qword ptr [rax]
0xb227: movzx   eax, word ptr [rax + rbp]
0xb22b: test    ah, 0x40
0xb22e: jne     0xb47d
0xb234: test    r12b, r12b
0xb237: jne     0xb47d
0xb23d: nop     dword ptr [rax]
0xb240: mov     rax, qword ptr [rip + 0x1f739]
0xb247: xor     r12d, r12d
0xb24a: lea     r13, [rip + 0x1f8af]
0xb251: mov     dword ptr [rip + 0x1f751], 0
0xb25b: mov     dword ptr [rip + 0x1f743], 0
0xb265: mov     qword ptr [rip + 0x1f70c], rax
0xb26c: test    ebx, ebx
0xb26e: mov     eax, 0xffffffff
0xb273: movsxd  rbp, r12d
0xb276: cmovne  eax, ebx
0xb279: mov     rbx, qword ptr [rip + 0x200e0]
0xb280: mov     byte ptr [rbx + rbp], al
0xb283: add     rbp, 1
0xb287: mov     byte ptr [rbx + rbp], 0
0xb28b: cmp     qword ptr [rip + 0x1f6e5], 0
0xb293: je      0xb2b3
0xb295: mov     rdi, rbx
0xb298: call    0x99e0
0xb29d: mov     r15, rax
0xb2a0: cmp     rax, -1
0xb2a4: je      0xb33a
0xb2aa: test    rax, rax
0xb2ad: jne     0xb38f
0xb2b3: mov     ecx, dword ptr [rip + 0x1fc4b]

```

```

0xb2b9: test    ecx, ecx
0xb2bb: jne      0xb2ee
0xb2bd: mov      edx, dword ptr [rip + 0x1f6e5]
0xb2c3: test    edx, edx
0xb2c5: je       0xb380
0xb2cb: mov      rdi, rbx
0xb2ce: call    0xb080
0xb2d3: mov      dword ptr [rip + 0x1f6cf], 1
0xb2dd: jmp      0xb2ee
0xb2ee:
0xb2df: nop
0xb2e0: xor      eax, eax
0xb2e2: call    0xad50
0xb2e7: mov      ebx, eax
0xb2e9: cmp      ebx, -1
0xb2ec: jne      0xb32c
0xb2ee: xor      eax, eax
0xb2f0: call    0x18ab0
0xb2f5: mov      eax, dword ptr [rip + 0x1fc09]
0xb2fb: test    eax, eax
0xb2fd: je       0xb2e0
0xb2ff: movsxd   rsi, dword ptr [rip + 0x1fbfa]
0xb306: movsx    ebx, byte ptr [r13 + rsi]
0xb30c: mov      rdx, rsi
0xb30f: mov      ecx, ebx
0xb311: test    bl, bl
0xb313: js       0xb360
0xb315: sub      eax, 1
0xb318: add      edx, 1
0xb31b: mov      dword ptr [rip + 0x1fbe3], eax
0xb321: mov      dword ptr [rip + 0x1fbd9], edx
0xb327: cmp      ebx, -1
0xb32a: je       0xb2ee
0xb32c: add      r12d, 1
0xb330: cmp      r12d, 0xf
0xb334: jne      0xb26c
0xb33a: mov      edi, 1
0xb33f: xor      r15d, r15d
0xb342: call    0x4aa0
0xb347: mov      dword ptr [rip + 0x1f65b], 0
0xb351: jmp      0xb433
0xb433:
0xb356: nop      word ptr cs:[rax + rax]
0xb360: and      ecx, 0x7f
0xb363: add      r12d, 1
0xb367: mov      byte ptr [r13 + rsi], cl
0xb36c: cmp      r12d, 0xf
0xb370: je       0xb33a

```

```

0xb372: mov     eax, 0x1b
0xb377: jmp     0xb279
0xb279:
0xb37c: nop     dword ptr [rax]
0xb380: mov     edi, 1
0xb385: call    0x4aa0
0xb38a: jmp     0xb2d3
0xb2d3:
0xb38f: test    r14, r14
0xb392: je      0xb41f
0xb398: mov     ecx, dword ptr [rip + 0x1fb66]
0xb39e: cmp     r12d, ecx
0xb3a1: jge     0xb41f
0xb3a3: movsxd  rax, dword ptr [rip + 0x1fb56]
0xb3aa: lea     r9d, [r12 + 1]
0xb3af: lea     rdx, [rip + 0x1f74a]
0xb3b6: movsxd  rdi, r9d
0xb3b9: mov     qword ptr [rsp + 0x18], rdi
0xb3be: lea     r8d, [r9 + rax]
0xb3c2: lea     r13, [rax + rdx]
0xb3c6: jmp     0xb3f1
0xb3f1:
0xb3c8: nop     dword ptr [rax + rax]
0xb3d0: sub     ecx, r9d
0xb3d3: add     dword ptr [r14], 1
0xb3d7: add     r13, qword ptr [rsp + 0x18]
0xb3dc: mov     dword ptr [rip + 0x1fb1d], r8d
0xb3e3: add     r8d, r9d
0xb3e6: mov     dword ptr [rip + 0x1fb18], ecx
0xb3ec: cmp     r12d, ecx
0xb3ef: jge     0xb41f
0xb3f1: mov     rdx, rbp
0xb3f4: mov     rsi, r13
0xb3f7: mov     rdi, rbx
0xb3fa: mov     dword ptr [rsp + 0x14], r8d
0xb3ff: mov     dword ptr [rsp + 0x10], r9d
0xb404: mov     dword ptr [rsp + 0xc], ecx
0xb408: call    0x4af0
0xb40d: mov     ecx, dword ptr [rsp + 0xc]
0xb411: mov     r9d, dword ptr [rsp + 0x10]
0xb416: test    eax, eax
0xb418: mov     r8d, dword ptr [rsp + 0x14]
0xb41d: je      0xb3d0
0xb41f: mov     edi, 1
0xb424: call    0x4aa0
0xb429: mov     dword ptr [rip + 0x1f579], 0
0xb433: add     rsp, 0x28
0xb437: mov     rax, r15

```



```

0xb43a: pop    rbx
0xb43b: pop    rbp
0xb43c: pop    r12
0xb43e: pop    r13
0xb440: pop    r14
0xb442: pop    r15
0xb444: ret
0xb445: test   eax, eax
0xb447: jne    0xb240
0xb44d: cmp    ebx, 0xa
0xb450: sete   bpl
0xb454: cmp    ebx, 0xd
0xb457: sete   al
0xb45a: or     ebp, eax
0xb45c: call   0x4f00
0xb461: mov    rax, qword ptr [rax]
0xb464: test   bpl, bpl
0xb467: je     0xb4c6
0xb469: mov    ebx, 0xa
0xb46e: test   byte ptr [rax + 0x15], 0x40
0xb472: je     0xb240
0xb478: mov    ebx, 0xa
0xb47d: mov    rax, qword ptr [rip + 0x1f4dc]
0xb484: lea    r15, [rip + 0x1f4d5]
0xb48b: mov    byte ptr [rax], bl
0xb48d: mov    rax, qword ptr [rip + 0x1f4cc]
0xb494: mov    byte ptr [rax + 1], 0
0xb498: jmp    0xb433
0xb433:
0xb49a: nop    word ptr [rax + rax]
0xb4a0: and    ecx, 0x7f
0xb4a3: mov    byte ptr [rbp + rsi], cl
0xb4a7: test   r14, r14
0xb4aa: je     0xb4b3
0xb4ac: mov    dword ptr [r14], 1
0xb4b3: mov    eax, dword ptr [rip + 0x1ff43]
0xb4b9: mov    ebx, 0x1b
0xb4be: test   eax, eax
0xb4c0: jne    0xb240
0xb4c6: movsxd rbp, ebx
0xb4c9: add    rbp, rbp
0xb4cc: cmp    ebx, 7
0xb4cf: sete   r12b
0xb4d3: jmp    0xb21f
0xb21f:
0xb4d8: nop    dword ptr [rax + rax]
0xb4e0: endbr64
0xb4e4: mov    eax, dword ptr [rip + 0x1f4c2]

```

```

0xb4ea: test    eax, eax
0xb4ec: jne      0xb4f0
0xb4ee: ret
0xb4ef: nop
0xb4f0: sub      rsp, 8
0xb4f4: mov      rdi, qword ptr [rip + 0x1fe65]
0xb4fb: call     0xb080
0xb500: mov      dword ptr [rip + 0x1f49e], 1
0xb50a: add      rsp, 8
0xb50e: ret
0xb50f: nop
0xb510: endbr64
0xb514: mov      rax, qword ptr [rip + 0x1fe45]
0xb51b: ret
0xb51c: nop      dword ptr [rax]
0xb520: endbr64
0xb524: push     r15
0xb526: mov      esi, 0x5413
0xb52b: mov      edi, 1
0xb530: push     r14
0xb532: push     r13
0xb534: push     r12
0xb536: push     rbp
0xb537: push     rbx
0xb538: sub      rsp, 0x58
0xb53c: mov      rax, qword ptr fs:[0x28]
0xb545: mov      qword ptr [rsp + 0x48], rax
0xb54a: xor      eax, eax
0xb54c: lea      rdx, [rsp + 0x18]
0xb551: mov      dword ptr [rsp + 0x10], 0
0xb559: mov      dword ptr [rsp + 0x14], 0
0xb561: call     0x4a70
0xb566: test     eax, eax
0xb568: jne      0xb5a0
0xb56a: mov      ebx, eax
0xb56c: movzx    eax, word ptr [rsp + 0x1a]
0xb571: test     ax, ax
0xb574: je       0xb790
0xb57a: movzx    ebp, word ptr [rsp + 0x18]
0xb57f: test     bp, bp
0xb582: je       0xb790
0xb588: movzx    ebx, ax
0xb58b: lea      r12d, [rbp - 1]
0xb58f: lea      eax, [rbx - 1]
0xb592: mov      dword ptr [rsp + 0xc], eax
0xb596: jmp      0xb5b2
0xb5b2:
0xb598: nop      dword ptr [rax + rax]

```

```

0xb5a0: mov     dword ptr [rsp + 0xc], 0xffffffff
0xb5a8: mov     r12d, 0xffffffff
0xb5ae: xor     ebx, ebx
0xb5b0: xor     ebp, ebp
0xb5b2: lea     r13, [rip + 0x15667]
0xb5b9: lea     r15, [rip + 0x15a13]
0xb5c0: mov     rdi, r13
0xb5c3: call    0x47e0
0xb5c8: mov     rdi, rax
0xb5cb: test    rax, rax
0xb5ce: je      0xb5df
0xb5d0: lea     rdx, [rsp + 0x14]
0xb5d5: mov     rsi, r15
0xb5d8: xor     eax, eax
0xb5da: call    0x4c60
0xb5df: lea     r14, [rip + 0x15642]
0xb5e6: mov     rdi, r14
0xb5e9: call    0x47e0
0xb5ee: mov     rdi, rax
0xb5f1: test    rax, rax
0xb5f4: je      0xb605
0xb5f6: lea     rdx, [rsp + 0x10]
0xb5fb: mov     rsi, r15
0xb5fe: xor     eax, eax
0xb600: call    0x4c60
0xb605: cmp     dword ptr [rsp + 0xc], 0x3ff
0xb60d: mov     eax, dword ptr [rip + 0x1e8fd]
0xb613: jbe     0xb628
0xb615: mov     ebx, dword ptr [rsp + 0x14]
0xb619: lea     edx, [rbx - 1]
0xb61c: cmp     edx, 0x3ff
0xb622: ja      0xb7c0
0xb628: mov     dword ptr [rip + 0x1f36a], ebx
0xb62e: cmp     r12d, 0x3ff
0xb635: jbe     0xb64a
0xb637: mov     ebp, dword ptr [rsp + 0x10]
0xb63b: lea     edx, [rbp - 1]
0xb63e: cmp     edx, 0x3ff
0xb644: ja      0xb7a8
0xb64a: lea     r12, [rsp + 0x20]
0xb64f: mov     r8d, ebp
0xb652: mov     rcx, r15
0xb655: mov     edx, 0x20
0xb65a: mov     esi, 1
0xb65f: mov     rdi, r12
0xb662: xor     eax, eax
0xb664: mov     dword ptr [rip + 0x1f332], ebp
0xb66a: call    0x4f10

```

```

0xb66f: mov     rsi, r12
0xb672: mov     rdi, r14
0xb675: call    0x18380
0xb67a: mov     rcx, r15
0xb67d: mov     edx, 0x20
0xb682: mov     rdi, r12
0xb685: mov     r8d, dword ptr [rip + 0x1f30c]
0xb68c: mov     esi, 1
0xb691: xor     eax, eax
0xb693: call    0x4f10
0xb698: mov     rdi, r13
0xb69b: mov     rsi, r12
0xb69e: call    0x18380
0xb6a3: mov     rdi, qword ptr [rip + 0x1fc8e]
0xb6aa: test    rdi, rdi
0xb6ad: je      0xb6b4
0xb6af: call    0x18d10
0xb6b4: mov     rdi, qword ptr [rip + 0x1fc75]
0xb6bb: test    rdi, rdi
0xb6be: je      0xb6c5
0xb6c0: call    0x18d10
0xb6c5: mov     rdi, qword ptr [rip + 0x1fc5c]
0xb6cc: test    rdi, rdi
0xb6cf: je      0xb6d6
0xb6d1: call    0x18d10
0xb6d6: mov     rdi, qword ptr [rip + 0x1fc43]
0xb6dd: test    rdi, rdi
0xb6e0: je      0xb6e7
0xb6e2: call    0x18d10
0xb6e7: mov     edi, dword ptr [rip + 0x1f2ab]
0xb6ed: imul    edi, dword ptr [rip + 0x1f2a8]
0xb6f4: mov     esi, 1
0xb6f9: movsxd  rdi, edi
0xb6fc: call    0x1d740
0xb701: mov     edi, dword ptr [rip + 0x1f291]
0xb707: mov     esi, 1
0xb70c: imul    edi, dword ptr [rip + 0x1f289]
0xb713: mov     qword ptr [rip + 0x1fc1e], rax
0xb71a: movsxd  rdi, edi
0xb71d: call    0x1d740
0xb722: mov     edi, dword ptr [rip + 0x1f270]
0xb728: mov     esi, 1
0xb72d: imul    edi, dword ptr [rip + 0x1f268]
0xb734: mov     qword ptr [rip + 0x1fbf5], rax
0xb73b: movsxd  rdi, edi
0xb73e: call    0x1d740
0xb743: mov     edi, dword ptr [rip + 0x1f24f]
0xb749: mov     esi, 1

```

```

0xb74e: imul    edi, dword ptr [rip + 0x1f247]
0xb755: mov     qword ptr [rip + 0x1fbcc], rax
0xb75c: movsxd  rdi, edi
0xb75f: call    0x1d740
0xb764: mov     qword ptr [rip + 0x1fbb5], rax
0xb76b: mov     rax, qword ptr [rsp + 0x48]
0xb770: sub     rax, qword ptr fs:[0x28]
0xb779: jne     0xb7dc
0xb77b: add     rsp, 0x58
0xb77f: pop     rbx
0xb780: pop     rbp
0xb781: pop     r12
0xb783: pop     r13
0xb785: pop     r14
0xb787: pop     r15
0xb789: ret
0xb78a: nop     word ptr [rax + rax]
0xb790: mov     dword ptr [rsp + 0xc], 0xffffffff
0xb798: xor     ebp, ebp
0xb79a: mov     r12d, 0xffffffff
0xb7a0: jmp     0xb5b2
0xb5b2
0xb7a5: nop     dword ptr [rax]
0xb7a8: lea     edx, [rax - 1]
0xb7ab: mov     ebp, 0x18
0xb7b0: cmp     edx, 0x400
0xb7b6: cmovb   ebp, eax
0xb7b9: jmp     0xb64a
0xb64a
0xb7be: nop
0xb7c0: mov     ebx, dword ptr [rip + 0x1e72a]
0xb7c6: lea     edx, [rbx - 1]
0xb7c9: cmp     edx, 0x3ff
0xb7cf: mov     edx, 0x50
0xb7d4: cmova   ebx, edx
0xb7d7: jmp     0xb628
0xb628
0xb7dc: call    0x49d0
0xb7e1: nop     word ptr cs:[rax + rax]
0xb7ec: nop     dword ptr [rax]
0xb7f0: endbr64
0xb7f4: push    r15
0xb7f6: push    r14
0xb7f8: push    r13
0xb7fa: push    r12
0xb7fc: mov     r12, rdi
0xb7ff: mov     edi, 0x5858
0xb804: push    rbp

```

```

0xb805: push    rbx
0xb806: mov     ebx, esi
0xb808: sub     rsp, 0x38
0xb80c: mov     rax, qword ptr fs:[0x28]
0xb815: mov     qword ptr [rsp + 0x28], rax
0xb81a: movabs  rax, 0x7363762f7665642f
0xb824: mov     word ptr [rsp + 8], di
0xb829: mov     byte ptr [rsp + 0xa], 0
0xb82e: mov     dword ptr [rsp + 0x18], 0x585861
0xb836: mov     qword ptr [rsp], rax
0xb83a: mov     qword ptr [rsp + 0x10], rax
0xb83f: cmp     esi, 1
0xb842: je      0xb978
0xb848: mov     esi, dword ptr [rip + 0x1fbb2]
0xb84e: mov     r13, qword ptr [rip + 0x1d783]
0xb855: mov     r14d, 1
0xb85b: test    esi, esi
0xb85d: jne     0xb90c
0xb863: mov     rax, qword ptr [rip + 0x1f126]
0xb86a: lea     r15, [rsp + 0x10]
0xb86f: mov     esi, r14d
0xb872: mov     dword ptr [rip + 0x1fb84], 0
0xb87c: mov     rdi, r15
0xb87f: movzx   edx, byte ptr [rax + 8]
0xb883: mov     byte ptr [rsp + 0x19], dl
0xb887: movzx   eax, byte ptr [rax + 9]
0xb88b: mov     byte ptr [rsp + 0x1a], al
0xb88f: call    0x4930
0xb894: mov     ebp, eax
0xb896: cmp     eax, -1
0xb899: je      0xb8f0
0xb89b: mov     eax, dword ptr [rip + 0x1f0fb]
0xb8a1: imul    eax, dword ptr [rip + 0x1f0f0]
0xb8a8: mov     edi, ebp
0xb8aa: mov     rsi, r12
0xb8ad: lea     edx, [rax + rax + 4]
0xb8b1: xor     eax, eax
0xb8b3: call    r13
0xb8b6: mov     edi, ebp
0xb8b8: call    0x4ab0
0xb8bd: cmp     ebx, 2
0xb8c0: je      0xb990
0xb8c6: mov     eax, 1
0xb8cb: mov     rdx, qword ptr [rsp + 0x28]
0xb8d0: sub     rdx, qword ptr fs:[0x28]
0xb8d9: jne     0xbb07
0xb8df: add     rsp, 0x38
0xb8e3: pop     rbx

```

```

0xb8e4: pop    rbp
0xb8e5: pop    r12
0xb8e7: pop    r13
0xb8e9: pop    r14
0xb8eb: pop    r15
0xb8ed: ret
0xb8ee: nop
0xb8f0: mov     ecx, 0x30
0xb8f5: mov     esi, r14d
0xb8f8: mov     rdi, r15
0xb8fb: mov     word ptr [rsp + 0x19], cx
0xb900: call    0x4930
0xb905: mov     ebp, eax
0xb907: cmp     eax, -1
0xb90a: jne     0xb89b
0xb90c: mov     dword ptr [rip + 0x1faea], 1
0xb916: cmp     ebx, 2
0xb919: je      0xba30
0xb91f: mov     rax, qword ptr [rip + 0x1f06a]
0xb926: mov     r15, rsp
0xb929: mov     esi, r14d
0xb92c: mov     rdi, r15
0xb92f: movzxb  edx, byte ptr [rax + 8]
0xb933: movzxb  eax, byte ptr [rax + 9]
0xb937: mov     byte ptr [rsp + 8], dl
0xb93b: mov     byte ptr [rsp + 9], al
0xb93f: call    0x4930
0xb944: mov     ebp, eax
0xb946: cmp     eax, -1
0xb949: je      0xba00
0xb94f: mov     edx, dword ptr [rip + 0x1f043]
0xb955: mov     edi, ebp
0xb957: mov     rsi, r12
0xb95a: xor     eax, eax
0xb95c: imul    edx, dword ptr [rip + 0x1f039]
0xb963: add     edx, 4
0xb966: call    r13
0xb969: mov     edi, ebp
0xb96b: call    0x4ab0
0xb970: jmp     0xb8c6
0xb8c6:
0xb975: nop     dword ptr [rax]
0xb978: mov     r13, qword ptr [rip + 0x1d661]
0xb97f: xor     r14d, r14d
0xb982: jmp     0xb863
0xb863:
0xb987: nop     word ptr [rax + rax]
0xb990: movsxb  esi, byte ptr [r12 + 2]

```

```

0xb996: movsx    edi, byte ptr [r12 + 3]
0xb99c: call     0xa7a0
0xb9a1: mov      edx, dword ptr [rip + 0x1f561]
0xb9a7: test     edx, edx
0xb9a9: jle      0xb9e5
0xb9ab: xor      ebp, ebp
0xb9ad: lea      rbx, [rip + 0x1f56c]
0xb9b4: jmp      0xb9cc
0xb9cc:
0xb9b6: nop      word ptr cs:[rax + rax]
0xb9c0: mov      edx, dword ptr [rip + 0x1f542]
0xb9c6: add      ebp, eax
0xb9c8: cmp      edx, ebp
0xb9ca: jle      0xb9e5
0xb9cc: sub      edx, ebp
0xb9ce: movsxd   rsi, ebp
0xb9d1: mov      edi, 1
0xb9d6: add      rsi, rbx
0xb9d9: movsxd   rdx, edx
0xb9dc: call     0x18da0
0xb9e1: test     eax, eax
0xb9e3: jns      0xb9c0
0xb9e5: mov      dword ptr [rip + 0x1f519], 0
0xb9ef: mov      eax, 1
0xb9f4: jmp      0xb8cb
0xb8cb:
0xb9f9: nop      dword ptr [rax]
0xba00: mov      edx, 0x30
0xba05: mov      esi, r14d
0xba08: mov      rdi, r15
0xba0b: mov      word ptr [rsp + 8], dx
0xba10: call     0x4930
0xba15: mov      ebp, eax
0xba17: cmp      eax, -1
0xba1a: jne      0xb94f
0xba20: xor      eax, eax
0xba22: jmp      0xb8cb
0xb8cb:
0xba27: nop      word ptr [rax + rax]
0xba30: xor      eax, eax
0xba32: mov      r15, rsp
0xba35: call     0xa300
0xba3a: mov      rax, qword ptr [rip + 0x1ef4f]
0xba41: mov      esi, r14d
0xba44: mov      rdi, r15
0xba47: movzx    edx, byte ptr [rax + 8]
0xba4b: movzx    eax, byte ptr [rax + 9]
0xba4f: mov      byte ptr [rsp + 8], dl

```



```

0xba53: mov     byte ptr [rsp + 9], al
0xba57: call    0x4930
0xba5c: mov     ebp, eax
0xba5e: cmp     eax, -1
0xba61: je      0xbae0
0xba63: mov     edx, dword ptr [rip + 0x1ef2f]
0xba69: mov     rsi, r12
0xba6c: mov     edi, ebp
0xba6e: xor     eax, eax
0xba70: imul    edx, dword ptr [rip + 0x1ef25]
0xba77: add     edx, 4
0xba7a: call    r13
0xba7d: mov     edi, ebp
0xba7f: call    0x4ab0
0xba84: mov     eax, dword ptr [rip + 0x1ef12]
0xba8a: xor     esi, esi
0xba8c: lea     edi, [rax - 1]
0xba8f: call    0xa7a0
0xba94: mov     edx, dword ptr [rip + 0x1f46e]
0xba9a: test    edx, edx
0xba9c: jle     0xb9e5
0xbaa2: xor     ebp, ebp
0xbaa4: lea     rbx, [rip + 0x1f475]
0xbaab: jmp     0xbac0
0xbac0:
0xbaad: nop     dword ptr [rax]
0xbab0: mov     edx, dword ptr [rip + 0x1f452]
0xbab6: add     ebp, eax
0xbab8: cmp     edx, ebp
0xbaba: jle     0xb9e5
0xbac0: sub     edx, ebp
0xbac2: movsxd  rsi, ebp
0xbac5: mov     edi, 1
0xbaca: add     rsi, rbx
0xbacd: movsxd  rdx, edx
0xbad0: call    0x18da0
0xbad5: test    eax, eax
0xbad7: jns     0xbab0
0xbad9: jmp     0xb9e5
0xb9e5:
0xbade: nop
0xbae0: mov     eax, 0x30
0xbae5: mov     esi, r14d
0xbae8: mov     rdi, r15
0xbaeb: mov     word ptr [rsp + 8], ax
0xbaf0: call    0x4930
0xbaf5: mov     ebp, eax
0xbaf7: cmp     eax, -1

```

```

0xbafa: jne      0xba63
0xbb00: xor       eax, eax
0xbb02: jmp       0xb8cb
0xb8cb:
0xbb07: call      0x49d0
0xbb0c: nop       dword ptr [rax]
0xbb10: endbr64
0xbb14: mov       eax, dword ptr [rip + 0x1ee6e]
0xbb1a: test      eax, eax
0xbb1c: jne       0xbb20
0xbb1e: ret
0xbb1f: nop
0xbb20: sub       rsp, 8
0xbb24: mov       esi, 1
0xbb29: xor       eax, eax
0xbb2b: call      0xb7f0
0xbb30: mov       dword ptr [rip + 0x1f8ce], eax
0xbb36: add       rsp, 8
0xbb3a: ret
0xbb3b: nop       dword ptr [rax + rax]
0xbb40: endbr64
0xbb44: push      rbp
0xbb45: xor       eax, eax
0xbb47: mov       rbp, rdi
0xbb4a: call      0xa6c0
0xbb4f: mov       edx, dword ptr [rip + 0x1ee33]
0xbb55: test      edx, edx
0xbb57: je        0xbb94
0xbb59: mov       eax, dword ptr [rip + 0x1f8a5]
0xbb5f: test      eax, eax
0xbb61: je        0xbb94
0xbb63: mov       edx, dword ptr [rip + 0x1ee33]
0xbb69: imul      edx, dword ptr [rip + 0x1ee28]
0xbb70: xor       esi, esi
0xbb72: mov       rdi, qword ptr [rip + 0x1f7af]
0xbb79: movsxd    rdx, edx
0xbb7c: call      0x4a50
0xbb81: xor       eax, eax
0xbb83: mov       esi, 2
0xbb88: mov       rdi, rbp
0xbb8b: call      0xb7f0
0xbb90: test      eax, eax
0xbb92: jne       0xbba0
0xbb94: xor       eax, eax
0xbb96: pop       rbp
0xbb97: jmp       0xa300
0xa300:
0xbb9c: nop       dword ptr [rax]

```

```

0xbba0: mov     edx, dword ptr [rip + 0x1edf6]
0xbba6: imul    edx, dword ptr [rip + 0x1edeb]
0xbbad: xor     esi, esi
0xbbaef: mov     rdi, qword ptr [rip + 0x1f782]
0xbbbb6: pop     rbp
0xbbbb7: movsxd  rdx, edx
0xbbbba: jmp     0x4a50
0x4a50
0xbbbbf: nop
0xbbbc0: endbr64
0xbbbc4: push    rbp
0xbbbc5: mov     rbp, rdi
0xbbbc8: push    rbx
0xbbbc9: sub     rsp, 8
0xbbbcd: cmp     dword ptr [rip + 0x1ed84], 1
0xbbbd4: je      0xbbcd0
0xbbbda: xor     eax, eax
0xbbbdc: call    0xa6c0
0xbbbe1: test    rbp, rbp
0xbbbe4: je      0xbbbf4
0xbbbe6: mov     eax, dword ptr [rip + 0x1ed9c]
0xbbec: test    eax, eax
0xbbee: jne     0xbcc0
0xbbbf4: mov     eax, dword ptr [rip + 0x1eda2]
0xbbbfa: mov     rdi, qword ptr [rip + 0x1e347]
0xbc01: lea     rdx, [rip - 0x2198]
0xbc08: lea     esi, [rax - 1]
0xbc0b: call    0x4820
0xbc10: mov     edx, dword ptr [rip + 0x1f2f2]
0xbc16: test    edx, edx
0xbc18: jle     0xbc4d
0xbc1a: xor     ebp, ebp
0xbc1c: lea     rbx, [rip + 0x1f2fd]
0xbc23: jmp     0xbc34
0xbc34
0xbc25: nop     dword ptr [rax]
0xbc28: mov     edx, dword ptr [rip + 0x1f2da]
0xbc2e: add     ebp, eax
0xbc30: cmp     edx, ebp
0xbc32: jle     0xbc4d
0xbc34: sub     edx, ebp
0xbc36: movsxd  rsi, ebp
0xbc39: mov     edi, 1
0xbc3e: add     rsi, rbx
0xbc41: movsxd  rdx, edx
0xbc44: call    0x18da0
0xbc49: test    eax, eax
0xbc4b: jns     0xbc28

```

```

0xbc4d: mov     edi, dword ptr [rip + 0x1ed49]
0xbc53: xor     esi, esi
0xbc55: mov     dword ptr [rip + 0x1f2a9], 0
0xbc5f: call    0xa7a0
0xbc64: mov     edx, dword ptr [rip + 0x1f29e]
0xbc6a: test    edx, edx
0xbc6c: jle     0xbca5
0xbc6e: xor     ebp, ebp
0xbc70: lea     rbx, [rip + 0x1f2a9]
0xbc77: jmp     0xbc8c
0xbc8c:
0xbc79: nop     dword ptr [rax]
0xbc80: mov     edx, dword ptr [rip + 0x1f282]
0xbc86: add     ebp, eax
0xbc88: cmp     edx, ebp
0xbc8a: jle     0xbca5
0xbc8c: sub     edx, ebp
0xbc8e: movsxd  rsi, ebp
0xbc91: mov     edi, 1
0xbc96: add     rsi, rbx
0xbc99: movsxd  rdx, edx
0xbc9c: call    0x18da0
0xbca1: test    eax, eax
0xbca3: jns     0xbc80
0xbca5: mov     dword ptr [rip + 0x1f259], 0
0xbcaf: add     rsp, 8
0xbcb3: mov     edi, 0xa
0xbcb8: pop     rbx
0xbcb9: pop     rbp
0xbcba: jmp     0x4810
0x4810:
0xbcbf: nop
0xbcc0: mov     rdi, rbp
0xbcc3: call    0xbb40
0xbcc8: jmp     0xbbf4
0xbbf4:
0xbccd: nop     dword ptr [rax]
0xbcd0: lea     rdx, [rip + 0x1f6e9]
0xbcd7: mov     esi, 1
0xbcdc: mov     edi, 1
0xbce1: call    0x4d70
0xbce6: mov     dword ptr [rip + 0x1ec68], 0
0xbcf0: jmp     0xbbda
0xbbda:
0xbcf5: nop     word ptr cs:[rax + rax]
0xbd00: endbr64
0xbd04: push    r13
0xbd06: lea     r13, [rip + 0x1cc33]

```

```

0xbd0d: push    r12
0xbd0f: push    rbp
0xbd10: mov     rbp, rdi
0xbd13: lea     rdi, [rip + 0x14f14]
0xbd1a: push    rbx
0xbd1b: xor     ebx, ebx
0xbd1d: sub     rsp, 8
0xbd21: jmp     0xbd2d
0xbd2d:
0xbd23: nop     dword ptr [rax + rax]
0xbd28: mov     rdi, qword ptr [r13 + rbx*8]
0xbd2d: mov     rsi, rbp
0xbd30: mov     r12d, ebx
0xbd33: call    0x4b30
0xbd38: test    eax, eax
0xbd3a: je      0xbd60
0xbd3c: add     rbx, 1
0xbd40: cmp     rbx, 0xa
0xbd44: jne     0xbd28
0xbd46: add     rsp, 8
0xbd4a: mov     r12d, 0xffffffff
0xbd50: pop     rbx
0xbd51: mov     eax, r12d
0xbd54: pop     rbp
0xbd55: pop     r12
0xbd57: pop     r13
0xbd59: ret
0xbd5a: nop     word ptr [rax + rax]
0xbd60: lea     eax, [rbx - 8]
0xbd63: cmp     ebx, 7
0xbd66: cmovg   r12d, eax
0xbd6a: add     rsp, 8
0xbd6e: pop     rbx
0xbd6f: pop     rbp
0xbd70: mov     eax, r12d
0xbd73: pop     r12
0xbd75: pop     r13
0xbd77: ret
0xbd78: nop     dword ptr [rax + rax]
0xbd80: endbr64
0xbd84: push    r12
0xbd86: mov     r12, rdi
0xbd89: push    rbp
0xbd8a: lea     rbp, [rip + 0x1e1e7]
0xbd91: push    rbx
0xbd92: mov     ebx, 0x11
0xbd97: jmp     0xbdac
0xbdac:

```

```

0xbd99: nop      dword ptr [rax]
0xbda0: add      ebx, 1
0xbda3: add      rbp, 0x20
0xbda7: cmp      ebx, 0x26
0xbdaa: je       0xbdd8
0xbdac: mov      rdi, qword ptr [rbp]
0xbdb0: mov      rsi, r12
0xbdb3: call     0x4b30
0xbdb8: test     eax, eax
0xbdba: jne      0xbda0
0xbdbc: movsxd   rbx, ebx
0xbdbf: lea      rax, [rip + 0x1df7a]
0xbdc6: shl      rbx, 5
0xbdca: mov      rax, qword ptr [rax + rbx + 8]
0xbdcf: pop      rbx
0xbdd0: pop      rbp
0xbdd1: pop      r12
0xbdd3: ret
0xbdd4: nop      dword ptr [rax]
0xbdd8: pop      rbx
0xbdd9: xor      eax, eax
0xbddb: pop      rbp
0xbddc: pop      r12
0xbdde: ret
0xbddf: nop
0xbde0: endbr64
0xbde4: push     r14
0xbde6: lea      rdi, [rip + 0x1479d]
0xbded: push     r13
0xbdef: push     r12
0xbdf1: push     rbp
0xbdf2: push     rbx
0xbdf3: sub      rsp, 0xb0
0xbdfa: mov      rax, qword ptr fs:[0x28]
0xbe03: mov      qword ptr [rsp + 0xa8], rax
0xbe0b: xor      eax, eax
0xbe0d: call     0x47e0
0xbe12: lea      rsi, [rsp + 0x10]
0xbe17: mov      edi, 1
0xbe1c: mov      rbp, rax
0xbe1f: call     0x4eb0
0xbe24: mov      rax, qword ptr [rsp + 0x38]
0xbe29: shr      rax, 8
0xbe2d: cmp      rax, 4
0xbe31: sete     al
0xbe34: movzx    eax, al
0xbe37: mov      dword ptr [rip + 0x1eb4b], eax
0xbe3d: test     rbp, rbp

```

```

0xbe40: je      0xc36d
0xbe46: mov     rdi, rbp
0xbe49: call    0x49b0
0xbe4e: cmp     rax, 0x3f
0xbe52: ja      0xc308
0xbe58: lea     r12, [rip + 0x14ff1]
0xbe5f: lea     rbx, [rip + 0x14d6f]
0xbe66: lea     r14, [rip + 0x14e02]
0xbe6d: lea     r13, [rip + 0x14e07]
0xbe74: nop     dword ptr [rax]
0xbe78: xor     edi, edi
0xbe7a: mov     rsi, rbp
0xbe7d: call    0x4e90
0xbe82: cmp     eax, -1
0xbe85: je      0xbf38
0xbe8b: test    eax, eax
0xbe8d: jne     0xbff8
0xbe93: mov     rdi, qword ptr [rip + 0x1e9e6]
0xbe9a: mov     esi, 1
0xbe9f: mov     r8, rbp
0xbea2: mov     rdx, r12
0xbea5: mov     rcx, qword ptr [rip + 0x1ea3c]
0xbeac: lea     r9, [rip + 0x1de5d]
0xebb3: call    0x4e60
0xebb8: mov     rsi, rbx
0xebbb: mov     rdi, rbp
0xebbe: call    0x4b30
0xbec3: test    eax, eax
0xbec5: je      0xbf98
0xbecb: mov     rdi, rbp
0xbece: call    0x9940
0xbed3: test    eax, eax
0xbed5: jne     0xbfc8
0xbedb: mov     rsi, r14
0xbede: mov     rdi, rbp
0xbee1: call    0x4b30
0xbee6: test    eax, eax
0xbee8: je      0xbf01
0xbeea: lea     rsi, [rip + 0x14d84]
0xbef1: mov     rdi, rbp
0xbef4: call    0x4b30
0xbef9: test    eax, eax
0xbefb: jne     0xc3b4
0xbf01: mov     rdi, qword ptr [rip + 0x1e978]
0xbf08: mov     rcx, qword ptr [rip + 0x1e9d9]
0xbf0f: mov     esi, 1
0xbf14: xor     eax, eax
0xbf16: mov     rdx, r13

```

```

0xbf19: lea      rbp, [rip + 0x14cba]
0xbf20: call     0x4e60
0xbf25: xor      edi, edi
0xbf27: mov      rsi, rbp
0xbf2a: call     0x4e90
0xbf2f: cmp      eax, -1
0xbf32: jne      0xbe8b
0xbf38: mov      rcx, qword ptr [rip + 0x1e9a9]
0xbf3f: mov      rdi, qword ptr [rip + 0x1e93a]
0xbf46: xor      eax, eax
0xbf48: lea      r8, [rip + 0x1ddc1]
0xbf4f: lea      rdx, [rip + 0x14ea2]
0xbf56: mov      esi, 1
0xbf5b: call     0x4e60
0xbf60: mov      rdi, qword ptr [rip + 0x1e919]
0xbf67: xor      eax, eax
0xbf69: mov      rcx, qword ptr [rip + 0x1e978]
0xbf70: lea      r8, [rip + 0x1dd89]
0xbf77: lea      rdx, [rip + 0x14ea2]
0xbf7e: mov      esi, 1
0xbf83: call     0x4e60
0xbf88: mov      edi, 1
0xbf8d: call     0x4e20
0xbf92: nop      word ptr [rax + rax]
0xbf98: mov      rcx, qword ptr [rip + 0x1e949]
0xbf9f: mov      rdi, qword ptr [rip + 0x1e8da]
0xbfa6: lea      rdx, [rip + 0x14c99]
0xbfad: mov      esi, 1
0xbfb2: lea      rbp, [rip + 0x14c21]
0xbfb9: call     0x4e60
0xbfbe: jmp      0xbe78
0xbe78:
0xbfc3: nop      dword ptr [rax + rax]
0xbfc8: mov      rcx, qword ptr [rip + 0x1e919]
0xbfcf: mov      rdi, qword ptr [rip + 0x1e8aa]
0xbfd6: lea      rdx, [rip + 0x14c7d]
0xbfdd: xor      eax, eax
0xbfdf: mov      esi, 1
0xbfe4: lea      rbp, [rip + 0x14bf6]
0xbfeb: call     0x4e60
0xbff0: jmp      0xbe78
0xbe78:
0xbff5: nop      dword ptr [rax]
0xbff8: mov      rdi, rbp
0xbffb: lea      r13, [rsp + 8]
0xc000: call     0x1d7a0
0xc005: mov      edi, 0x800
0xc00a: mov      qword ptr [rip + 0x1e93f], rax

```



```

0xc011: call    0x1d670
0xc016: mov     rdi, qword ptr [rip + 0x1de03]
0xc01d: mov     rsi, r13
0xc020: mov     qword ptr [rsp + 8], rax
0xc025: call    0x4ca0
0xc02a: xor     edx, edx
0xc02c: test    rax, rax
0xc02f: je      0xc034
0xc031: movzx   edx, byte ptr [rax]
0xc034: mov     rdi, qword ptr [rip + 0x1de25]
0xc03b: mov     rsi, r13
0xc03e: mov     byte ptr [rip + 0x1e904], dl
0xc044: call    0x4ca0
0xc049: mov     rdi, qword ptr [rip + 0x1ddf0]
0xc050: mov     rsi, r13
0xc053: mov     qword ptr [rip + 0x1e8e6], rax
0xc05a: call    0x4ca0
0xc05f: cmp     qword ptr [rip + 0x1e8d9], 0
0xc067: mov     qword ptr [rip + 0x1e8ca], rax
0xc06e: je      0xc2c0
0xc074: test    rax, rax
0xc077: je      0xc2c0
0xc07d: mov     rdi, qword ptr [rip + 0x1dcbc]
0xc084: mov     rsi, r13
0xc087: call    0x4ca0
0xc08c: mov     rdi, qword ptr [rip + 0x1dced]
0xc093: mov     rsi, r13
0xc096: mov     qword ptr [rip + 0x1dcab], rax
0xc09d: call    0x4ca0
0xc0a2: mov     rdi, qword ptr [rip + 0x1dcb7]
0xc0a9: mov     rsi, r13
0xc0ac: mov     qword ptr [rip + 0x1dcd5], rax
0xc0b3: call    0x4ca0
0xc0b8: cmp     qword ptr [rip + 0x1dc88], 0
0xc0c0: mov     qword ptr [rip + 0x1dca1], rax
0xc0c7: je      0xc352
0xc0cd: mov     rdi, qword ptr [rip + 0x1ddac]
0xc0d4: mov     rsi, r13
0xc0d7: call    0x4ca0
0xc0dc: xor     edx, edx
0xc0de: mov     qword ptr [rip + 0x1dda3], rax
0xc0e5: test    rax, rax
0xc0e8: je      0xc0ff
0xc0ea: mov     rdi, qword ptr [rip + 0x1ddcf]
0xc0f1: mov     qword ptr [rip + 0x1dc70], rax
0xc0f8: call    0x4c90
0xc0fd: mov     edx, eax
0xc0ff: mov     rdi, qword ptr [rip + 0x1dd9a]

```

```

0xc106: mov     dword ptr [rip + 0x1ddc4], edx
0xc10c: call    0x4c50
0xc111: mov     dword ptr [rip + 0x1dd99], eax
0xc117: test    eax, eax
0xc119: js      0xc13c
0xc11b: mov     qword ptr [rip + 0x1dc62], 0
0xc126: mov     qword ptr [rip + 0x1dc37], 0
0xc131: mov     qword ptr [rip + 0x1dc0c], 0
0xc13c: mov     rdi, qword ptr [rip + 0x1dd9d]
0xc143: call    0x4c50
0xc148: mov     rdi, qword ptr [rip + 0x1ddb1]
0xc14f: mov     dword ptr [rip + 0x1dd9b], eax
0xc155: call    0x4c50
0xc15a: mov     rdi, qword ptr [rip + 0x1dc3f]
0xc161: mov     rsi, r13
0xc164: mov     dword ptr [rip + 0x1dda6], eax
0xc16a: call    0x4ca0
0xc16f: mov     rdi, qword ptr [rip + 0x1dc4a]
0xc176: mov     rsi, r13
0xc179: mov     qword ptr [rip + 0x1dc28], rax
0xc180: call    0x4ca0
0xc185: cmp     qword ptr [rip + 0x1dc1b], 0
0xc18d: mov     qword ptr [rip + 0x1dc34], rax
0xc194: je      0xc2a0
0xc19a: test    rax, rax
0xc19d: je      0xc2a0
0xc1a3: mov     rdi, qword ptr [rip + 0x1dc36]
0xc1aa: mov     rsi, r13
0xc1ad: lea     rbp, [rip + 0x1ddac]
0xc1b4: lea     rbx, [rbp - 0x220]
0xc1bb: call    0x4ca0
0xc1c0: mov     rdi, qword ptr [rip + 0x1dc39]
0xc1c7: mov     rsi, r13
0xc1ca: lea     r12, [rbx + 0x4c0]
0xc1d1: mov     qword ptr [rip + 0x1dc10], rax
0xc1d8: call    0x4ca0
0xc1dd: mov     rdi, qword ptr [rip + 0x1dd3c]
0xc1e4: mov     rsi, r13
0xc1e7: mov     qword ptr [rip + 0x1dc1a], rax
0xc1ee: call    0x4ca0
0xc1f3: mov     rdi, qword ptr [rip + 0x1dd46]
0xc1fa: mov     rsi, r13
0xc1fd: mov     qword ptr [rip + 0x1dd24], rax
0xc204: call    0x4ca0
0xc209: mov     qword ptr [rip + 0x1dd38], rax
0xc210: mov     rdi, qword ptr [rbp]
0xc214: mov     rsi, r13
0xc217: add     rbp, 0x20

```

```

0xc21b: call    0x4ca0
0xc220: mov     qword ptr [rbp - 0x18], rax
0xc224: cmp     rbp, r12
0xc227: jne     0xc210
0xc229: xor     ebp, ebp
0xc22b: lea     r13, [rip + 0x161b3]
0xc232: lea     r14, [rip + 0x14c57]
0xc239: jmp     0xc249
0xc249:
0xc23b: nop     dword ptr [rax + rax]
0xc240: add     rbx, 0x20
0xc244: cmp     rbx, r12
0xc247: je      0xc268
0xc249: cmp     qword ptr [rbx + 8], 0
0xc24e: jne     0xc240
0xc250: mov     eax, dword ptr [rbx + 0x14]
0xc253: test    eax, eax
0xc255: jne     0xc2e0
0xc25b: mov     qword ptr [rbx + 8], r13
0xc25f: add     rbx, 0x20
0xc263: cmp     rbx, r12
0xc266: jne     0xc249
0xc268: test    ebp, ebp
0xc26a: jne     0xc390
0xc270: mov     rax, qword ptr [rsp + 0xa8]
0xc278: sub     rax, qword ptr fs:[0x28]
0xc281: jne     0xc3be
0xc287: add     rsp, 0xb0
0xc28e: pop     rbx
0xc28f: pop     rbp
0xc290: pop     r12
0xc292: pop     r13
0xc294: pop     r14
0xc296: ret
0xc297: nop     word ptr [rax + rax]
0xc2a0: mov     qword ptr [rip + 0x1dafd], 0
0xc2ab: mov     qword ptr [rip + 0x1db12], 0
0xc2b6: jmp     0xc1a3
0xc1a3:
0xc2bb: nop     dword ptr [rax + rax]
0xc2c0: mov     qword ptr [rip + 0x1e66d], 0
0xc2cb: mov     qword ptr [rip + 0x1e66a], 0
0xc2d6: jmp     0xc07d
0xc07d:
0xc2db: nop     dword ptr [rax + rax]
0xc2e0: mov     r8, qword ptr [rbx]
0xc2e3: mov     rcx, qword ptr [rip + 0x1e5fe]
0xc2ea: mov     rdx, r14

```

```

0xc2ed: xor     eax, eax
0xc2ef: mov     rdi, qword ptr [rip + 0x1e58a]
0xc2f6: mov     esi, 1
0xc2fb: add     ebp, 1
0xc2fe: call    0x4e60
0xc303: jmp     0xc240
0xc240
0xc308: mov     rcx, qword ptr [rip + 0x1e5d9]
0xc30f: mov     rdi, qword ptr [rip + 0x1e56a]
0xc316: lea     rdx, [rip + 0x14aab]
0xc31d: xor     eax, eax
0xc31f: mov     esi, 1
0xc324: call    0x4e60
0xc329: mov     rcx, qword ptr [rip + 0x1e550]
0xc330: mov     edx, 0x11
0xc335: mov     esi, 1
0xc33a: lea     rdi, [rip + 0x148f3]
0xc341: lea     rbp, [rip + 0x1d9d1]
0xc348: call    0x4e40
0xc34d: jmp     0xbe58
0xbe58
0xc352: mov     qword ptr [rip + 0x1da2b], 0
0xc35d: mov     qword ptr [rip + 0x1da00], 0
0xc368: jmp     0xc0cd
0xc0cd
0xc36d: mov     rcx, qword ptr [rip + 0x1e574]
0xc374: mov     rdi, qword ptr [rip + 0x1e505]
0xc37b: lea     rdx, [rip + 0x14a16]
0xc382: xor     eax, eax
0xc384: mov     esi, 1
0xc389: call    0x4e60
0xc38e: jmp     0xc329
0xc329
0xc390: mov     rcx, qword ptr [rip + 0x1e551]
0xc397: mov     rdi, qword ptr [rip + 0x1e4e2]
0xc39e: mov     r8d, ebp
0xc3a1: xor     eax, eax
0xc3a3: lea     rdx, [rip + 0x14b16]
0xc3aa: mov     esi, 1
0xc3af: call    0x4e60
0xc3b4: mov     edi, 1
0xc3b9: call    0x4e20
0xc3be: call    0x49d0
0xc3c3: nop     word ptr cs:[rax + rax]
0xc3ce: nop
0xc3d0: endbr64
0xc3d4: push    rbx
0xc3d5: mov     ebx, edi

```

```

0xc3d7: xor     edi, edi
0xc3d9: call    0x48f0
0xc3de: test    eax, eax
0xc3e0: je      0xc453
0xc3e2: mov     edi, 1
0xc3e7: call    0x48f0
0xc3ec: test    eax, eax
0xc3ee: je      0xc453
0xc3f0: mov     edi, 1
0xc3f5: call    0x4cd0
0xc3fa: mov     qword ptr [rip + 0x1e58f], rax
0xc401: test    rax, rax
0xc404: je      0xc47e
0xc406: lea     rsi, [rip + 0x1efb3]
0xc40d: mov     edi, 1
0xc412: call    0x4d60
0xc417: mov     edi, 0x40
0xc41c: call    0x1d670
0xc421: mov     dword ptr [rip + 0x1efd5], ebx
0xc427: pop     rbx
0xc428: mov     qword ptr [rip + 0x1ef31], rax
0xc42f: mov     qword ptr [rip + 0x1e52a], rax
0xc436: xor     eax, eax
0xc438: mov     qword ptr [rip + 0x1e52d], 0
0xc443: mov     qword ptr [rip + 0x1e51a], 0
0xc44e: jmp     0xbde0
0xbde0
0xc453: mov     rdi, qword ptr [rip + 0x1e426]
0xc45a: mov     rcx, qword ptr [rip + 0x1e487]
0xc461: lea     rdx, [rip + 0x14a88]
0xc468: xor     eax, eax
0xc46a: mov     esi, 1
0xc46f: call    0x4e60
0xc474: mov     edi, 1
0xc479: call    0x4e20
0xc47e: mov     rdi, qword ptr [rip + 0x1e3fb]
0xc485: mov     rcx, qword ptr [rip + 0x1e45c]
0xc48c: mov     esi, 1
0xc491: lea     rdx, [rip + 0x147f8]
0xc498: call    0x4e60
0xc49d: mov     edi, 1
0xc4a2: call    0x4e20
0xc4a7: nop     word ptr [rax + rax]
0xc4b0: endbr64
0xc4b4: push    r12
0xc4b6: push    rbp
0xc4b7: mov     rbp, rdi
0xc4ba: mov     rdi, qword ptr [rip + 0x1e48f]

```

```

0xc4c1: push    rbx
0xc4c2: call     0x9940
0xc4c7: test     eax, eax
0xc4c9: jne      0xc4d0
0xc4cb: pop      rbx
0xc4cc: pop      rbp
0xc4cd: pop      r12
0xc4cf: ret
0xc4d0: mov      rdi, rbp
0xc4d3: call     0x49b0
0xc4d8: lea      rdi, [rax + 0x81]
0xc4df: mov      rbx, rax
0xc4e2: call     0x1d670
0xc4e7: mov      rdi, rbp
0xc4ea: mov      r12, rax
0xc4ed: call     0x1d7a0
0xc4f2: mov      rsi, rbx
0xc4f5: add      ebx, 0x7f
0xc4f8: mov      rbp, rax
0xc4fb: mov      rdi, rax
0xc4fe: call     0x184d0
0xc503: push     7
0xc505: lea      rcx, [rip + 0x147a2]
0xc50c: xor      eax, eax
0xc50e: push     rbp
0xc50f: mov      rdx, -1
0xc516: mov      esi, 1
0xc51b: mov      rdi, r12
0xc51e: lea      r9, [rip + 0x14040]
0xc525: mov      r8d, 0x1b
0xc52b: call     0x4f10
0xc530: mov      eax, dword ptr [rip + 0x1e462]
0xc536: pop      rdx
0xc537: pop      rcx
0xc538: add      eax, eax
0xc53a: cmp      ebx, eax
0xc53c: jl       0xc555
0xc53e: cdqe
0xc540: mov      byte ptr [r12 + rax], 7
0xc545: mov      eax, dword ptr [rip + 0x1e44d]
0xc54b: add      eax, eax
0xc54d: cdqe
0xc54f: mov      byte ptr [r12 + rax + 1], 0
0xc555: mov      rdi, r12
0xc558: call     0x49b0
0xc55d: mov      rsi, r12
0xc560: mov      edi, 1
0xc565: mov      rdx, rax

```

```

0xc568: call    0x18da0
0xc56d: mov     rdi, rbp
0xc570: call    0x18d10
0xc575: mov     rdi, r12
0xc578: call    0x18d10
0xc57d: mov     rdi, qword ptr [rip + 0x1e2bc]
0xc584: pop     rbx
0xc585: pop     rbp
0xc586: pop     r12
0xc588: jmp     0x4c40
0x4c40
0xc58d: nop     dword ptr [rax]
0xc590: endbr64
0xc594: sub     rsp, 8
0xc598: mov     edi, 0x18
0xc59d: call    0x1d670
0xc5a2: mov     qword ptr [rax], 0
0xc5a9: mov     qword ptr [rax + 8], 0
0xc5b1: add     rsp, 8
0xc5b5: ret
0xc5b6: nop     word ptr cs:[rax + rax]
0xc5c0: endbr64
0xc5c4: test    rdi, rdi
0xc5c7: je      0xc5d0
0xc5c9: jmp     0x18d10
0x18d10
0xc5ce: nop
0xc5d0: ret
0xc5d1: nop     word ptr cs:[rax + rax]
0xc5dc: nop     dword ptr [rax]
0xc5e0: endbr64
0xc5e4: mov     dword ptr [rdi], esi
0xc5e6: mov     dword ptr [rdi + 4], edx
0xc5e9: mov     dword ptr [rdi + 8], ecx
0xc5ec: mov     dword ptr [rdi + 0xc], r8d
0xc5f0: ret
0xc5f1: nop     word ptr cs:[rax + rax]
0xc5fc: nop     dword ptr [rax]
0xc600: endbr64
0xc604: mov     rax, rdi
0xc607: mov     rdi, rsi
0xc60a: mov     esi, edx
0xc60c: mov     ecx, dword ptr [rax + 0x10]
0xc60f: lea     r8d, [rcx + rdx]
0xc613: mov     edx, dword ptr [rax + 0xc]
0xc616: mov     dword ptr [rax + 0x10], r8d
0xc61a: cmp     edx, ecx
0xc61c: jle     0xc640

```

```

0xc61e: mov     r9d, dword ptr [rax + 8]
0xc622: cmp     dword ptr [rax + 0x14], r9d
0xc626: jge     0xc640
0xc628: cmp     r8d, edx
0xc62b: jle     0xc631
0xc62d: sub     edx, ecx
0xc62f: mov     esi, edx
0xc631: jmp     0xa1a0
0xa1a0
0xc636: nop     word ptr cs:[rax + rax]
0xc640: xor     eax, eax
0xc642: ret
0xc643: nop     word ptr cs:[rax + rax]
0xc64e: nop
0xc650: endbr64
0xc654: mov     ecx, dword ptr [rdi + 0x10]
0xc657: lea     edx, [rcx + 1]
0xc65a: mov     dword ptr [rdi + 0x10], edx
0xc65d: cmp     edx, dword ptr [rdi + 0xc]
0xc660: jg      0xc66a
0xc662: mov     ecx, dword ptr [rdi + 8]
0xc665: cmp     dword ptr [rdi + 0x14], ecx
0xc668: jl      0xc670
0xc66a: xor     eax, eax
0xc66c: ret
0xc66d: nop     dword ptr [rax]
0xc670: mov     edi, esi
0xc672: jmp     0xa230
0xa230
0xc677: nop     word ptr [rax + rax]
0xc680: endbr64
0xc684: mov     rax, rdi
0xc687: mov     edi, esi
0xc689: mov     dword ptr [rax + 0x10], edx
0xc68c: add     edx, dword ptr [rax]
0xc68e: mov     dword ptr [rax + 0x14], esi
0xc691: add     edi, dword ptr [rax + 4]
0xc694: mov     esi, edx
0xc696: jmp     0xa4b0
0xa4b0
0xc69b: nop     dword ptr [rax + rax]
0xc6a0: endbr64
0xc6a4: mov     eax, dword ptr [rdi]
0xc6a6: ret
0xc6a7: nop     word ptr [rax + rax]
0xc6b0: endbr64
0xc6b4: mov     eax, dword ptr [rdi + 4]
0xc6b7: ret

```



```

0xc6b8: nop      dword ptr [rax + rax]
0xc6c0: endbr64
0xc6c4: mov      eax, dword ptr [rdi + 8]
0xc6c7: ret
0xc6c8: nop      dword ptr [rax + rax]
0xc6d0: endbr64
0xc6d4: mov      eax, dword ptr [rdi + 0xc]
0xc6d7: ret
0xc6d8: nop      dword ptr [rax + rax]
0xc6e0: push     r14
0xc6e2: push     r13
0xc6e4: push     r12
0xc6e6: mov      r12, qword ptr [rip + 0x1ed1b]
0xc6ed: push     rbp
0xc6ee: push     rbx
0xc6ef: mov      r13, qword ptr [r12 + 0x20]
0xc6f4: mov      rbx, qword ptr [r12 + 0x18]
0xc6f9: cmp      r13, rbx
0xc6fc: je       0xc761
0xc6fe: mov      ebp, edi
0xc700: jbe      0xc770
0xc702: test     bpl, 1
0xc706: je       0xc75c
0xc708: mov      rdi, qword ptr [r12 + 0x60]
0xc70d: test     rdi, rdi
0xc710: je       0xc71e
0xc712: call     0x18d10
0xc717: mov      r12, qword ptr [rip + 0x1ecea]
0xc71e: mov      r14, r13
0xc721: sub      r14, rbx
0xc724: lea      rdi, [r14 + 1]
0xc728: call     0x1d670
0xc72d: mov      rdx, r14
0xc730: mov      qword ptr [r12 + 0x60], rax
0xc735: mov      rax, qword ptr [rip + 0x1eccc]
0xc73c: mov      rsi, qword ptr [rax + 0x58]
0xc740: mov      rdi, qword ptr [rax + 0x60]
0xc744: add      rsi, rbx
0xc747: call     0x4bb0
0xc74c: mov      rax, qword ptr [rip + 0x1ecb5]
0xc753: mov      rax, qword ptr [rax + 0x60]
0xc757: mov      byte ptr [rax + r14], 0
0xc75c: and      ebp, 2
0xc75f: jne      0xc788
0xc761: pop      rbx
0xc762: pop      rbp
0xc763: pop      r12
0xc765: pop      r13

```

```

0xc767: pop      r14
0xc769: ret
0xc76a: nop      word ptr [rax + rax]
0xc770: mov      rax, rbx
0xc773: mov      rbx, r13
0xc776: mov      r13, rax
0xc779: test     bpl, 1
0xc77d: je       0xc75c
0xc77f: jmp      0xc708
0xc708:
0xc781: nop      dword ptr [rax]
0xc788: mov      rax, qword ptr [rip + 0x1ec79]
0xc78f: mov      rdx, rbx
0xc792: sub      rdx, r13
0xc795: add      qword ptr [rax + 0x48], rdx
0xc799: mov      rdi, qword ptr [rax + 0x58]
0xc79d: add      qword ptr [rax + 0x38], rdx
0xc7a1: mov      qword ptr [rax + 0x18], rbx
0xc7a5: lea      rsi, [rdi + r13]
0xc7a9: add      rdi, rbx
0xc7ac: mov      qword ptr [rax + 0x20], rbx
0xc7b0: call     0x48b0
0xc7b5: mov      rbx, qword ptr [rip + 0x1ec4c]
0xc7bc: mov      rax, qword ptr [rbx + 0x38]
0xc7c0: mov      rdi, qword ptr [rbx + 0x58]
0xc7c4: lea      rsi, [rax + 1]
0xc7c8: mov      qword ptr [rbx + 0x50], rsi
0xc7cc: call     0x1d690
0xc7d1: mov      qword ptr [rbx + 0x58], rax
0xc7d5: pop      rbx
0xc7d6: pop      rbp
0xc7d7: pop      r12
0xc7d9: pop      r13
0xc7db: pop      r14
0xc7dd: ret
0xc7de: nop
0xc7e0: endbr64
0xc7e4: test     rdi, rdi
0xc7e7: je       0xc818
0xc7e9: push     rbp
0xc7ea: mov      rbp, rdi
0xc7ed: mov      rdi, qword ptr [rdi + 0x58]
0xc7f1: test     rdi, rdi
0xc7f4: je       0xc7fb
0xc7f6: call     0x18d10
0xc7fb: mov      rdi, qword ptr [rbp + 0x60]
0xc7ff: test     rdi, rdi
0xc802: je       0xc809

```

```

0xc804: call    0x18d10
0xc809: mov     rdi, rbp
0xc80c: pop      rbp
0xc80d: jmp     0x18d10
0x18d10
0xc812: nop     word ptr [rax + rax]
0xc818: ret
0xc819: nop     dword ptr [rax]
0xc820: endbr64
0xc824: mov     rax, qword ptr [rip + 0x1ebdd]
0xc82b: push    rbp
0xc82c: mov     rdi, qword ptr [rax + 0x68]
0xc830: xor     eax, eax
0xc832: call    0x1aac0
0xc837: mov     rax, qword ptr [rip + 0x1ebca]
0xc83e: mov     rdi, qword ptr [rax]
0xc841: call    0xc5c0
0xc846: mov     rbp, qword ptr [rip + 0x1ebbb]
0xc84d: test    rbp, rbp
0xc850: je      0xc876
0xc852: mov     rdi, qword ptr [rbp + 0x58]
0xc856: test    rdi, rdi
0xc859: je      0xc860
0xc85b: call    0x18d10
0xc860: mov     rdi, qword ptr [rbp + 0x60]
0xc864: test    rdi, rdi
0xc867: je      0xc86e
0xc869: call    0x18d10
0xc86e: mov     rdi, rbp
0xc871: call    0x18d10
0xc876: mov     qword ptr [rip + 0x1eb87], 0
0xc881: pop     rbp
0xc882: ret
0xc883: nop     word ptr cs:[rax + rax]
0xc88e: nop
0xc890: endbr64
0xc894: mov     rax, qword ptr [rip + 0x1eb6d]
0xc89b: movsxd  rcx, edi
0xc89e: mov     edx, esi
0xc8a0: mov     r8, rcx
0xc8a3: mov     qword ptr [rax + 0x28], rcx
0xc8a7: movsxd  rcx, esi
0xc8aa: mov     rdi, qword ptr [rax]
0xc8ad: xor     esi, esi
0xc8af: mov     qword ptr [rax + 0x30], rcx
0xc8b3: mov     ecx, 1
0xc8b8: jmp     0xc5e0
0xc5e0

```

```

0xc8bd: nop      dword ptr [rax]
0xc8c0: endbr64
0xc8c4: push     r12
0xc8c6: mov      edi, 0x70
0xc8cb: push     rbx
0xc8cc: sub      rsp, 8
0xc8d0: call     0x1d670
0xc8d5: mov      rbx, qword ptr [rip + 0x1eb2c]
0xc8dc: mov      r12, rax
0xc8df: movdqu   xmm0, xmmword ptr [rbx]
0xc8e3: movups   xmmword ptr [rax], xmm0
0xc8e6: movdqu   xmm1, xmmword ptr [rbx + 0x10]
0xc8eb: movups   xmmword ptr [rax + 0x10], xmm1
0xc8ef: movdqu   xmm2, xmmword ptr [rbx + 0x20]
0xc8f4: movups   xmmword ptr [rax + 0x20], xmm2
0xc8f8: movdqu   xmm3, xmmword ptr [rbx + 0x30]
0xc8fd: movups   xmmword ptr [rax + 0x30], xmm3
0xc901: movdqu   xmm4, xmmword ptr [rbx + 0x40]
0xc906: movups   xmmword ptr [rax + 0x40], xmm4
0xc90a: movdqu   xmm5, xmmword ptr [rbx + 0x50]
0xc90f: movups   xmmword ptr [rax + 0x50], xmm5
0xc913: movdqu   xmm6, xmmword ptr [rbx + 0x60]
0xc918: cmp      qword ptr [rax + 0x58], 0
0xc91d: movups   xmmword ptr [rax + 0x60], xmm6
0xc921: je       0xc94a
0xc923: mov      rdi, qword ptr [rax + 0x50]
0xc927: call     0x1d670
0xc92c: mov      rdx, qword ptr [r12 + 0x50]
0xc931: mov      qword ptr [rbx + 0x58], rax
0xc935: mov      rax, qword ptr [rip + 0x1eacc]
0xc93c: mov      rsi, qword ptr [r12 + 0x58]
0xc941: mov      rdi, qword ptr [rax + 0x58]
0xc945: call     0x4bb0
0xc94a: mov      rdi, qword ptr [r12 + 0x60]
0xc94f: test     rdi, rdi
0xc952: je       0xc964
0xc954: mov      rbx, qword ptr [rip + 0x1eaad]
0xc95b: call     0x1d7a0
0xc960: mov      qword ptr [rbx + 0x60], rax
0xc964: add      rsp, 8
0xc968: mov      rax, r12
0xc96b: pop      rbx
0xc96c: pop      r12
0xc96e: ret
0xc96f: nop
0xc970: endbr64
0xc974: test     rdi, rdi
0xc977: je       0xc9f0

```

```

0xc979: push    r13
0xc97b: push    r12
0xc97d: push    rbp
0xc97e: push    rbx
0xc97f: mov     rbx, rdi
0xc982: sub     rsp, 8
0xc986: mov     rbp, qword ptr [rip + 0x1ea7b]
0xc98d: mov     rdi, qword ptr [rbp + 0x58]
0xc991: mov     r13, qword ptr [rbp + 0x28]
0xc995: mov     r12, qword ptr [rbp + 0x30]
0xc999: test    rdi, rdi
0xc99c: je      0xc9a3
0xc99e: call    0x18d10
0xc9a3: mov     rdi, qword ptr [rbp + 0x60]
0xc9a7: test    rdi, rdi
0xc9aa: je      0xc9b1
0xc9ac: call    0x18d10
0xc9b1: mov     rdi, rbp
0xc9b4: call    0x18d10
0xc9b9: movsxd  rax, r13d
0xc9bc: mov     edx, r12d
0xc9bf: movsxd  r12, r12d
0xc9c2: mov     qword ptr [rbx + 0x28], rax
0xc9c6: mov     rdi, qword ptr [rbx]
0xc9c9: mov     r8d, r13d
0xc9cc: mov     ecx, 1
0xc9d1: mov     qword ptr [rbx + 0x30], r12
0xc9d5: xor     esi, esi
0xc9d7: mov     qword ptr [rip + 0x1ea2a], rbx
0xc9de: add     rsp, 8
0xc9e2: pop     rbx
0xc9e3: pop     rbp
0xc9e4: pop     r12
0xc9e6: pop     r13
0xc9e8: jmp     0xc5e0
0xc5e0
0xc9ed: nop     dword ptr [rax]
0xc9f0: ret
0xc9f1: nop     word ptr cs:[rax + rax]
0xc9fc: nop     dword ptr [rax]
0xca00: endbr64
0xca04: mov     rdx, qword ptr [rip + 0x1e9fd]
0xca0b: mov     rax, qword ptr [rdx + 0x18]
0xca0f: sub     rax, qword ptr [rdx + 0x40]
0xca13: ret
0xca14: nop     word ptr cs:[rax + rax]
0xca1f: nop
0xca20: endbr64

```

```

0xca24: mov     rax, qword ptr [rip + 0x1e9dd]
0xca2b: mov     r8d, dword ptr [rax + 8]
0xca2f: mov     dword ptr [rax + 8], edi
0xca32: mov     eax, r8d
0xca35: ret
0xca36: nop     word ptr cs:[rax + rax]
0xca40: endbr64
0xca44: mov     rax, qword ptr [rip + 0x1e9bd]
0xca4b: cmp     qword ptr [rax + 0x48], 0
0xca50: sete    al
0xca53: movzx   eax, al
0xca56: ret
0xca57: nop     word ptr [rax + rax]
0xca60: endbr64
0xca64: mov     rax, qword ptr [rip + 0x1e99d]
0xca6b: mov     rdx, qword ptr [rax + 0x18]
0xca6f: mov     qword ptr [rax + 0x20], rdx
0xca73: ret
0xca74: nop     word ptr cs:[rax + rax]
0xca7f: nop
0xca80: endbr64
0xca84: sub     rsp, 8
0xca88: xor     eax, eax
0xca8a: mov     edi, 3
0xca8f: call    0xc6e0
0xca94: mov     rax, qword ptr [rip + 0x1e96d]
0xca9b: mov     dword ptr [rax + 0x10], 0x18
0xcaa2: add     rsp, 8
0xcaa6: ret
0xcaa7: nop     word ptr [rax + rax]
0xcab0: endbr64
0xcab4: sub     rsp, 8
0xcab8: xor     eax, eax
0xcaba: mov     edi, 1
0xcabf: call    0xc6e0
0xcac4: mov     rax, qword ptr [rip + 0x1e93d]
0xcacb: mov     dword ptr [rax + 0x10], 0x19
0xcad2: add     rsp, 8
0xcad6: ret
0xcad7: nop     word ptr [rax + rax]
0xcae0: endbr64
0xcae4: mov     rax, qword ptr [rip + 0x1e91d]
0xcaeb: mov     rdx, qword ptr [rax + 0x18]
0xcaef: mov     rcx, qword ptr [rax + 0x20]
0xcaf3: movsxd  rdx, edx
0xcaf6: mov     qword ptr [rax + 0x18], rcx
0xcafa: mov     qword ptr [rax + 0x20], rdx
0xcafe: ret

```

```

0xcaff: nop
0xcb00: endbr64
0xcb04: mov     rax, qword ptr [rip + 0x1e8fd]
0xcb0b: mov     rdx, qword ptr [rax + 0x18]
0xcb0f: cmp     rdx, qword ptr [rax + 0x40]
0xcb13: jbe     0xcb24
0xcb15: sub     rdx, 1
0xcb19: mov     dword ptr [rax + 0x10], 1
0xcb20: mov     qword ptr [rax + 0x18], rdx
0xcb24: ret
0xcb25: nop     word ptr cs:[rax + rax]
0xcb30: endbr64
0xcb34: mov     rax, qword ptr [rip + 0x1e8cd]
0xcb3b: mov     rdx, qword ptr [rax + 0x18]
0xcb3f: cmp     rdx, qword ptr [rax + 0x38]
0xcb43: jae     0xcb54
0xcb45: add     rdx, 1
0xcb49: mov     dword ptr [rax + 0x10], 2
0xcb50: mov     qword ptr [rax + 0x18], rdx
0xcb54: ret
0xcb55: nop     word ptr cs:[rax + rax]
0xcb60: endbr64
0xcb64: mov     rdx, qword ptr [rip + 0x1e89d]
0xcb6b: mov     rax, qword ptr [rdx + 0x18]
0xcb6f: mov     rdi, qword ptr [rdx + 0x40]
0xcb73: cmp     rax, rdi
0xcb76: jbe     0xcc20
0xcb7c: mov     r8, qword ptr [rdx + 0x58]
0xcb80: jmp     0xcba9
0xcba9:
0xcb82: nop     word ptr [rax + rax]
0xcb88: lea     ecx, [rsi - 0x30]
0xcb8b: cmp     cl, 9
0xcb8e: jbe     0xcbbf
0xcb90: sub     esi, 0x24
0xcb93: cmp     sil, 1
0xcb97: jbe     0xcbbf
0xcb99: mov     qword ptr [rdx + 0x18], rax
0xcb9d: mov     dword ptr [rdx + 0x10], 1
0xcba4: cmp     rdi, rax
0xcba7: jae     0xcbef
0xcba9: sub     rax, 1
0xcbad: movzx   esi, byte ptr [r8 + rax]
0xcbb2: mov     ecx, esi
0xcbb4: and     ecx, 0xffffffff
0xcbb7: sub     ecx, 0x41
0xcbb8: cmp     cl, 0x19
0xcbbd: ja     0xcb88

```

```

0xcbbf: mov     rax, qword ptr [rdx + 0x18]
0xcbc3: cmp     rdi, rax
0xcbc6: jae     0xcbef
0xcbc8: sub     rax, 1
0xcbcc: movzx   esi, byte ptr [r8 + rax]
0xcbd1: mov     ecx, esi
0xcbd3: and     ecx, 0xffffffff
0xcbd6: sub     ecx, 0x41
0xcbd9: cmp     cl, 0x19
0xcbd9: jbe     0xcc00
0xcbde: lea     ecx, [rsi - 0x30]
0xcbe1: cmp     cl, 9
0xcbe4: jbe     0xcc00
0xcbe6: sub     esi, 0x24
0xcbe9: cmp     sil, 1
0xcbed: jbe     0xcc00
0xcbef: mov     dword ptr [rdx + 0x10], 3
0xcbf6: ret
0xcbf7: nop     word ptr [rax + rax]
0xcc00: mov     qword ptr [rdx + 0x18], rax
0xcc04: mov     dword ptr [rdx + 0x10], 1
0xcc0b: cmp     rdi, rax
0xcc0e: jb      0xcbc8
0xcc10: mov     dword ptr [rdx + 0x10], 3
0xcc17: ret
0xcc18: nop     dword ptr [rax + rax]
0xcc20: ret
0xcc21: nop     word ptr cs:[rax + rax]
0xcc2c: nop     dword ptr [rax]
0xcc30: endbr64
0xcc34: mov     rdx, qword ptr [rip + 0x1e7cd]
0xcc3b: mov     rax, qword ptr [rdx + 0x18]
0xcc3f: mov     rdi, qword ptr [rdx + 0x38]
0xcc43: cmp     rax, rdi
0xcc46: jae     0xcce8
0xcc4c: mov     r8, qword ptr [rdx + 0x58]
0xcc50: jmp     0xcc7d
0xcc52: nop     word ptr [rax + rax]
0xcc58: lea     ecx, [rsi - 0x30]
0xcc5b: cmp     cl, 9
0xcc5e: jbe     0xcc8f
0xcc60: sub     esi, 0x24
0xcc63: cmp     sil, 1
0xcc67: jbe     0xcc8f
0xcc69: add     rax, 1
0xcc6d: mov     dword ptr [rdx + 0x10], 2
0xcc74: mov     qword ptr [rdx + 0x18], rax

```



```

0xcc78: cmp     rdi, rax
0xcc7b: je      0xccbb
0xcc7d: movzx   esi, byte ptr [r8 + rax]
0xcc82: mov     ecx, esi
0xcc84: and     ecx, 0xffffffff
0xcc87: sub     ecx, 0x41
0xcc8a: cmp     cl, 0x19
0xcc8d: ja      0xcc58
0xcc8f: mov     rax, qword ptr [rdx + 0x18]
0xcc93: cmp     rdi, rax
0xcc96: jbe     0xccbb
0xcc98: movzx   esi, byte ptr [r8 + rax]
0xcc9d: mov     ecx, esi
0xcc9f: and     ecx, 0xffffffff
0xcca2: sub     ecx, 0x41
0xcca5: cmp     cl, 0x19
0xcca8: jbe     0xcc88
0xccaa: lea     ecx, [rsi - 0x30]
0ccad: cmp     cl, 9
0ccb0: jbe     0xcc88
0ccb2: sub     esi, 0x24
0ccb5: cmp     sil, 1
0ccb9: jbe     0xcc88
0ccbb: mov     dword ptr [rdx + 0x10], 4
0ccc2: ret
0ccc3: nop     dword ptr [rax + rax]
0ccc8: add     rax, 1
0cccc: mov     dword ptr [rdx + 0x10], 2
0ccd3: mov     qword ptr [rdx + 0x18], rax
0ccd7: cmp     rdi, rax
0ccda: jne     0xcc98
0ccdc: mov     dword ptr [rdx + 0x10], 4
0xcce3: ret
0xcce4: nop     dword ptr [rax]
0xcce8: ret
0xcce9: nop     dword ptr [rax]
0ccf0: endbr64
0ccf4: mov     rax, qword ptr [rip + 0x1e70d]
0ccfb: mov     rdx, qword ptr [rax + 0x40]
0ccff: mov     dword ptr [rax + 0x10], 5
0xcd06: mov     qword ptr [rax + 0x18], rdx
0xcd0a: ret
0xcd0b: nop     dword ptr [rax + rax]
0xcd10: endbr64
0xcd14: mov     rax, qword ptr [rip + 0x1e6ed]
0xcd1b: mov     rdx, qword ptr [rax + 0x38]
0xcd1f: mov     dword ptr [rax + 0x10], 6
0xcd26: mov     qword ptr [rax + 0x18], rdx

```

```

0xcd2a: ret
0xcd2b: nop      dword ptr [rax + rax]
0xcd30: endbr64
0xcd34: push     rbp
0xcd35: push     rbx
0xcd36: mov      ebx, edi
0xcd38: sub      rsp, 8
0xcd3c: call     0x4f00
0xcd41: movsxd   rdx, ebx
0xcd44: mov      rax, qword ptr [rax]
0xcd47: test     byte ptr [rax + rdx*2 + 1], 0x40
0xcd4c: je       0xcdab
0xcd4e: mov      rbp, qword ptr [rip + 0x1e6b3]
0xcd55: mov      rsi, qword ptr [rbp + 0x38]
0xcd59: lea      rdx, [rsi + 1]
0xcd5d: cmp      rdx, qword ptr [rbp + 0x50]
0xcd61: jae      0xcdb8
0xcd63: mov      rax, qword ptr [rbp + 0x18]
0xcd67: mov      rcx, qword ptr [rbp + 0x58]
0xcd6b: sub      rdx, rax
0xcd6e: lea      rsi, [rcx + rax]
0xcd72: lea      rdi, [rcx + rax + 1]
0xcd77: call     0x4d40
0xcd7c: mov      rax, qword ptr [rip + 0x1e685]
0xcd83: mov      rdx, qword ptr [rax + 0x58]
0xcd87: mov      rax, qword ptr [rax + 0x18]
0xcd8b: mov      byte ptr [rdx + rax], bl
0xcd8e: mov      rax, qword ptr [rip + 0x1e673]
0xcd95: add      qword ptr [rax + 0x18], 1
0xcd9a: add      qword ptr [rax + 0x38], 1
0xcd9f: add      qword ptr [rax + 0x48], 1
0xcda4: mov      dword ptr [rax + 0x10], 7
0xcdab: add      rsp, 8
0xcdaf: pop      rbx
0xcdb0: pop      rbp
0xcdb1: ret
0xcdb2: nop      word ptr [rax + rax]
0xcdb8: add      rsi, 0x21
0xcdbc: mov      rdi, qword ptr [rbp + 0x58]
0xcdc0: mov      qword ptr [rbp + 0x50], rsi
0xcdc4: call     0x1d690
0xcdc9: mov      qword ptr [rbp + 0x58], rax
0xcdcd: mov      rbp, qword ptr [rip + 0x1e634]
0xcdd4: mov      rax, qword ptr [rbp + 0x38]
0xcdd8: lea      rdx, [rax + 1]
0xcddc: jmp      0xcd63
0xcd63
0xcdde: nop

```

```

0xcde0: endbr64
0xcde4: mov     rax, qword ptr [rip + 0x1e61d]
0xcdeb: mov     rcx, qword ptr [rax + 0x18]
0xcdef: mov     rdx, qword ptr [rax + 0x38]
0xcdf3: cmp     rcx, rdx
0xcdf6: jb      0xce00
0xcdf8: ret
0xcdf9: nop     dword ptr [rax]
0xce00: push    rbx
0xce01: mov     rax, qword ptr [rax + 0x58]
0xce05: add     rdx, 1
0xce09: sub     rdx, rcx
0xce0c: lea     rsi, [rax + rcx + 1]
0xce11: lea     rdi, [rax + rcx]
0xce15: call    0x4bb0
0xce1a: mov     rbx, qword ptr [rip + 0x1e5e7]
0xce21: mov     rsi, qword ptr [rbx + 0x38]
0xce25: sub     qword ptr [rbx + 0x48], 1
0xce2a: lea     rax, [rsi - 1]
0xce2e: mov     qword ptr [rbx + 0x38], rax
0xce32: test    al, 0xf
0xce34: je      0xce40
0xce36: mov     dword ptr [rbx + 0x10], 8
0xce3d: pop     rbx
0xce3e: ret
0xce3f: nop
0xce40: mov     qword ptr [rbx + 0x50], rsi
0xce44: mov     rdi, qword ptr [rbx + 0x58]
0xce48: call    0x1d690
0xce4d: mov     qword ptr [rbx + 0x58], rax
0xce51: mov     rbx, qword ptr [rip + 0x1e5b0]
0xce58: mov     dword ptr [rbx + 0x10], 8
0xce5f: pop     rbx
0xce60: ret
0xce61: nop     word ptr cs:[rax + rax]
0xce6c: nop     dword ptr [rax]
0xce70: endbr64
0xce74: mov     rax, qword ptr [rip + 0x1e58d]
0xce7b: mov     rcx, qword ptr [rax + 0x18]
0xce7f: cmp     rcx, qword ptr [rax + 0x40]
0xce83: ja      0xce90
0xce85: ret
0xce86: nop     word ptr cs:[rax + rax]
0xce90: push    rbx
0xce91: mov     rdx, qword ptr [rax + 0x38]
0xce95: mov     rsi, qword ptr [rax + 0x58]
0xce99: add     rdx, 1
0xce9d: lea     rdi, [rsi + rcx - 1]

```

```

0xcea2: sub     rdx, rcx
0xcea5: add     rsi, rcx
0xcea8: call    0x4bb0
0xcead: mov     rbx, qword ptr [rip + 0x1e554]
0xceb4: mov     rsi, qword ptr [rbx + 0x38]
0xceb8: sub     qword ptr [rbx + 0x18], 1
0xcebd: sub     qword ptr [rbx + 0x48], 1
0xcec2: lea     rax, [rsi - 1]
0xcec6: mov     qword ptr [rbx + 0x38], rax
0xceca: test    al, 0xf
0xcecc: je      0xcee0
0xcece: mov     dword ptr [rbx + 0x10], 9
0xcd5: pop     rbx
0xcd6: ret
0xcd7: nop     word ptr [rax + rax]
0xcee0: mov     qword ptr [rbx + 0x50], rsi
0xcee4: mov     rdi, qword ptr [rbx + 0x58]
0xcee8: call    0x1d690
0ceed: mov     qword ptr [rbx + 0x58], rax
0xcef1: mov     rbx, qword ptr [rip + 0x1e510]
0xcef8: mov     dword ptr [rbx + 0x10], 9
0xceff: pop     rbx
0xcf00: ret
0xcf01: nop     word ptr cs:[rax + rax]
0xcf0c: nop     dword ptr [rax]
0xcf10: endbr64
0xcf14: push    rbp
0xcf15: push    rbx
0xcf16: sub     rsp, 8
0xcf1a: mov     r9, qword ptr [rip + 0x1e4e7]
0xcf21: mov     rax, qword ptr [r9 + 0x18]
0xcf25: mov     rbx, qword ptr [r9 + 0x20]
0xcf29: mov     qword ptr [r9 + 0x20], rax
0xcf2d: xor     eax, eax
0xcf2f: call    0xcc30
0xcf34: xor     eax, eax
0xcf36: mov     edi, 3
0xcf3b: mov     rbp, qword ptr [r9 + 0x18]
0xcf3f: call    0xc6e0
0xcf44: mov     rax, qword ptr [rip + 0x1e4bd]
0xcf4b: mov     rdx, qword ptr [rax + 0x18]
0xcf4f: cmp     rdx, rbx
0xcf52: jae     0xcf65
0xcf54: lea     rcx, [rdx + rbx]
0xcf58: sub     rcx, rbp
0xcf5b: cmp     rbx, rbp
0xcf5e: cmovae  rdx, rcx
0xcf62: mov     rbx, rdx

```

```

0xc65: mov     qword ptr [rax + 0x20], rbx
0xc69: mov     dword ptr [rax + 0x10], 0xa
0xc70: add     rsp, 8
0xc74: pop     rbx
0xc75: pop     rbp
0xc76: ret
0xc77: nop     word ptr [rax + rax]
0xc80: endbr64
0xc84: push    rbp
0xc85: push    rbx
0xc86: sub     rsp, 8
0xc8a: mov     rax, qword ptr [rip + 0x1e477]
0xc91: mov     rbp, qword ptr [rax + 0x18]
0xc95: mov     rbx, qword ptr [rax + 0x20]
0xc99: mov     qword ptr [rax + 0x20], rbp
0xc9d: xor     eax, eax
0xc9f: call    0xcb60
0xcfa4: xor     eax, eax
0xcfa6: mov     edi, 3
0xcfab: call    0xc6e0
0xcfb0: mov     rax, qword ptr [rip + 0x1e451]
0xcfb7: mov     rdx, qword ptr [rax + 0x18]
0xcbb: cmp     rdx, rbx
0xcbe: jae     0xcd1
0xcfc0: lea     rcx, [rdx + rbx]
0xcfc4: sub     rcx, rbp
0xcfc7: cmp     rbx, rbp
0xcfc9: cmovae rdx, rcx
0xcfce: mov     rbx, rdx
0xcd1: mov     qword ptr [rax + 0x20], rbx
0xcd5: mov     dword ptr [rax + 0x10], 0xb
0xcdc: add     rsp, 8
0xcfe0: pop     rbx
0xcfe1: pop     rbp
0xcfe2: ret
0xcfe3: nop     word ptr cs:[rax + rax]
0xcfee: nop
0xcff0: endbr64
0xcff4: push    rbx
0xcff5: mov     rbx, qword ptr [rip + 0x1e40c]
0xcffc: mov     esi, 1
0xd001: mov     qword ptr [rbx + 0x18], 0
0xd009: mov     rdi, qword ptr [rbx + 0x58]
0xd00d: mov     qword ptr [rbx + 0x20], 0
0xd015: mov     qword ptr [rbx + 0x38], 0
0xd01d: mov     qword ptr [rbx + 0x40], 0
0xd025: mov     qword ptr [rbx + 0x48], 0
0xd02d: mov     qword ptr [rbx + 0x50], 1

```

```

0xd035: call    0x1d690
0xd03a: mov     qword ptr [rbx + 0x58], rax
0xd03e: mov     rax, qword ptr [rip + 0x1e3c3]
0xd045: mov     rax, qword ptr [rax + 0x58]
0xd049: mov     byte ptr [rax], 0
0xd04c: mov     rax, qword ptr [rip + 0x1e3b5]
0xd053: mov     dword ptr [rax + 0x10], 0xc
0xd05a: pop     rbx
0xd05b: ret
0xd05c: nop     dword ptr [rax]
0xd060: endbr64
0xd064: push    rbx
0xd065: mov     edi, 0x70
0xd06a: sub     rsp, 0x10
0xd06e: mov     rax, qword ptr fs:[0x28]
0xd077: mov     qword ptr [rsp + 8], rax
0xd07c: xor     eax, eax
0xd07e: call    0x1d670
0xd083: mov     qword ptr [rax + 8], 1
0xd08b: mov     qword ptr [rax + 0x58], 0
0xd093: mov     qword ptr [rax + 0x60], 0
0xd09b: mov     qword ptr [rax + 0x28], 0
0xd0a3: mov     qword ptr [rax + 0x30], 0
0xd0ab: mov     qword ptr [rip + 0x1e356], rax
0xd0b2: xor     eax, eax
0xd0b4: call    0xcff0
0xd0b9: mov     rbx, qword ptr [rip + 0x1e348]
0xd0c0: xor     eax, eax
0xd0c2: call    0xc590
0xd0c7: lea     rdi, [rip + 0x135e4]
0xd0ce: mov     qword ptr [rbx], rax
0xd0d1: call    0x17c50
0xd0d6: mov     ecx, 1
0xd0db: mov     rsi, rsp
0xd0de: mov     edx, 1
0xd0e3: lea     rdi, [rip + 0x13e42]
0xd0ea: call    0x17320
0xd0ef: mov     rdi, qword ptr [rsp]
0xd0f3: lea     rax, [rip + 0x13e24]
0xd0fa: mov     rbx, qword ptr [rip + 0x1e307]
0xd101: test    rdi, rdi
0xd104: cmove   rdi, rax
0xd108: xor     eax, eax
0xd10a: call    0x19940
0xd10f: lea     rdi, [rip + 0x1cb32]
0xd116: mov     qword ptr [rbx + 0x68], rax
0xd11a: mov     eax, dword ptr [rip + 0x1cb40]
0xd120: test    eax, eax

```

```

0xd122: lea    rax, [rip + 0x1cb07]
0xd129: cmov     rdi, rax
0xd12d: call     0x17c50
0xd132: lea     rdi, [rip + 0x1d0c7]
0xd139: mov      edx, 6
0xd13e: lea     rsi, [rip + 0x1d0db]
0xd145: call     0x17f70
0xd14a: xor      eax, eax
0xd14c: call     0x1a310
0xd151: mov      rax, qword ptr [rip + 0x1e2b0]
0xd158: mov      rdi, qword ptr [rax + 0x68]
0xd15c: xor      eax, eax
0xd15e: call     0x1a900
0xd163: nop      dword ptr [rax + rax]
0xd168: xor      eax, eax
0xd16a: call     0x1ab40
0xd16f: test     rax, rax
0xd172: jne      0xd168
0xd174: mov      edi, 0x200
0xd179: call     0x1a620
0xd17e: mov      rax, qword ptr [rsp + 8]
0xd183: sub      rax, qword ptr fs:[0x28]
0xd18c: jne      0xd194
0xd18e: add      rsp, 0x10
0xd192: pop      rbx
0xd193: ret
0xd194: call     0x49d0
0xd199: nop      dword ptr [rax]
0xd1a0: endbr64
0xd1a4: sub      rsp, 8
0xd1a8: mov      rax, qword ptr [rip + 0x1e259]
0xd1af: or       edi, 2
0xd1b2: mov      rdx, qword ptr [rax + 0x40]
0xd1b6: mov      dword ptr [rax + 0x10], 6
0xd1bd: mov      qword ptr [rax + 0x20], rdx
0xd1c1: mov      rdx, qword ptr [rax + 0x38]
0xd1c5: mov      qword ptr [rax + 0x18], rdx
0xd1c9: xor      eax, eax
0xd1cb: call     0xc6e0
0xd1d0: mov      rax, qword ptr [rip + 0x1e231]
0xd1d7: mov      rdx, qword ptr [rax + 0x18]
0xd1db: mov      dword ptr [rax + 0x10], 0xd
0xd1e2: mov      qword ptr [rax + 0x20], rdx
0xd1e6: add      rsp, 8
0xd1ea: ret
0xd1eb: nop      dword ptr [rax + rax]
0xd1f0: endbr64
0xd1f4: mov      rax, qword ptr [rip + 0x1e20d]

```

```

0xd1fb: push    rbx
0xd1fc: mov     edi, 3
0xd201: mov     rdx, qword ptr [rax + 0x18]
0xd205: mov     rsi, qword ptr [rax + 0x20]
0xd209: mov     dword ptr [rax + 0x10], 5
0xd210: mov     rcx, qword ptr [rax + 0x40]
0xd214: mov     qword ptr [rax + 0x20], rdx
0xd218: mov     rbx, rsi
0xd21b: mov     qword ptr [rax + 0x18], rcx
0xd21f: sub     rbx, rdx
0xd222: cmp     rsi, rdx
0xd225: cmovbe  rbx, rcx
0xd229: xor     eax, eax
0xd22b: call    0xc6e0
0xd230: mov     rdx, qword ptr [rip + 0x1e1d1]
0xd237: mov     rax, qword ptr [rdx + 0x38]
0xd23b: mov     dword ptr [rdx + 0x10], 0xe
0xd242: cmp     rax, rbx
0xd245: cmova   rax, rbx
0xd249: mov     qword ptr [rdx + 0x20], rax
0xd24d: pop     rbx
0xd24e: ret
0xd24f: nop
0xd250: endbr64
0xd254: mov     rax, qword ptr [rip + 0x1e1ad]
0xd25b: push    rbx
0xd25c: mov     edi, 3
0xd261: mov     rdx, qword ptr [rax + 0x18]
0xd265: mov     rbx, qword ptr [rax + 0x20]
0xd269: mov     dword ptr [rax + 0x10], 6
0xd270: mov     qword ptr [rax + 0x20], rdx
0xd274: mov     rdx, qword ptr [rax + 0x38]
0xd278: mov     qword ptr [rax + 0x18], rdx
0xd27c: xor     eax, eax
0xd27e: call    0xc6e0
0xd283: mov     rdx, qword ptr [rip + 0x1e17e]
0xd28a: mov     rax, qword ptr [rdx + 0x38]
0xd28e: mov     dword ptr [rdx + 0x10], 0xf
0xd295: cmp     rax, rbx
0xd298: cmova   rax, rbx
0xd29c: mov     qword ptr [rdx + 0x20], rax
0xd2a0: pop     rbx
0xd2a1: ret
0xd2a2: nop     word ptr cs:[rax + rax]
0xd2ad: nop     dword ptr [rax]
0xd2b0: endbr64
0xd2b4: mov     rax, qword ptr [rip + 0x1e14d]
0xd2bb: mov     rdx, qword ptr [rax + 0x58]

```



```

0xd2bf: mov     rax, qword ptr [rax + 0x18]
0xd2c3: cmp     byte ptr [rdx + rax], 0x20
0xd2c7: je      0xd2d0
0xd2c9: ret
0xd2ca: nop     word ptr [rax + rax]
0xd2d0: sub     rsp, 8
0xd2d4: nop     dword ptr [rax]
0xd2d8: xor     eax, eax
0xd2da: call    0xcde0
0xd2df: mov     rax, qword ptr [rip + 0x1e122]
0xd2e6: mov     rdx, qword ptr [rax + 0x58]
0xd2ea: mov     rcx, qword ptr [rax + 0x18]
0xd2ee: cmp     byte ptr [rdx + rcx], 0x20
0xd2f2: je      0xd2d8
0xd2f4: cmp     qword ptr [rax + 0x48], 0
0xd2f9: jne     0xd31d
0xd2fb: jmp     0xd324
0xd324:
0xd2fd: nop     dword ptr [rax]
0xd300: xor     eax, eax
0xd302: call    0xce70
0xd307: mov     rax, qword ptr [rip + 0x1e0fa]
0xd30e: cmp     qword ptr [rax + 0x48], 0
0xd313: je      0xd324
0xd315: mov     rdx, qword ptr [rax + 0x58]
0xd319: mov     rcx, qword ptr [rax + 0x18]
0xd31d: cmp     byte ptr [rdx + rcx - 1], 0x20
0xd322: je      0xd300
0xd324: mov     dword ptr [rax + 0x10], 0x11
0xd32b: add     rsp, 8
0xd32f: ret
0xd330: endbr64
0xd334: mov     rax, qword ptr [rip + 0x1e0cd]
0xd33b: mov     rdx, qword ptr [rax + 0x58]
0xd33f: mov     rax, qword ptr [rax + 0x18]
0xd343: cmp     byte ptr [rdx + rax], 0x20
0xd347: je      0xd350
0xd349: ret
0xd34a: nop     word ptr [rax + rax]
0xd350: sub     rsp, 8
0xd354: xor     eax, eax
0xd356: call    0xd2b0
0xd35b: mov     edi, 0x20
0xd360: call    0xcd30
0xd365: mov     rax, qword ptr [rip + 0x1e09c]
0xd36c: mov     dword ptr [rax + 0x10], 0x10
0xd373: add     rsp, 8
0xd377: ret

```

```

0xd378: nop      dword ptr [rax + rax]
0xd380: endbr64
0xd384: push     rbp
0xd385: push     rbx
0xd386: sub      rsp, 8
0xd38a: mov      rbp, qword ptr [rip + 0x1e077]
0xd391: mov      rbx, qword ptr [rbp + 0x18]
0xd395: cmp      rbx, qword ptr [rbp + 0x38]
0xd399: jb       0xd3a8
0xd39b: add      rsp, 8
0xd39f: pop      rbx
0xd3a0: pop      rbp
0xd3a1: ret
0xd3a2: nop      word ptr [rax + rax]
0xd3a8: xor      eax, eax
0xd3aa: call     0xcc30
0xd3af: cmp      rbx, qword ptr [rbp + 0x18]
0xd3b3: jae      0xd3e4
0xd3b5: call     0x4ef0
0xd3ba: nop      word ptr [rax + rax]
0xd3c0: mov      rdx, qword ptr [rbp + 0x58]
0xd3c4: mov      rcx, qword ptr [rax]
0xd3c7: add      rdx, rbx
0xd3ca: add      rbx, 1
0xd3ce: movsx    rsi, byte ptr [rdx]
0xd3d2: mov      ecx, dword ptr [rcx + rsi*4]
0xd3d5: mov      byte ptr [rdx], cl
0xd3d7: mov      rbp, qword ptr [rip + 0x1e02a]
0xd3de: cmp      qword ptr [rbp + 0x18], rbx
0xd3e2: ja       0xd3c0
0xd3e4: mov      dword ptr [rbp + 0x10], 0x12
0xd3eb: add      rsp, 8
0xd3ef: pop      rbx
0xd3f0: pop      rbp
0xd3f1: ret
0xd3f2: nop      word ptr cs:[rax + rax]
0xd3fd: nop      dword ptr [rax]
0xd400: endbr64
0xd404: push     rbp
0xd405: push     rbx
0xd406: sub      rsp, 8
0xd40a: mov      rbp, qword ptr [rip + 0x1dff7]
0xd411: mov      rbx, qword ptr [rbp + 0x18]
0xd415: cmp      rbx, qword ptr [rbp + 0x38]
0xd419: jb       0xd428
0xd41b: add      rsp, 8
0xd41f: pop      rbx
0xd420: pop      rbp

```

```

0xd421: ret
0xd422: nop      word ptr [rax + rax]
0xd428: xor      eax, eax
0xd42a: call     0xcc30
0xd42f: cmp      rbx, qword ptr [rbp + 0x18]
0xd433: jae      0xd464
0xd435: call     0x47c0
0xd43a: nop      word ptr [rax + rax]
0xd440: mov      rdx, qword ptr [rbp + 0x58]
0xd444: mov      rcx, qword ptr [rax]
0xd447: add      rdx, rbx
0xd44a: add      rbx, 1
0xd44e: movsx    rsi, byte ptr [rdx]
0xd452: mov      ecx, dword ptr [rcx + rsi*4]
0xd455: mov      byte ptr [rdx], cl
0xd457: mov      rbp, qword ptr [rip + 0x1dfaa]
0xd45e: cmp      qword ptr [rbp + 0x18], rbx
0xd462: ja       0xd440
0xd464: mov      dword ptr [rbp + 0x10], 0x13
0xd46b: add      rsp, 8
0xd46f: pop      rbx
0xd470: pop      rbp
0xd471: ret
0xd472: nop      word ptr cs:[rax + rax]
0xd47d: nop      dword ptr [rax]
0xd480: endbr64
0xd484: push     r14
0xd486: push     r13
0xd488: push     r12
0xd48a: push     rbp
0xd48b: mov      rbp, qword ptr [rip + 0x1df76]
0xd492: push     rbx
0xd493: mov      rbx, qword ptr [rbp + 0x18]
0xd497: cmp      rbx, qword ptr [rbp + 0x38]
0xd49b: jb       0xd4b0
0xd49d: pop      rbx
0xd49e: pop      rbp
0xd49f: pop      r12
0xd4a1: pop      r13
0xd4a3: pop      r14
0xd4a5: ret
0xd4a6: nop      word ptr cs:[rax + rax]
0xd4b0: xor      eax, eax
0xd4b2: call     0xcc30
0xd4b7: cmp      rbx, qword ptr [rbp + 0x18]
0xd4bb: jae      0xd528
0xd4bd: call     0x4f00
0xd4c2: mov      r14d, 1

```

```

0xd4c8: mov     r13, rax
0xd4cb: jmp     0xd4f3
0xd4f3:
0xd4cd: nop     dword ptr [rax]
0xd4d0: call    0x47c0
0xd4d5: xor     r14d, r14d
0xd4d8: mov     rax, qword ptr [rax]
0xd4db: mov     eax, dword ptr [rax + r12*4]
0xd4df: mov     byte ptr [rbp], al
0xd4e2: mov     rbp, qword ptr [rip + 0x1df1f]
0xd4e9: add     rbx, 1
0xd4ed: cmp     qword ptr [rbp + 0x18], rbx
0xd4f1: jbe     0xd528
0xd4f3: mov     rax, qword ptr [rbp + 0x58]
0xd4f7: add     rax, rbx
0xd4fa: movsx   r12, byte ptr [rax]
0xd4fe: mov     rbp, rax
0xd501: mov     rax, qword ptr [r13]
0xd505: test    byte ptr [rax + r12*2], 8
0xd50a: je      0xd4e2
0xd50c: test    r14d, r14d
0xd50f: jne     0xd4d0
0xd511: call    0x4ef0
0xd516: mov     rax, qword ptr [rax]
0xd519: mov     eax, dword ptr [rax + r12*4]
0xd51d: mov     byte ptr [rbp], al
0xd520: jmp     0xd4e2
0xd4e2:
0xd522: nop     word ptr [rax + rax]
0xd528: mov     dword ptr [rbp + 0x10], 0x14
0xd52f: pop     rbx
0xd530: pop     rbp
0xd531: pop     r12
0xd533: pop     r13
0xd535: pop     r14
0xd537: ret
0xd538: nop     dword ptr [rax + rax]
0xd540: endbr64
0xd544: push    r12
0xd546: mov     r12, rdi
0xd549: push    rbp
0xd54a: push    rbx
0xd54b: call    0x49b0
0xd550: mov     rbp, qword ptr [rip + 0x1deb1]
0xd557: mov     rbx, rax
0xd55a: mov     rsi, qword ptr [rbp + 0x40]
0xd55e: sub     rax, rsi
0xd561: add     qword ptr [rbp + 0x18], rax

```

```

0xd565: add    qword ptr [rbp + 0x20], rax
0xd569: mov    rax, qword ptr [rbp + 0x48]
0xd56d: lea    rdx, [rax + 1]
0xd571: lea    r8, [rbx + rdx]
0xd575: cmp    r8, qword ptr [rbp + 0x50]
0xd579: ja     0xd5d8
0xd57b: mov    rdi, qword ptr [rbp + 0x58]
0xd57f: add    rsi, rdi
0xd582: add    rdi, rbx
0xd585: call   0x4d40
0xd58a: mov    rax, qword ptr [rip + 0x1de77]
0xd591: mov    rdx, rbx
0xd594: mov    rsi, r12
0xd597: mov    rdi, qword ptr [rax + 0x58]
0xd59b: call   0x4bb0
0xd5a0: mov    rax, qword ptr [rip + 0x1de61]
0xd5a7: mov    rsi, rbx
0xd5aa: mov    rdi, qword ptr [rax + 0x58]
0xd5ae: call   0x184d0
0xd5b3: mov    rax, qword ptr [rip + 0x1de4e]
0xd5ba: mov    qword ptr [rax + 0x40], rbx
0xd5be: add    rbx, qword ptr [rax + 0x48]
0xd5c2: mov    qword ptr [rax + 0x38], rbx
0xd5c6: mov    dword ptr [rax + 0x10], 0x15
0xd5cd: pop    rbx
0xd5ce: pop    rbp
0xd5cf: pop    r12
0xd5d1: ret
0xd5d2: nop    word ptr [rax + rax]
0xd5d8: mov    qword ptr [rbp + 0x50], r8
0xd5dc: mov    rdi, qword ptr [rbp + 0x58]
0xd5e0: mov    rsi, r8
0xd5e3: call   0x1d690
0xd5e8: mov    qword ptr [rbp + 0x58], rax
0xd5ec: mov    rbp, qword ptr [rip + 0x1de15]
0xd5f3: mov    rax, qword ptr [rbp + 0x48]
0xd5f7: mov    rsi, qword ptr [rbp + 0x40]
0xd5fb: lea    rdx, [rax + 1]
0xd5ff: jmp    0xd57b
0xd57b:
0xd604: nop    word ptr cs:[rax + rax]
0xd60f: nop
0xd610: endbr64
0xd614: test   rdi, rdi
0xd617: je     0xd6f0
0xd61d: push   r12
0xd61f: push   rbp
0xd620: mov    rbp, rdi

```

```

0xd623: push    rbx
0xd624: call    0x49b0
0xd629: mov     r12, qword ptr [rip + 0x1ddd8]
0xd630: mov     rbx, rax
0xd633: mov     rax, qword ptr [r12 + 0x38]
0xd638: lea     rdx, [rax + 1]
0xd63c: mov     rax, qword ptr [r12 + 0x50]
0xd641: lea     rcx, [rbx + rdx]
0xd645: cmp     rcx, rax
0xd648: ja      0xd6c0
0xd64a: mov     rax, qword ptr [r12 + 0x18]
0xd64f: mov     rdi, qword ptr [r12 + 0x58]
0xd654: sub     rdx, rax
0xd657: lea     rsi, [rdi + rax]
0xd65b: add     rax, rbx
0xd65e: add     rdi, rax
0xd661: call    0x4d40
0xd666: mov     rax, qword ptr [rip + 0x1dd9b]
0xd66d: mov     rdx, rbx
0xd670: mov     rsi, rbp
0xd673: mov     rdi, qword ptr [rax + 0x18]
0xd677: add     rdi, qword ptr [rax + 0x58]
0xd67b: call    0x4bb0
0xd680: mov     rax, qword ptr [rip + 0x1dd81]
0xd687: mov     rsi, rbx
0xd68a: mov     rdi, qword ptr [rax + 0x18]
0xd68e: add     rdi, qword ptr [rax + 0x58]
0xd692: call    0x184d0
0xd697: mov     rax, qword ptr [rip + 0x1dd6a]
0xd69e: add     qword ptr [rax + 0x18], rbx
0xd6a2: add     qword ptr [rax + 0x38], rbx
0xd6a6: add     qword ptr [rax + 0x48], rbx
0xd6aa: mov     dword ptr [rax + 0x10], 0x16
0xd6b1: pop     rbx
0xd6b2: pop     rbp
0xd6b3: pop     r12
0xd6b5: ret
0xd6b6: nop     word ptr cs:[rax + rax]
0xd6c0: lea     rsi, [rax + rbx + 0x21]
0xd6c5: mov     rdi, qword ptr [r12 + 0x58]
0xd6ca: mov     qword ptr [r12 + 0x50], rsi
0xd6cf: call    0x1d690
0xd6d4: mov     qword ptr [r12 + 0x58], rax
0xd6d9: mov     r12, qword ptr [rip + 0x1dd28]
0xd6e0: mov     rax, qword ptr [r12 + 0x38]
0xd6e5: lea     rdx, [rax + 1]
0xd6e9: jmp     0xd64a
0xd64a:

```

```

0xd6ee: nop
0xd6f0: ret
0xd6f1: nop      word ptr cs:[rax + rax]
0xd6fc: nop      dword ptr [rax]
0xd700: endbr64
0xd704: mov      rax, qword ptr [rip + 0x1dcfd]
0xd70b: mov      rdi, qword ptr [rax + 0x60]
0xd70f: test     rdi, rdi
0xd712: je       0xd730
0xd714: sub      rsp, 8
0xd718: call     0xd610
0xd71d: mov      rax, qword ptr [rip + 0x1dce4]
0xd724: mov      dword ptr [rax + 0x10], 0x1a
0xd72b: add      rsp, 8
0xd72f: ret
0xd730: ret
0xd731: nop      word ptr cs:[rax + rax]
0xd73c: nop      dword ptr [rax]
0xd740: endbr64
0xd744: mov      rax, qword ptr [rip + 0x1dcbd]
0xd74b: push     r12
0xd74d: push     rbp
0xd74e: push     rbx
0xd74f: mov      rbx, qword ptr [rax + 0x40]
0xd753: mov      r12, qword ptr [rax + 0x28]
0xd757: mov      eax, dword ptr [rax + 0xc]
0xd75a: lea      rbp, [rbx + 3]
0xd75e: test     eax, eax
0xd760: je       0xd7f0
0xd766: mov      edx, dword ptr [rip + 0x1caa4]
0xd76c: mov      esi, dword ptr [rip + 0x1ca9a]
0xd772: mov      edi, dword ptr [rip + 0x1ca9c]
0xd778: call     0xa570
0xd77d: mov      rcx, qword ptr [rip + 0x1dc84]
0xd784: cmp      r12, rbp
0xd787: jb       0xd810
0xd78d: mov      rsi, qword ptr [rcx + 0x18]
0xd791: mov      rdx, qword ptr [rcx + 0x40]
0xd795: xor      ebx, ebx
0xd797: mov      rdi, qword ptr [rcx + 0x28]
0xd79b: mov      rax, rdi
0xd79e: sub      rax, rdx
0xd7a1: mov      edx, 1
0xd7a6: mov      r8, rax
0xd7a9: shr      r8, 2
0xd7ad: cmp      rax, 3
0xd7b1: cmova    edx, r8d
0xd7b5: xor      eax, eax

```

```

0xd7b7: cmp      rdi, rsi
0xd7ba: ja          0xd7d3
0xd7bc: mov      rax, rsi
0xd7bf: sub      rax, rdi
0xd7c2: movsxd   rdi, edx
0xd7c5: lea      r8d, [rdx + rax]
0xd7c9: xor      edx, edx
0xd7cb: div      rdi
0xd7ce: mov      eax, r8d
0xd7d1: sub      eax, edx
0xd7d3: mov      edx, esi
0xd7d5: mov      rdi, qword ptr [rcx]
0xd7d8: xor      esi, esi
0xd7da: sub      edx, eax
0xd7dc: call     0xc680
0xd7e1: cmp      r12, rbp
0xd7e4: jb       0xd840
0xd7e6: pop      rbx
0xd7e7: pop      rbp
0xd7e8: pop      r12
0xd7ea: ret
0xd7eb: nop      dword ptr [rax + rax]
0xd7f0: mov      edx, dword ptr [rip + 0x1ca0e]
0xd7f6: mov      esi, dword ptr [rip + 0x1ca04]
0xd7fc: mov      edi, dword ptr [rip + 0x1ca06]
0xd802: jmp      0xd778
0xd778:
0xd807: nop      word ptr [rax + rax]
0xd810: mov      rbx, qword ptr [rcx + 0x40]
0xd814: mov      rsi, qword ptr [rcx + 0x18]
0xd818: mov      qword ptr [rcx + 0x40], 0
0xd820: xor      edx, edx
0xd822: add      qword ptr [rcx + 0x58], rbx
0xd826: sub      rsi, rbx
0xd829: sub      qword ptr [rcx + 0x38], rbx
0xd82d: sub      qword ptr [rcx + 0x20], rbx
0xd831: movsxd   rbx, ebx
0xd834: mov      qword ptr [rcx + 0x18], rsi
0xd838: jmp      0xd797
0xd797:
0xd83d: nop      dword ptr [rax]
0xd840: mov      rax, qword ptr [rip + 0x1dbc1]
0xd847: movsxd   rbx, ebx
0xd84a: mov      qword ptr [rax + 0x40], rbx
0xd84e: sub      qword ptr [rax + 0x58], rbx
0xd852: add      qword ptr [rax + 0x38], rbx
0xd856: add      qword ptr [rax + 0x18], rbx
0xd85a: add      qword ptr [rax + 0x20], rbx

```



```

0xd85e: pop      rbx
0xd85f: pop      rbp
0xd860: pop      r12
0xd862: ret
0xd863: nop      word ptr cs:[rax + rax]
0xd86e: nop
0xd870: endbr64
0xd874: push     r15
0xd876: push     r14
0xd878: push     r13
0xd87a: push     r12
0xd87c: push     rbp
0xd87d: push     rbx
0xd87e: sub      rsp, 0x28
0xd882: mov      rax, qword ptr fs:[0x28]
0xd88b: mov      qword ptr [rsp + 0x18], rax
0xd890: mov      rax, qword ptr [rip + 0x1db71]
0xd897: lea      r14, [rsp + 0x17]
0xd89c: mov      rdi, r14
0xd89f: mov      rbx, qword ptr [rax + 0x40]
0xd8a3: mov      r13, qword ptr [rax + 0x28]
0xd8a7: call     0xa6a0
0xd8ac: mov      rax, qword ptr [rip + 0x1db55]
0xd8b3: add      rbx, 3
0xd8b7: mov      ecx, dword ptr [rax + 0xc]
0xd8ba: test     ecx, ecx
0xd8bc: je       0xda20
0xd8c2: mov      edx, dword ptr [rip + 0x1c948]
0xd8c8: mov      esi, dword ptr [rip + 0x1c93e]
0xd8ce: mov      edi, dword ptr [rip + 0x1c940]
0xd8d4: call     0xa570
0xd8d9: mov      rdx, qword ptr [rip + 0x1db28]
0xd8e0: cmp      r13, rbx
0xd8e3: jb       0xda90
0xd8e9: mov      qword ptr [rsp + 8], 0
0xd8f2: mov      rax, qword ptr [rdx + 0x18]
0xd8f6: mov      rcx, qword ptr [rdx + 0x40]
0xd8fa: mov      rdi, qword ptr [rdx + 0x28]
0xd8fe: mov      r15, rdi
0xd901: sub      r15, rcx
0xd904: cmp      r15, 3
0xd908: jbe      0xda80
0xd90e: shr      r15, 2
0xd912: movsxd   rcx, r15d
0xd915: cmp      rdi, rax
0xd918: ja       0xda70
0xd91e: mov      edx, eax
0xd920: sub      rax, rdi

```

```

0xd923: sub     edx, edi
0xd925: add     r15d, edx
0xd928: xor     edx, edx
0xd92a: div     rcx
0xd92d: sub     r15d, edx
0xd930: mov     r12d, r15d
0xd933: call    0x1d670
0xd938: mov     esi, 0x20
0xd93d: mov     rbp, rax
0xd940: mov     rax, qword ptr [rip + 0x1dac1]
0xd947: mov     rdi, rbp
0xd94a: mov     rdx, qword ptr [rax + 0x28]
0xd94e: call    0x4a50
0xd953: mov     rax, qword ptr [rip + 0x1daae]
0xd95a: mov     edx, dword ptr [rax + 8]
0xd95d: test    edx, edx
0xd95f: je      0xda40
0xd965: mov     rdx, qword ptr [rax + 0x38]
0xd969: sub     rdx, qword ptr [rax + 0x40]
0xd96d: mov     rdi, rbp
0xd970: sub     rdx, r12
0xd973: mov     rsi, qword ptr [rax + 0x28]
0xd977: sub     rsi, qword ptr [rax + 0x40]
0xd97b: cmp     rdx, rsi
0xd97e: cmova   rdx, rsi
0xd982: mov     rsi, qword ptr [rax + 0x40]
0xd986: add     rsi, r12
0xd989: add     rsi, qword ptr [rax + 0x58]
0xd98d: call    0x4bb0
0xd992: mov     rax, qword ptr [rip + 0x1da6f]
0xd999: xor     edx, edx
0xd99b: xor     esi, esi
0xd99d: mov     rdi, qword ptr [rax]
0xd9a0: call    0xc680
0xd9a5: cmp     r13, rbx
0xd9a8: jae     0xdaf0
0xd9ae: mov     rax, qword ptr [rip + 0x1da53]
0xd9b5: mov     rsi, rbp
0xd9b8: mov     rdi, qword ptr [rax]
0xd9bb: mov     rdx, qword ptr [rax + 0x28]
0xd9bf: sub     edx, dword ptr [rax + 0x40]
0xd9c2: call    0xc600
0xd9c7: mov     rax, qword ptr [rip + 0x1da3a]
0xd9ce: xor     esi, esi
0xd9d0: mov     edx, dword ptr [rax + 0x18]
0xd9d3: mov     rdi, qword ptr [rax]
0xd9d6: sub     edx, r15d
0xd9d9: call    0xc680

```

```

0xd9de: cmp     r13, rbx
0xd9e1: jb      0xdac8
0xd9e7: mov     rdi, rbp
0xd9ea: call    0x18d10
0xd9ef: mov     rdi, r14
0xd9f2: call    0xa6b0
0xd9f7: mov     rax, qword ptr [rsp + 0x18]
0xd9fc: sub     rax, qword ptr fs:[0x28]
0xda05: jne     0xdb0b
0xda0b: add     rsp, 0x28
0xda0f: pop     rbx
0xda10: pop     rbp
0xda11: pop     r12
0xda13: pop     r13
0xda15: pop     r14
0xda17: pop     r15
0xda19: ret
0xda1a: nop     word ptr [rax + rax]
0xda20: mov     edx, dword ptr [rip + 0x1c7de]
0xda26: mov     esi, dword ptr [rip + 0x1c7d4]
0xda2c: mov     edi, dword ptr [rip + 0x1c7d6]
0xda32: jmp     0xd8d4
0xd8d4
0xda37: nop     word ptr [rax + rax]
0xda40: mov     rsi, qword ptr [rax + 0x40]
0xda44: mov     rdx, qword ptr [rax + 0x38]
0xda48: mov     rdi, rbp
0xda4b: mov     rax, qword ptr [rax + 0x28]
0xda4f: sub     rdx, rsi
0xda52: sub     rdx, r12
0xda55: sub     rax, rsi
0xda58: mov     esi, 0x2a
0xda5d: cmp     rdx, rax
0xda60: cmova   rdx, rax
0xda64: call    0x4a50
0xda69: jmp     0xd992
0xd992
0xda6e: nop
0xda70: xor     r12d, r12d
0xda73: xor     r15d, r15d
0xda76: jmp     0xd933
0xd933
0xda7b: nop     dword ptr [rax + rax]
0xda80: mov     ecx, 1
0xda85: mov     r15d, 1
0xda8b: jmp     0xd915
0xd915
0xda90: mov     r12, qword ptr [rdx + 0x40]

```

```

0xda94: mov     rax, qword ptr [rdx + 0x18]
0xda98: mov     qword ptr [rdx + 0x40], 0
0xdaa0: add     qword ptr [rdx + 0x58], r12
0xdaa4: movsxd  rcx, r12d
0xdaa7: sub     rax, r12
0xdaaa: sub     qword ptr [rdx + 0x38], r12
0xdaae: mov     qword ptr [rsp + 8], rcx
0xdab3: xor     ecx, ecx
0xdab5: sub     qword ptr [rdx + 0x20], r12
0xdab9: mov     qword ptr [rdx + 0x18], rax
0xdabd: jmp     0xd8fa
0xd8fa:
0xdac2: nop     word ptr [rax + rax]
0xdac8: mov     rax, qword ptr [rip + 0x1d939]
0xdacf: movsxd  r12, dword ptr [rsp + 8]
0xdad4: sub     qword ptr [rax + 0x58], r12
0xdad8: add     qword ptr [rax + 0x38], r12
0xdadc: add     qword ptr [rax + 0x18], r12
0xdae0: add     qword ptr [rax + 0x20], r12
0xdae4: mov     qword ptr [rax + 0x40], r12
0xdae8: jmp     0xd9e7
0xd9e7:
0xdaed: nop     dword ptr [rax]
0xdaf0: mov     rax, qword ptr [rip + 0x1d911]
0xdaf7: mov     rsi, qword ptr [rax + 0x58]
0xdafb: mov     edx, dword ptr [rax + 0x40]
0xdafe: mov     rdi, qword ptr [rax]
0xdb01: call    0xc600
0xdb06: jmp     0xd9ae
0xd9ae:
0xdb0b: call    0x49d0
0xdb10: endbr64
0xdb14: mov     rax, qword ptr [rip + 0x1d8ed]
0xdb1b: push    rbx
0xdb1c: mov     rbx, rdi
0xdb1f: mov     rdi, qword ptr [rdi]
0xdb22: mov     rsi, qword ptr [rax + 0x48]
0xdb26: add     rsi, 1
0xdb2a: call    0x1d690
0xdb2f: mov     qword ptr [rbx], rax
0xdb32: mov     rdi, rax
0xdb35: mov     rax, qword ptr [rip + 0x1d8cc]
0xdb3c: mov     rcx, qword ptr [rax + 0x48]
0xdb40: mov     rsi, qword ptr [rax + 0x40]
0xdb44: add     rsi, qword ptr [rax + 0x58]
0xdb48: lea     rdx, [rcx + 1]
0xdb4c: call    0x4bb0
0xdb51: mov     rax, qword ptr [rip + 0x1d8b0]

```

```

0xdb58: pop      rbx
0xdb59: mov      eax, dword ptr [rax + 0x48]
0xdb5c: ret
0xdb5d: nop      dword ptr [rax]
0xdb60: endbr64
0xdb64: push     rbp
0xdb65: xor      eax, eax
0xdb67: mov      rbp, rdi
0xdb6a: call     0xcff0
0xdb6f: test     rbp, rbp
0xdb72: lea      rax, [rip + 0x133bf]
0xdb79: cmov     rbp, rax
0xdb7d: mov      rdi, rbp
0xdb80: call     0xd540
0xdb85: xor      eax, eax
0xdb87: call     0xd870
0xdb8c: xor      eax, eax
0xdb8e: pop      rbp
0xdb8f: jmp      0xd740
0xd740:
0xdb94: nop      word ptr cs:[rax + rax]
0xdb9f: nop
0xdba0: endbr64
0xdba4: mov      rax, qword ptr [rip + 0x1d85d]
0xdbab: mov      dword ptr [rax + 0xc], edi
0xdbae: ret
0xdbaf: nop
0xdbb0: endbr64
0xdbb4: push     r12
0xdbb6: push     rbp
0xdbb7: push     rbx
0xdbb8: cmp      edi, 1
0xdbbb: je       0xdbd0
0xdbbd: cmp      edi, 2
0xdbc0: je       0xdc90
0xdbc6: test     edi, edi
0xdbc8: je       0xdc40
0xdbca: pop      rbx
0xdbcb: pop      rbp
0xdbcc: pop      r12
0xdbce: ret
0xdbcf: nop
0xdbd0: mov      ecx, dword ptr [rip + 0x1d83e]
0xdbd6: test     ecx, ecx
0xdbd8: je       0xdd38
0xdbde: xor      eax, eax
0xdbe0: call     0x1ab10
0xdbe5: mov      rbp, rax

```

```

0xdbe8: test    rax, rax
0xdbeb: je      0xdbca
0xdbed: mov     r12, qword ptr [rbp]
0xdbf1: mov     rbx, qword ptr [rip + 0x1d810]
0xdbf8: mov     rdi, r12
0xdbfb: call    0x49b0
0xdc00: mov     rdi, qword ptr [rbx + 0x40]
0xdc04: mov     qword ptr [rbx + 0x48], rax
0xdc08: lea     rsi, [rax + rdi]
0xdc0c: mov     qword ptr [rbx + 0x38], rsi
0xdc10: mov     qword ptr [rbx + 0x18], rsi
0xdc14: add     rsi, 1
0xdc18: cmp     rsi, qword ptr [rbx + 0x50]
0xdc1c: ja      0xdd70
0xdc22: add     rdi, qword ptr [rbx + 0x58]
0xdc26: mov     rsi, r12
0xdc29: call    0x48b0
0xdc2e: pop     rbx
0xdc2f: xor     eax, eax
0xdc31: pop     rbp
0xdc32: pop     r12
0xdc34: jmp     0xd870
0xd870:
0xdc39: nop     dword ptr [rax]
0xdc40: mov     eax, dword ptr [rip + 0x1d7ce]
0xdc46: test    eax, eax
0xdc48: jne     0xdd18
0xdc4e: xor     eax, eax
0xdc50: call    0x1ab10
0xdc55: test    rax, rax
0xdc58: je      0xdd98
0xdc5e: mov     rdx, qword ptr [rip + 0x1d7a3]
0xdc65: mov     rsi, qword ptr [rax]
0xdc68: mov     rbp, qword ptr [rdx + 0x40]
0xdc6c: add     rbp, qword ptr [rdx + 0x58]
0xdc70: mov     rdi, rbp
0xdc73: call    0x4b30
0xdc78: test    eax, eax
0xdc7a: jne     0xdd58
0xdc80: pop     rbx
0xdc81: xor     eax, eax
0xdc83: pop     rbp
0xdc84: pop     r12
0xdc86: jmp     0x1ab40
0x1ab40:
0xdc8b: nop     dword ptr [rax + rax]
0xdc90: mov     edx, dword ptr [rip + 0x1d77e]
0xdc96: test    edx, edx

```

```

0xdc98: je      0xdcf8
0xdc9a: xor      eax, eax
0xdc9c: call     0x1ab40
0xdca1: mov      rbp, rax
0xdca4: test     rax, rax
0xdca7: jne      0xdbed
0xdcad: mov      rax, qword ptr [rip + 0x1d754]
0xdcb4: mov      edi, 2
0xdcb9: mov      rdx, qword ptr [rax + 0x40]
0xdcbd: mov      dword ptr [rax + 0x10], 6
0xdcc4: mov      qword ptr [rax + 0x20], rdx
0xdcc8: mov      rdx, qword ptr [rax + 0x38]
0xdccc: mov      qword ptr [rax + 0x18], rdx
0xcdcd0: xor      eax, eax
0xcdcd2: call     0xc6e0
0xcdcd7: mov      rax, qword ptr [rip + 0x1d72a]
0xcdcde: mov      rdx, qword ptr [rax + 0x18]
0xdce2: mov      dword ptr [rax + 0x10], 0xd
0xdce9: mov      qword ptr [rax + 0x20], rdx
0xcdced: jmp      0xdc2e
0xdc2e
0xdcf2: nop      word ptr [rax + rax]
0xdcf8: mov      dword ptr [rip + 0x1d712], 1
0xdd02: xor      eax, eax
0xdd04: call     0x1a570
0xdd09: mov      dword ptr [rip + 0x1d701], eax
0xdd0f: jmp      0xdc9a
0xdc9a
0xdd11: nop      dword ptr [rax]
0xdd18: mov      edi, dword ptr [rip + 0x1d6f2]
0xdd1e: xor      eax, eax
0xdd20: call     0x1ac30
0xdd25: mov      dword ptr [rip + 0x1d6e5], 0
0xdd2f: jmp      0xdc4e
0xdc4e
0xdd34: nop      dword ptr [rax]
0xdd38: mov      dword ptr [rip + 0x1d6d2], 1
0xdd42: xor      eax, eax
0xdd44: call     0x1a570
0xdd49: mov      dword ptr [rip + 0x1d6c1], eax
0xdd4f: jmp      0xdbde
0xdbde
0xdd54: nop      dword ptr [rax]
0xdd58: xor      eax, eax
0xdd5a: mov      rdi, rbp
0xdd5d: call     0x1a380
0xdd62: xor      eax, eax
0xdd64: call     0x1ab40

```

```

0xdd69: jmp      0xdc80
0xdc80:
0xdd6e: nop
0xdd70: mov     qword ptr [rbx + 0x50], rsi
0xdd74: mov     rdi, qword ptr [rbx + 0x58]
0xdd78: call    0x1d690
0xdd7d: mov     qword ptr [rbx + 0x58], rax
0xdd81: mov     rbx, qword ptr [rip + 0x1d680]
0xdd88: mov     r12, qword ptr [rbp]
0xdd8c: mov     rdi, qword ptr [rbx + 0x40]
0xdd90: jmp     0xdc22
0xdc22:
0xdd95: nop     dword ptr [rax]
0xdd98: mov     rax, qword ptr [rip + 0x1d669]
0xdd9f: mov     rdi, qword ptr [rax + 0x40]
0xdda3: add     rdi, qword ptr [rax + 0x58]
0xdda7: xor     eax, eax
0xdda9: call    0x1a380
0xddae: jmp     0xdc80
0xdc80:
0xddb3: nop     word ptr cs:[rax + rax]
0xddbd: nop     dword ptr [rax]
0dddc0: endbr64
0dddc4: mov     eax, dword ptr [rip + 0x1be96]
0xddca: push    rbp
0ddccb: mov     rbp, rdi
0xddce: lea     rdi, [rip + 0x1be73]
0ddd5: test    eax, eax
0ddd7: lea     rax, [rip + 0x1be52]
0ddde: cmov    rdi, rax
0dde2: call    0x17c50
0dde7: mov     edx, 9
0ddeec: lea     rsi, [rip + 0x1c4ad]
0ddf3: lea     rdi, [rip + 0x1c466]
0dddfa: call    0x17f70
0dddff: mov     rdi, rbp
0xde02: call    0x1d7a0
0xde07: mov     rdi, rax
0xde0a: mov     rbp, rax
0xde0d: mov     qword ptr [rip + 0x1d794], rax
0xde14: call    0x49b0
0xde19: mov     rdi, rbp
0xde1c: mov     rsi, rax
0xde1f: call    0x184d0
0xde24: xor     eax, eax
0xde26: call    0xc590
0xde2b: lea     rdi, [rip + 0x1d5ee]
0xde32: pop     rbp

```



```

0xde33: mov     qword ptr [rip + 0xd78e], rax
0xde3a: jmp     0x4a30
0x4a30:
0xde3f: nop
0xde40: endbr64
0xde44: mov     rdi, qword ptr [rip + 0xd77d]
0xde4b: jmp     0xc5c0
0xc5c0:
0xde50: endbr64
0xde54: push    rbp
0xde55: mov     rbp, rdi
0xde58: push    rbx
0xde59: mov     rbx, rsi
0xde5c: sub     rsp, 8
0xde60: mov     rdi, qword ptr [rip + 0xd749]
0xde67: test    rdi, rdi
0xde6a: je      0xde71
0xde6c: call    0x18d10
0xde71: mov     rdi, rbp
0xde74: call    0x1d670
0xde79: mov     rdi, qword ptr [rip + 0xd748]
0xde80: mov     edx, ebx
0xde82: xor     esi, esi
0xde84: mov     qword ptr [rip + 0xd725], rax
0xde8b: add     rsp, 8
0xde8f: mov     r8d, ebp
0xde92: mov     ecx, 1
0xde97: pop     rbx
0xde98: pop     rbp
0xde99: jmp     0xc5e0
0xc5e0:
0xde9e: nop
0xdea0: endbr64
0xdea4: push    rbp
0xdea5: mov     rbp, rsp
0xdea8: push    r15
0xdeaa: push    r14
0xdeac: push    r13
0xdeae: push    r12
0xdeb0: push    rbx
0xdeb1: sub     rsp, 0x68
0xdeb5: mov     rax, qword ptr fs:[0x28]
0xdeb6: mov     qword ptr [rbp - 0x38], rax
0xdec2: xor     eax, eax
0xdec4: lea     rax, [rbp - 0x61]
0xdec8: mov     rdi, rax
0xdecb: mov     qword ptr [rbp - 0x80], rax
0xdecf: call    0xa6a0

```

```

0xded4: mov     rax, qword ptr [rip + 0x1d6ed]
0xdedb: mov     esi, 0x20
0xdee0: mov     rdi, qword ptr [rip + 0x1d6c9]
0xdee7: movsxd  rdx, dword ptr [rax + 0xc]
0xdeeb: call    0x4a50
0xdef0: mov     rdi, qword ptr [rip + 0x1d6c9]
0xdef7: call    0x49b0
0xdefc: lea     r14, [rax + 1]
0xdf00: mov     rdi, r14
0xdf03: call    0x1d670
0xdf08: mov     rcx, qword ptr [rip + 0x1d6b1]
0xdf0f: mov     r13, rax
0xdf12: movzx   eax, byte ptr [rcx]
0xdf15: test    al, al
0xdf17: je      0xe430
0xdf1d: xor     esi, esi
0xdf1f: xor     ebx, ebx
0xdf21: lea     r15, [rip + 0x13188]
0xdf28: xor     edx, edx
0xdf2a: mov     dword ptr [rbp - 0x74], esi
0xdf2d: jmp     0xdf48
0xdf48:
0xdf2f: nop
0xdf30: cmp     al, 9
0xdf32: je      0xdf80
0xdf34: add     dword ptr [rbp - 0x74], 1
0xdf38: mov     byte ptr [r13 + r12], al
0xdf3d: movsxd  rdx, ebx
0xdf40: movzx   eax, byte ptr [rcx + rdx]
0xdf44: test    al, al
0xdf46: je      0xdfc0
0xdf48: movsxd  r12, dword ptr [rbp - 0x74]
0xdf4c: add     ebx, 1
0xdf4f: cmp     al, 0x5c
0xdf51: jne     0xdf30
0xdf53: lea     rdx, [rcx + rdx + 1]
0xdf58: movzx   eax, byte ptr [rdx]
0xdf5b: test    al, al
0xdf5d: je      0xe320
0xdf63: sub     eax, 0x5c
0xdf66: cmp     al, 0x17
0xdf68: ja      0xe0d8
0xdf6e: movzx   eax, al
0xdf71: movsxd  rax, dword ptr [r15 + rax*4]
0xdf75: add     rax, r15
0xdf78: jmp     rax
rax

```

->->->

indirect JMP

```

0x8c38: mov     rdi, qword ptr [r12]
0x8c3c: call    0x18d10
0x8c41: mov     qword ptr [r12], 0
0x8c49: xor     eax, eax
0x8c4b: mov     rdx, qword ptr [rsp + 0x58]
0x8c50: sub     rdx, qword ptr fs:[0x28]
0x8c59: jne     0x929f
0x8c5f: add     rsp, 0x68
0x8c63: pop     rbx
0x8c64: pop     rbp
0x8c65: pop     r12
0x8c67: pop     r13
0x8c69: pop     r14
0x8c6b: pop     r15
0x8c6d: ret
0x8c6e: nop
0x8c70: mov     rdi, r15
0x8c73: call    0xf510
0x8c78: cmp     eax, 1
0x8c7b: jne     0x8d61
0x8c81: nop     dword ptr [rax]
0x8c88: mov     rdi, r15
0x8c8b: call    0xf490
0x8c90: mov     r14, rax
0x8c93: mov     rdi, r14
0x8c96: call    0x49b0
0x8c9b: lea     rdi, [rax + 3]
0x8c9f: call    0x1d670
0x8ca4: mov     r8, r14
0x8ca7: mov     qword ptr [rsp + 0x50], rax
0x8cac: mov     rdi, rax
0x8caf: lea     rcx, [rip + 0x185df]
0x8cb6: mov     rdx, -1
0x8cbd: mov     esi, 1
0x8cc2: xor     eax, eax
0x8cc4: call    0x4f10
0x8cc9: mov     byte ptr [rbx], 0
0x8ccc: mov     rdi, qword ptr [rsp + 0x50]
0x8cd1: lea     r14, [rbp + 1]
0x8cd5: test    rdi, rdi
0x8cd8: je      0x8bd9
0x8cde: call    0x49b0
0x8ce3: mov     rdi, qword ptr [r12]
0x8ce7: add     r13, rax
0x8cea: mov     rsi, r13
0x8ced: call    0x1d690
0x8cf2: mov     rsi, qword ptr [rsp + 0x50]
0x8cf7: mov     qword ptr [r12], rax

```

```

0x8cfb: mov     rdi, rax
0x8cfe: call    0x4df0
0x8d03: mov     rbx, qword ptr [r12]
0x8d07: mov     rdi, rbx
0x8d0a: call    0x49b0
0x8d0f: mov     rdi, qword ptr [rsp + 0x50]
0x8d14: add     rbx, rax
0x8d17: call    0x18d10
0x8d1c: mov     qword ptr [rsp + 0x50], 0
0x8d25: movzx   eax, byte ptr [r14]
0x8d29: test    al, al
0x8d2b: jne     0x8be5
0x8d31: nop     dword ptr [rax]
0x8d38: mov     eax, dword ptr [rsp + 0x24]
0x8d3c: add     eax, 1
0x8d3f: mov     byte ptr [rbx], 0
0x8d42: jmp     0x8c4b
0x8c4b:
0x8d47: nop     word ptr [rax + rax]
0x8d50: mov     rdi, r15
0x8d53: call    0xf510
0x8d58: cmp     eax, 4
0x8d5b: je      0x8c88
0x8d61: mov     rdi, qword ptr [rsp + 0x50]
0x8d66: test    rdi, rdi
0x8d69: je      0x8d70
0x8d6b: call    0x18d10
0x8d70: mov     qword ptr [r12], 0
0x8d78: mov     eax, 0xffffffff
0x8d7d: jmp     0x8c4b
0x8c4b:
0x8d82: nop     word ptr [rax + rax]
0x8d88: mov     rdi, r15
0x8d8b: call    0xf510
0x8d90: cmp     eax, 3
0x8d93: je      0x8c88
0x8d99: jmp     0x8d61
0x8d61:
0x8d9b: nop     dword ptr [rax + rax]
0x8da0: mov     rdi, qword ptr [r15 + 0x18]
0x8da4: call    0x49b0
0x8da9: lea     rdi, [rax + 3]
0x8dad: call    0x1d670
0x8db2: mov     r8, qword ptr [r15 + 0x18]
0x8db6: mov     qword ptr [rsp + 0x50], rax
0x8dbb: mov     rdi, rax
0x8dbe: jmp     0x8caf
0x8caf:

```

```

0x8dc3: nop      dword ptr [rax + rax]
0x8dc8: mov      rdi, r15
0x8dcb: call     0xf4b0
0x8dd0: mov      edi, eax
0x8dd2: mov      r14d, eax
0x8dd5: call     0x4970
0x8dda: test     rax, rax
0x8ddd: je       0x8eab
0x8de3: mov      rdi, qword ptr [rax]
0x8de6: call     0x1d7a0
0x8deb: mov      qword ptr [rsp + 0x50], rax
0x8df0: jmp      0x8cc9
0x8cc9
0x8df5: nop      dword ptr [rax]
0x8df8: mov      edi, 0x10
0x8dfd: call     0x1d670
0x8e02: mov      rdi, r15
0x8e05: mov      qword ptr [rsp + 0x50], rax
0x8e0a: call     0xf4f0
0x8e0f: mov      rdi, qword ptr [rsp + 0x50]
0x8e14: mov      esi, 1
0x8e19: lea      rcx, [rip + 0x1771c]
0x8e20: and      eax, 0xffff
0x8e25: mov      rdx, -1
0x8e2c: mov      r8d, eax
0x8e2f: xor      eax, eax
0x8e31: call     0x4f10
0x8e36: jmp      0x8cc9
0x8cc9
0x8e3b: nop      dword ptr [rax + rax]
0x8e40: xor      eax, eax
0x8e42: cmp      dl, 0x69
0x8e45: mov      edx, dword ptr [rip + 0x21aad]
0x8e4b: setne    al
0x8e4e: add      eax, 1
0x8e51: mov      dword ptr [rsp + 0x24], eax
0x8e55: test     edx, edx
0x8e57: je       0x9198
0x8e5d: mov      eax, dword ptr [r15 + 0x70]
0x8e61: test     eax, eax
0x8e63: je       0x9198
0x8e69: mov      edi, 0x15
0x8e6e: call     0x1d670
0x8e73: movdqa   xmm0, xmmword ptr [rip + 0x17d35]
0x8e7b: mov      qword ptr [rsp + 0x50], rax
0x8e80: mov      byte ptr [rax + 0x10], 0
0x8e84: movups   xmmword ptr [rax], xmm0
0x8e87: jmp      0x8cc9

```

```

0x8cc9
0x8e8c: nop      dword ptr [rax]
0x8e90: mov      rdi, r15
0x8e93: call     0xf4d0
0x8e98: mov      edi, eax
0x8e9a: mov      r14d, eax
0x8e9d: call     0x4a10
0x8ea2: test     rax, rax
0x8ea5: jne      0x8de3
0x8eab: mov      edi, 0x10
0x8eb0: call     0x1d670
0x8eb5: mov      r8d, r14d
0x8eb8: mov      esi, 1
0x8ebd: lea      rcx, [rip + 0x17678]
0x8ec4: mov      qword ptr [rsp + 0x50], rax
0x8ec9: mov      rdi, rax
0x8ecc: mov      rdx, -1
0x8ed3: xor      eax, eax
0x8ed5: call     0x4f10
0x8eda: jmp      0x8cc9
0x8cc9
0x8edf: nop
0x8ee0: mov      rdi, r15
0x8ee3: call     0xf510
0x8ee8: cmp      eax, 2
0x8eeb: je       0x8c88
0x8ef1: jmp      0x8d61
0x8d61
0x8ef6: nop      word ptr cs:[rax + rax]
0x8f00: mov      rdi, r15
0x8f03: call     0xf510
0x8f08: test     eax, eax
0x8f0a: je       0x8c88
0x8f10: jmp      0x8d61
0x8d61
0x8f15: nop      dword ptr [rax]
0x8f18: mov      rdi, qword ptr [r15 + 0x18]
0x8f1c: mov      esi, 0x2f
0x8f21: lea      r14, [rip + 0x17604]
0x8f28: call     0x4a20
0x8f2d: mov      edi, 8
0x8f32: cmp      byte ptr [rax + 1], 0
0x8f36: je       0x8c9f
0x8f3c: lea      r14, [rax + 1]
0x8f40: jmp      0x8c93
0x8c93
0x8f45: mov      esi, dword ptr [rip + 0x219ad]
0x8f4b: test     esi, esi

```

```

0x8f4d: jne      0x8c28
0x8f53: cmp      byte ptr [r14 + 2], 0x7b
0x8f58: jne      0x8c38
0x8f5e: add      r14, 3
0x8f62: mov      esi, 0x7d
0x8f67: mov      rdi, r14
0x8f6a: call     0x4a00
0x8f6f: mov      rbp, rax
0x8f72: test     rax, rax
0x8f75: je       0x8c38
0x8f7b: mov      byte ptr [rax], 0
0x8f7e: mov      edx, 2
0x8f83: lea      rsi, [rip + 0x175a8]
0x8f8a: mov      rdi, r14
0x8f8d: call     0x8070
0x8f92: mov      byte ptr [rbp], 0x7d
0x8f96: cmp      al, 0x79
0x8f98: je       0x8cc9
0x8f9e: jmp      0x8d61
0x8d61:
0x8fa3: nop      dword ptr [rax + rax]
0x8fa8: mov      ecx, dword ptr [rip + 0x2194a]
0x8fae: test     ecx, ecx
0x8fb0: jne      0x8c28
0x8fb6: cmp      byte ptr [r14 + 2], 0x7b
0x8fbb: jne      0x8c38
0x8fc1: add      r14, 3
0x8fc5: mov      esi, 0x2c
0x8fca: mov      rdi, r14
0x8fcd: call     0x4a00
0x8fd2: mov      rbp, rax
0x8fd5: test     rax, rax
0x8fd8: je       0x8c38
0x8fde: mov      byte ptr [rax], 0
0x8fe1: mov      rax, qword ptr [rsp + 0x18]
0x8fe6: lea      r15, [rsp + 0x48]
0x8feb: mov      rcx, qword ptr [rsp + 0x10]
0x8ff0: mov      rdx, qword ptr [rsp + 8]
0x8ff5: mov      rsi, r15
0x8ff8: mov      dword ptr [rip + 0x218f6], 1
0x9002: mov      r8, qword ptr [rax + 8]
0x9006: mov      qword ptr [rax + 8], r14
0x900a: mov      rdi, rax
0x900d: mov      r14, rax
0x9010: xor      eax, eax
0x9012: mov      qword ptr [rsp + 0x28], r8
0x9017: call     0x8b50
0x901c: mov      r8, qword ptr [rsp + 0x28]

```

```

0x9021: mov     dword ptr [rip + 0x218cd], 0
0x902b: mov     qword ptr [r14 + 8], r8
0x902f: test    eax, eax
0x9031: jle     0x9290
0x9037: mov     rax, qword ptr [rsp + 0x18]
0x903c: mov     rdi, qword ptr [rax]
0x903f: call    0x49b0
0x9044: mov     r14, qword ptr [rsp + 0x48]
0x9049: mov     qword ptr [rsp + 0x28], rax
0x904e: mov     rdi, r14
0x9051: call    0x49b0
0x9056: mov     rdx, qword ptr [rsp + 0x28]
0x905b: lea     rdi, [rdx + rax + 0x11]
0x9060: call    0x1d670
0x9065: mov     r9, r14
0x9068: mov     esi, 1
0x906d: lea     rcx, [rip + 0x174c1]
0x9074: mov     r10, rax
0x9077: mov     rax, qword ptr [rsp + 0x18]
0x907c: mov     rdx, -1
0x9083: mov     rdi, r10
0x9086: mov     qword ptr [rsp + 0x28], r10
0x908b: mov     r8, qword ptr [rax]
0x908e: xor     eax, eax
0x9090: call    0x4f10
0x9095: mov     rdi, r14
0x9098: lea     r14, [rbp + 1]
0x909c: call    0x18d10
0x90a1: mov     byte ptr [rbp], 0x2c
0x90a5: mov     esi, 0x7d
0x90aa: mov     rdi, r14
0x90ad: mov     qword ptr [rsp + 0x48], 0
0x90b6: call    0x4a00
0x90bb: mov     rbp, rax
0x90be: test    rax, rax
0x90c1: je      0x8c38
0x90c7: mov     byte ptr [rax], 0
0x90ca: mov     rdi, qword ptr [rsp + 0x28]
0x90cf: call    0x49b0
0x90d4: mov     r10, qword ptr [rsp + 0x28]
0x90d9: cmp     rax, 0x32
0x90dd: jbe     0x90e4
0x90df: mov     byte ptr [r10 + 0x32], 0
0x90e4: mov     rax, qword ptr [rsp + 0x18]
0x90e9: mov     rcx, qword ptr [rsp + 0x10]
0x90ee: mov     rsi, r15
0x90f1: mov     qword ptr [rsp + 0x30], r10
0x90f6: mov     dword ptr [rip + 0x217f8], 1

```



```

0x9100: mov     rdx, qword ptr [rsp + 8]
0x9105: mov     r8, qword ptr [rax + 8]
0x9109: mov     qword ptr [rax + 8], r14
0x910d: mov     r15, rax
0x9110: mov     rdi, rax
0x9113: xor     eax, eax
0x9115: mov     qword ptr [rsp + 0x28], r8
0x911a: call    0x8b50
0x911f: mov     r8, qword ptr [rsp + 0x28]
0x9124: mov     r10, qword ptr [rsp + 0x30]
0x9129: mov     dword ptr [rip + 0x217c5], 0
0x9133: test    eax, eax
0x9135: mov     qword ptr [r15 + 8], r8
0x9139: jle     0x9266
0x913f: mov     rax, qword ptr [rsp + 0x18]
0x9144: mov     r15, qword ptr [rsp + 0x48]
0x9149: mov     rdi, r10
0x914c: lea     rsi, [rsp + 0x50]
0x9151: mov     qword ptr [rsp + 0x28], r10
0x9156: mov     rcx, qword ptr [rax + 0x28]
0x915a: mov     rdx, r15
0x915d: call    0x83e0
0x9162: mov     rdi, qword ptr [rsp + 0x28]
0x9167: mov     r14, rax
0x916a: call    0x18d10
0x916f: mov     rdi, r15
0x9172: call    0x18d10
0x9177: mov     byte ptr [rbp], 0x7d
0x917b: mov     qword ptr [rsp + 0x48], 0
0x9184: test    r14, r14
0x9187: jne     0x8cc9
0x918d: jmp     0x8d61
0x8d61
0x9192: nop     word ptr [rax + rax]
0x9198: mov     qword ptr [rsp + 0x50], 0
0x91a1: mov     rdi, r15
0x91a4: xor     r14d, r14d
0x91a7: call    0x100c0
0x91ac: mov     qword ptr [rsp + 0x28], rbx
0x91b1: mov     qword ptr [rsp + 0x30], rbp
0x91b6: mov     qword ptr [rsp + 0x38], r12
0x91bb: jmp     0x9232
0x9232
0x91bd: nop     dword ptr [rax]
0x91c0: cdqe
0x91c2: lea     rcx, [rax + rax*4]
0x91c6: mov     rax, qword ptr [r15 + 0x38]
0x91ca: shl     rcx, 4

```

```

0x91ce: mov     rdi, qword ptr [rax + rcx]
0x91d2: mov     r12, rcx
0x91d5: call    0x49b0
0x91da: mov     rdi, qword ptr [rsp + 0x50]
0x91df: lea     rbp, [rax + r14]
0x91e3: lea     rsi, [rbp + 4]
0x91e7: lea     rbx, [rbp + 3]
0x91eb: call    0x1d690
0x91f0: mov     qword ptr [rsp + 0x50], rax
0x91f5: mov     byte ptr [rax + r14], 0x22
0x91fa: mov     rax, qword ptr [r15 + 0x38]
0x91fe: mov     rsi, qword ptr [rax + r12]
0x9202: mov     rax, qword ptr [rsp + 0x50]
0x9207: lea     rdi, [rax + r14 + 1]
0x920c: mov     r14, rbx
0x920f: call    0x48b0
0x9214: mov     rax, qword ptr [rsp + 0x50]
0x9219: mov     byte ptr [rax + rbp + 1], 0x22
0x921e: mov     rax, qword ptr [rsp + 0x50]
0x9223: mov     byte ptr [rax + rbp + 2], 0x20
0x9228: mov     rax, qword ptr [rsp + 0x50]
0x922d: mov     byte ptr [rax + rbp + 3], 0
0x9232: mov     rdi, r15
0x9235: call    0x100e0
0x923a: cmp     eax, -1
0x923d: jne     0x91c0
0x923f: mov     rbx, qword ptr [rsp + 0x28]
0x9244: mov     rbp, qword ptr [rsp + 0x30]
0x9249: mov     r12, qword ptr [rsp + 0x38]
0x924e: test    r14, r14
0x9251: jne     0x8cc9
0x9257: jmp     0x8d61
0x8d61:
0x925c: mov     eax, 1
0x9261: jmp     0x8d3f
0x8d3f:
0x9266: mov     byte ptr [rbp], 0x7d
0x926a: mov     rdi, r10
0x926d: mov     dword ptr [rsp + 8], eax
0x9271: call    0x18d10
0x9276: mov     eax, dword ptr [rsp + 8]
0x927a: test    eax, eax
0x927c: je      0x8c38
0x9282: jmp     0x8d61
0x8d61:
0x9287: nop     word ptr [rax + rax]
0x9290: mov     byte ptr [rbp], 0x2c
0x9294: je      0x8c38

```

```

0x929a: jmp      0x8d61
0x8d61
0x929f: call     0x49d0
0x92a4: nop      word ptr cs:[rax + rax]
0x92af: nop
0x92b0: endbr64
0x92b4: push     r13
0x92b6: mov      r13, rdx
0x92b9: push     r12
0x92bb: mov      r12, rsi
0x92be: push     rbp
0x92bf: mov      rbp, rdi
0x92c2: call     0x47e0
0x92c7: test     rax, rax
0x92ca: je       0x92d8
0x92cc: pop      rbp
0x92cd: pop      r12
0x92cf: pop      r13
0x92d1: ret
0x92d2: nop      word ptr [rax + rax]
0x92d8: test     r12, r12
0x92db: je       0x9300
0x92dd: mov      rdi, r12
0x92e0: call     0x47e0
0x92e5: mov      rsi, rax
0x92e8: test     rax, rax
0x92eb: je       0x9300
0x92ed: mov      rdi, rbp
0x92f0: pop      rbp
0x92f1: pop      r12
0x92f3: pop      r13
0x92f5: jmp      0x18380
0x18380
0x92fa: nop      word ptr [rax + rax]
0x9300: mov      rsi, r13
0x9303: mov      rdi, rbp
0x9306: pop      rbp
0x9307: pop      r12
0x9309: pop      r13
0x930b: jmp      0x18380
0x18380
0x9310: endbr64
0x9314: push     r15
0x9316: push     r14
0x9318: push     r13
0x931a: push     r12
0x931c: push     rbp
0x931d: push     rbx

```

```

0x931e: mov     ebx, edi
0x9320: sub     rsp, 0xb8
0x9327: mov     qword ptr [rsp + 0x18], rsi
0x932c: mov     rax, qword ptr fs:[0x28]
0x9335: mov     qword ptr [rsp + 0xa8], rax
0x933d: xor     eax, eax
0x933f: mov     dword ptr [rsi], 0
0x9345: cmp     edi, 0x7ff
0x934b: jg      0x95d8
0x9351: lea     rax, [rsp + 0x20]
0x9356: lea     r12, [rsp + 0x50]
0x935b: mov     qword ptr [rsp + 8], rax
0x9360: lea     r13, [rip + 0x2089e]
0x9367: jmp     0x94df
0x94df:
0x936c: nop     dword ptr [rax]
0x9370: mov     rdi, r12
0x9373: call    0xbd80
0x9378: test    rax, rax
0x937b: je      0x9508
0x9381: cmp     byte ptr [rax], 0
0x9384: je      0x94d0
0x938a: mov     edx, 0x50
0x938f: mov     rsi, rax
0x9392: mov     rdi, r12
0x9395: call    0x4cf0
0x939a: mov     dword ptr [rsp + 4], 0
0x93a2: mov     edi, 1
0x93a7: mov     esi, 0x30
0x93ac: call    0x1d740
0x93b1: mov     rdi, qword ptr [rsp + 0x20]
0x93b6: mov     rbp, rax
0x93b9: test    rdi, rdi
0x93bc: je      0x9560
0x93c2: call    0x1d7a0
0x93c7: mov     edi, 8
0x93cc: mov     qword ptr [rbp], rax
0x93d0: call    0x19100
0x93d5: mov     rdi, qword ptr [rsp + 0x30]
0x93da: mov     qword ptr [rbp + 0x28], rax
0x93de: test    rdi, rdi
0x93e1: je      0x93ec
0x93e3: call    0x1d7a0
0x93e8: mov     qword ptr [rbp + 0x10], rax
0x93ec: mov     rdi, qword ptr [rsp + 0x28]
0x93f1: test    rdi, rdi
0x93f4: je      0x9474
0x93f6: call    0x1d7a0

```

```

0x93fb: mov     qword ptr [rbp + 8], rax
0x93ff: mov     r14, qword ptr [rsp + 0x38]
0x9404: mov     eax, 1
0x9409: test    r14, r14
0x940c: je      0x9421
0x940e: call    0x4ef0
0x9413: movsx   rdx, byte ptr [r14]
0x9417: mov     rax, qword ptr [rax]
0x941a: cmp     dword ptr [rax + rdx*4], 0x79
0x941e: sete    al
0x9421: mov     byte ptr [rbp + 0x18], al
0x9424: mov     rdx, qword ptr [rsp + 0x40]
0x9429: test    rdx, rdx
0x942c: je      0x944c
0x942e: mov     qword ptr [rsp + 0x10], rdx
0x9433: call    0x4ef0
0x9438: mov     rdx, qword ptr [rsp + 0x10]
0x943d: mov     rax, qword ptr [rax]
0x9440: movsx   rdx, byte ptr [rdx]
0x9444: cmp     dword ptr [rax + rdx*4], 0x79
0x9448: sete    byte ptr [rbp + 0x19]
0x944c: mov     rdx, qword ptr [rsp + 0x48]
0x9451: test    rdx, rdx
0x9454: je      0x9474
0x9456: mov     qword ptr [rsp + 0x10], rdx
0x945b: call    0x4ef0
0x9460: mov     rdx, qword ptr [rsp + 0x10]
0x9465: mov     rax, qword ptr [rax]
0x9468: movsx   rdx, byte ptr [rdx]
0x946c: cmp     dword ptr [rax + rdx*4], 0x79
0x9470: sete    byte ptr [rbp + 0x1a]
0x9474: call    0x4f00
0x9479: mov     r15, qword ptr [rbp]
0x947d: mov     rax, qword ptr [rax]
0x9480: movsx   rdx, byte ptr [r15]
0x9484: test    byte ptr [rax + rdx*2 + 1], 2
0x9489: jne     0x9518
0x948f: mov     rdi, r12
0x9492: call    0x1d7a0
0x9497: cmp     byte ptr [rbp + 0x1b], 0
0x949b: mov     qword ptr [rbp + 0x20], rax
0x949f: jne     0x94ac
0x94a1: cmp     qword ptr [rbp + 8], 0
0x94a6: je      0x9590
0x94ac: mov     eax, dword ptr [rsp + 4]
0x94b0: test    eax, eax
0x94b2: je      0x94c5
0x94b4: mov     rdi, r12

```

```

0x94b7: call    0x9f00
0x94bc: test    rax, rax
0x94bf: je      0x95a0
0x94c5: mov     rsi, rbp
0x94c8: mov     rdi, r12
0x94cb: call    0xaf70
0x94d0: add     ebx, 1
0x94d3: cmp     ebx, 0x800
0x94d9: je      0x9580
0x94df: mov     rsi, qword ptr [rsp + 8]
0x94e4: xor     ecx, ecx
0x94e6: mov     edx, 6
0x94eb: mov     rdi, r12
0x94ee: call    0x17320
0x94f3: movzx   eax, byte ptr [rsp + 0x50]
0x94f8: test    al, al
0x94fa: je      0x95d8
0x9500: cmp     al, 0x5e
0x9502: jne     0x9370
0x9508: mov     dword ptr [rsp + 4], 1
0x9510: jmp     0x93a2
0x93a2
0x9515: nop     dword ptr [rax]
0x9518: lea     r14, [rip + 0x1fb01]
0x951f: jmp     0x9535
0x9535
0x9521: nop     dword ptr [rax]
0x9528: add     r14, 0x23
0x952c: cmp     r13, r14
0x952f: je      0x948f
0x9535: mov     rsi, r14
0x9538: mov     rdi, r15
0x953b: call    0x4b30
0x9540: test    eax, eax
0x9542: jne     0x9528
0x9544: mov     rdi, r15
0x9547: call    0x18d10
0x954c: mov     qword ptr [rbp], r14
0x9550: mov     byte ptr [rbp + 0x1b], 1
0x9554: jmp     0x948f
0x948f
0x9559: nop     dword ptr [rax]
0x9560: mov     rdi, rax
0x9563: add     ebx, 1
0x9566: call    0x18d10
0x956b: cmp     ebx, 0x800
0x9571: jne     0x94df
0x9577: nop     word ptr [rax + rax]

```

```

0x9580: mov     eax, 0x800
0x9585: jmp     0x95da
0x95da:
0x9587: nop     word ptr [rax + rax]
0x9590: cmp     qword ptr [rbp + 0x10], 0
0x9595: jne     0x94ac
0x959b: jmp     0x94d0
0x94d0:
0x95a0: mov     rcx, qword ptr [rip + 0x21341]
0x95a7: mov     rdi, qword ptr [rip + 0x212d2]
0x95ae: mov     r8, r12
0x95b1: xor     eax, eax
0x95b3: lea     rdx, [rip + 0x1720e]
0x95ba: mov     esi, 1
0x95bf: call    0x4e60
0x95c4: mov     rax, qword ptr [rsp + 0x18]
0x95c9: add     dword ptr [rax], 1
0x95cc: jmp     0x94d0
0x94d0:
0x95d1: nop     dword ptr [rax]
0x95d8: mov     eax, ebx
0x95da: mov     rdx, qword ptr [rsp + 0xa8]
0x95e2: sub     rdx, qword ptr fs:[0x28]
0x95eb: jne     0x95ff
0x95ed: add     rsp, 0xb8
0x95f4: pop     rbx
0x95f5: pop     rbp
0x95f6: pop     r12
0x95f8: pop     r13
0x95fa: pop     r14
0x95fc: pop     r15
0x95fe: ret
0x95ff: call    0x49d0
0x9604: nop     word ptr cs:[rax + rax]
0x960f: nop
0x9610: endbr64
0x9614: sub     rsp, 8
0x9618: xor     edi, edi
0x961a: call    0x9b00
0x961f: xor     eax, eax
0x9621: call    0xa6c0
0x9626: mov     rdi, qword ptr [rip + 0x2129b]
0x962d: add     rsp, 8
0x9631: jmp     0xbb40
0xbb40:
0x9636: nop     word ptr cs:[rax + rax]
0x9640: endbr64
0x9644: push    rbp

```

```

0x9645: lea      rbp, [rip + 0x205b9]
0x964c: push     rbx
0x964d: mov      rdi, rbp
0x9650: sub      rsp, 0x48
0x9654: mov      rax, qword ptr fs:[0x28]
0x965d: mov      qword ptr [rsp + 0x38], rax
0x9662: xor      eax, eax
0x9664: call     0x49b0
0x9669: mov      rdi, qword ptr [rip + 0x21240]
0x9670: mov      rbx, rax
0x9673: call     0x11100
0x9678: mov      edx, 0x33
0x967d: mov      rsi, rsp
0x9680: sub      edx, ebx
0x9682: mov      rdi, rax
0x9685: call     0x182b0
0x968a: mov      rsi, rbp
0x968d: mov      rdi, rax
0x9690: call     0x4df0
0x9695: mov      rdi, rax
0x9698: call     0xd540
0x969d: mov      rax, qword ptr [rsp + 0x38]
0x96a2: sub      rax, qword ptr fs:[0x28]
0x96ab: jne      0x96b4
0x96ad: add      rsp, 0x48
0x96b1: pop      rbx
0x96b2: pop      rbp
0x96b3: ret
0x96b4: call     0x49d0
0x96b9: nop      dword ptr [rax]
0x96c0: endbr64
0x96c4: sub      rsp, 8
0x96c8: mov      r8d, 1
0x96ce: mov      rdx, qword ptr [rip + 0x211db]
0x96d5: mov      rdi, qword ptr [rip + 0x211cc]
0x96dc: mov      rcx, -1
0x96e3: mov      esi, 0x22
0x96e8: call     0x13e30
0x96ed: mov      r8d, 1
0x96f3: mov      rdx, qword ptr [rip + 0x211ae]
0x96fa: mov      rdi, qword ptr [rip + 0x211af]
0x9701: mov      rcx, -1
0x9708: mov      esi, 0x22
0x970d: add      rsp, 8
0x9711: jmp      0x13e30
0x13e30:
0x9716: nop      word ptr cs:[rax + rax]
0x9720: endbr64

```



```

0x9724: sub    rsp, 8
0x9728: mov     rdx, qword ptr [rip + 0x211b9]
0x972f: mov     edi, 1
0x9734: xor     eax, eax
0x9736: lea     rsi, [rip + 0x170b3]
0x973d: call    0x4d20
0x9742: lea     rdi, [rip + 0x170cf]
0x9749: call    0x48d0
0x974e: lea     rdi, [rip + 0x170eb]
0x9755: call    0x48d0
0x975a: lea     rdi, [rip + 0x16dde]
0x9761: call    0x48d0
0x9766: lea     rdi, [rip + 0x170fb]
0x976d: call    0x48d0
0x9772: lea     rdi, [rip + 0x17117]
0x9779: call    0x48d0
0x977e: lea     rdi, [rip + 0x1713b]
0x9785: add     rsp, 8
0x9789: jmp     0x48d0
0x48d0
0x978e: nop
0x9790: cmp     dword ptr [rip + 0x21ba9], 1
0x9797: push    rbx
0x9798: mov     ebx, edi
0x979a: je      0x9830
0x97a0: cmp     ebx, 1
0x97a3: je      0x984a
0x97a9: mov     rdi, qword ptr [rip + 0x20598]
0x97b0: test    rdi, rdi
0x97b3: je      0x97c6
0x97b5: lea     rdx, [rip + 0x2b4]
0x97bc: mov     esi, 1
0x97c1: call    0x4820
0x97c6: movzxb  edx, byte ptr [rip + 0x21b83]
0x97cd: mov     dword ptr [rip + 0x21b75], 0
0x97d7: mov     dword ptr [rip + 0x21b67], 0
0x97e1: mov     eax, edx
0x97e3: and     eax, 0x7f
0x97e6: and     edx, 0x40
0x97e9: mov     byte ptr [rip + 0x21b61], al
0x97ef: jne     0x9810
0x97f1: mov     dword ptr [rip + 0x21b45], 1
0x97fb: and     eax, 0x7f
0x97fe: shl     ebx, 7
0x9801: or      eax, ebx
0x9803: mov     byte ptr [rip + 0x21b47], al
0x9809: pop     rbx
0x980a: ret

```

```

0x980b: nop      dword ptr [rax + rax]
0x9810: mov      dword ptr [rip + 0x21b2a], 0
0x981a: mov      edi, 1
0x981f: call     0x9870
0x9824: movzx    eax, byte ptr [rip + 0x21b25]
0x982b: jmp      0x97f1
0x97f1
0x982d: nop      dword ptr [rax]
0x9830: movzx    eax, byte ptr [rip + 0x21b19]
0x9837: shr      al, 7
0x983a: movzx    eax, al
0x983d: cmp      eax, edi
0x983f: je       0x9809
0x9841: cmp      ebx, 1
0x9844: jne      0x97a9
0x984a: mov      rdi, qword ptr [rip + 0x20517]
0x9851: test     rdi, rdi
0x9854: je       0x9824
0x9856: lea      rdx, [rip + 0x213]
0x985d: mov      esi, 1
0x9862: call     0x4820
0x9867: movzx    eax, byte ptr [rip + 0x21ae2]
0x986e: jmp      0x97f1
0x97f1
0x9870: cmp      dword ptr [rip + 0x21acd], 1
0x9877: push     rbx
0x9878: mov      ebx, edi
0x987a: je       0x98e0
0x987c: cmp      ebx, 1
0x987f: je       0x98f6
0x9881: mov      rdi, qword ptr [rip + 0x204c0]
0x9888: test     rdi, rdi
0x988b: je       0x989e
0x988d: lea      rdx, [rip + 0x1dc]
0x9894: mov      esi, 1
0x9899: call     0x4820
0x989e: movzx    eax, byte ptr [rip + 0x21aab]
0x98a5: mov      dword ptr [rip + 0x21a9d], 0
0x98af: mov      dword ptr [rip + 0x21a8f], 0
0x98b9: and      eax, 0xffffffffbf
0x98bc: mov      byte ptr [rip + 0x21a8e], al
0x98c2: test     al, al
0x98c4: js       0x9920
0x98c6: mov      dword ptr [rip + 0x21a74], 1
0x98d0: and      eax, 0xffffffffbf
0x98d3: shl      ebx, 6
0x98d6: or       eax, ebx
0x98d8: mov      byte ptr [rip + 0x21a72], al

```

```

0x98de: pop      rbx
0x98df: ret
0x98e0: movzx   eax, byte ptr [rip + 0x21a69]
0x98e7: sar      eax, 6
0x98ea: and      eax, 1
0x98ed: cmp      eax, edi
0x98ef: je       0x98de
0x98f1: cmp      ebx, 1
0x98f4: jne      0x9881
0x98f6: mov      rdi, qword ptr [rip + 0x2048b]
0x98fd: test     rdi, rdi
0x9900: je       0x9934
0x9902: lea      rdx, [rip + 0x167]
0x9909: mov      esi, 1
0x990e: call     0x4820
0x9913: movzx   eax, byte ptr [rip + 0x21a36]
0x991a: jmp      0x98c6
0x98c6
0x991c: nop      dword ptr [rax]
0x9920: mov      dword ptr [rip + 0x21a16], 0
0x992a: mov      edi, 1
0x992f: call     0x9790
0x9934: movzx   eax, byte ptr [rip + 0x21a15]
0x993b: jmp      0x98c6
0x98c6
0x993d: nop      dword ptr [rax]
0x9940: push     r12
0x9942: mov      edx, 5
0x9947: lea      rsi, [rip + 0x17293]
0x994e: mov      r12d, 1
0x9954: push     rbp
0x9955: mov      rbp, rdi
0x9958: sub      rsp, 8
0x995c: call     0x4890
0x9961: test     eax, eax
0x9963: je       0x99d3
0x9965: mov      edx, 4
0x996a: lea      rsi, [rip + 0x1725f]
0x9971: mov      rdi, rbp
0x9974: call     0x4890
0x9979: test     eax, eax
0x997b: je       0x99d3
0x997d: mov      edx, 9
0x9982: lea      rsi, [rip + 0x1724c]
0x9989: mov      rdi, rbp
0x998c: call     0x4890
0x9991: test     eax, eax
0x9993: je       0x99d3

```

```

0x9995: lea    rsi, [rip + 0x17243]
0x999c: mov     rdi, rbp
0x999f: call    0x4b30
0x99a4: test    eax, eax
0x99a6: je      0x99d3
0x99a8: lea     rsi, [rip + 0x17238]
0x99af: mov     rdi, rbp
0x99b2: call    0x4b30
0x99b7: test    eax, eax
0x99b9: je      0x99d3
0x99bb: lea     rsi, [rip + 0x1722b]
0x99c2: mov     rdi, rbp
0x99c5: xor     r12d, r12d
0x99c8: call    0x4b30
0x99cd: test    eax, eax
0x99cf: sete    r12b
0x99d3: add     rsp, 8
0x99d7: mov     eax, r12d
0x99da: pop     rbp
0x99db: pop     r12
0x99dd: ret
0x99de: nop
0x99e0: push    r13
0x99e2: push    r12
0x99e4: mov     r12, qword ptr [rip + 0x20f8d]
0x99eb: push    rbp
0x99ec: test    r12, r12
0x99ef: je      0x9a58
0x99f1: mov     r13, rdi
0x99f4: jmp     0x9a13
0x9a13:
0x99f6: nop     word ptr cs:[rax + rax]
0x9a00: js      0x9a30
0x9a02: mov     r12, qword ptr [r12 + 8]
0x9a07: mov     qword ptr [rip + 0x20f6a], r12
0x9a0e: test    r12, r12
0x9a11: je      0x9a58
0x9a13: mov     rbp, qword ptr [r12]
0x9a17: mov     rdi, r13
0x9a1a: mov     rsi, rbp
0x9a1d: call    0x4b30
0x9a22: test    eax, eax
0x9a24: jne     0x9a00
0x9a26: mov     rax, r12
0x9a29: pop     rbp
0x9a2a: pop     r12
0x9a2c: pop     r13
0x9a2e: ret

```

```

0x9a2f: nop
0x9a30: mov     rdi, r13
0x9a33: call    0x49b0
0x9a38: mov     rsi, rbp
0x9a3b: mov     rdi, r13
0x9a3e: mov     rdx, rax
0x9a41: call    0x4890
0x9a46: pop     rbp
0x9a47: neg     eax
0x9a49: sbb     r12, r12
0x9a4c: mov     rax, r12
0x9a4f: pop     r12
0x9a51: pop     r13
0x9a53: ret
0x9a54: nop     dword ptr [rax]
0x9a58: mov     r12, -1
0x9a5f: pop     rbp
0x9a60: mov     rax, r12
0x9a63: pop     r12
0x9a65: pop     r13
0x9a67: ret
0x9a68: nop     dword ptr [rax + rax]
0x9a70: endbr64
0x9a74: movsxd  rdx, dword ptr [rip + 0x2148d]
0x9a7b: push    r12
0x9a7d: push    rbp
0x9a7e: mov     ebp, edi
0x9a80: push    rbx
0x9a81: cmp     edx, 0x400
0x9a87: je      0x9ab0
0x9a89: lea     eax, [rdx + 1]
0x9a8c: lea     r12, [rip + 0x2148d]
0x9a93: mov     byte ptr [r12 + rdx], bpl
0x9a97: pop     rbx
0x9a98: mov     dword ptr [rip + 0x2146a], eax
0x9a9e: pop     rbp
0x9a9f: mov     eax, 1
0x9aa4: pop     r12
0x9aa6: ret
0x9aa7: nop     word ptr [rax + rax]
0x9ab0: xor     ebx, ebx
0x9ab2: lea     r12, [rip + 0x21467]
0x9ab9: jmp     0x9acc
0x9acc:
0x9abb: nop     dword ptr [rax + rax]
0x9ac0: mov     edx, dword ptr [rip + 0x21442]
0x9ac6: add     ebx, eax
0x9ac8: cmp     edx, ebx

```

```

0x9aca: jle      0x9ae5
0x9acc: sub      edx, ebx
0x9ace: movsxd    rsi, ebx
0x9ad1: mov      edi, 1
0x9ad6: add      rsi, r12
0x9ad9: movsxd    rdx, edx
0x9adc: call     0x18da0
0x9ae1: test     eax, eax
0x9ae3: jns      0x9ac0
0x9ae5: mov      eax, 1
0x9aea: xor      edx, edx
0x9aec: jmp      0x9a93
0x9a93:
0x9aee: nop
0x9af0: endbr64
0x9af4: mov      dword ptr [rip + 0x21862], edi
0x9afa: ret
0x9afb: nop      dword ptr [rax + rax]
0x9b00: endbr64
0x9b04: push     r12
0x9b06: push     rbp
0x9b07: push     rbx
0x9b08: mov      ebx, edi
0x9b0a: sub      rsp, 0x50
0x9b0e: mov      rax, qword ptr fs:[0x28]
0x9b17: mov      qword ptr [rsp + 0x48], rax
0x9b1c: xor      eax, eax
0x9b1e: cmp      edi, 1
0x9b21: je       0x9b60
0x9b23: lea      rdx, [rip + 0x21896]
0x9b2a: mov      esi, 1
0x9b2f: mov      edi, 1
0x9b34: call     0x4d70
0x9b39: mov      dword ptr [rip + 0x20e19], ebx
0x9b3f: mov      rax, qword ptr [rsp + 0x48]
0x9b44: sub      rax, qword ptr fs:[0x28]
0x9b4d: jne      0x9c7e
0x9b53: add      rsp, 0x50
0x9b57: pop      rbx
0x9b58: pop      rbp
0x9b59: pop      r12
0x9b5b: ret
0x9b5c: nop      dword ptr [rax]
0x9b60: mov      rax, qword ptr [rip + 0x21889]
0x9b67: mov      rdx, qword ptr [rip + 0x21852]
0x9b6e: lea      r12, [rip + 0x2180b]
0x9b75: mov      esi, 1
0x9b7a: movdqa   xmm0, xmmword ptr [rip + 0x2183e]

```

```

0x9b82: movdqa xmm1, xmmword ptr [rip + 0x21846]
0x9b8a: mov     edi, 1
0x9b8f: mov     qword ptr [rip + 0x2181a], rax
0x9b96: mov     eax, dword ptr [rip + 0x2185c]
0x9b9c: movaps  xmmword ptr [rip + 0x217dd], xmm0
0x9ba3: movdqa  xmm2, xmmword ptr [rip + 0x21835]
0x9bab: mov     dword ptr [rip + 0x21807], eax
0x9bb1: movabs  rax, 0xffffffffffffffa3c
0x9bbb: and     rax, rdx
0x9bbe: mov     rdx, r12
0x9bc1: movaps  xmmword ptr [rip + 0x217c8], xmm1
0x9bc8: mov     qword ptr [rip + 0x217b1], rax
0x9bcf: mov     eax, dword ptr [rip + 0x217b7]
0x9bd5: movaps  xmmword ptr [rip + 0x217c4], xmm2
0x9bdc: and     eax, 0xffffffff5
0x9bdf: mov     byte ptr [rip + 0x217bb], 0
0x9be6: mov     dword ptr [rip + 0x217a1], 7
0x9bf0: or      al, 0x81
0x9bf2: mov     dword ptr [rip + 0x21794], eax
0x9bf8: mov     eax, 0x100
0x9bfd: mov     word ptr [rip + 0x21792], ax
0x9c04: mov     qword ptr [rip + 0x2178a], 0x1a0000
0x9c0f: call    0x4d70
0x9c14: mov     rdi, r12
0x9c17: mov     r12, rsp
0x9c1a: call    0x4d10
0x9c1f: mov     edx, 1
0x9c24: mov     esi, 0x540a
0x9c29: mov     edi, 1
0x9c2e: mov     dword ptr [rip + 0x20cfc], eax
0x9c34: xor     eax, eax
0x9c36: call    0x4a70
0x9c3b: mov     esi, 1
0x9c40: mov     edi, 1
0x9c45: call    0x4db0
0x9c4a: mov     ebp, dword ptr [rip + 0x200d4]
0x9c50: mov     rsi, r12
0x9c53: mov     edi, 1
0x9c58: call    0x4d60
0x9c5d: mov     rdx, r12
0x9c60: mov     esi, 1
0x9c65: mov     edi, 1
0x9c6a: mov     byte ptr [rsp + 0x11], bpl
0x9c6f: mov     byte ptr [rsp + 0x12], 0
0x9c74: call    0x4d70
0x9c79: jmp     0x9b39
0x9b39
0x9c7e: call    0x49d0

```

```

0x9c83: nop      word ptr cs:[rax + rax]
0x9c8e: nop
0x9c90: endbr64
0x9c94: mov      eax, dword ptr [rip + 0x20cbe]
0x9c9a: ret
0x9c9b: nop      dword ptr [rax + rax]
0x9ca0: endbr64
0x9ca4: push     r12
0x9ca6: push     rbx
0x9ca7: mov      ebx, edi
0x9ca9: mov      edi, 1
0x9cae: sub      rsp, 0x58
0x9cb2: mov      rax, qword ptr fs:[0x28]
0x9cbb: mov      qword ptr [rsp + 0x48], rax
0x9cc0: xor      eax, eax
0x9cc2: mov      r12, rsp
0x9cc5: mov      rsi, r12
0x9cc8: call     0x4d60
0x9ccd: mov      rdx, r12
0x9cd0: mov      esi, 1
0x9cd5: mov      edi, 1
0x9cda: mov      byte ptr [rsp + 0x11], bl
0x9cde: mov      byte ptr [rsp + 0x12], 0
0x9ce3: call     0x4d70
0x9ce8: mov      dword ptr [rip + 0x20036], ebx
0x9cee: mov      rax, qword ptr [rsp + 0x48]
0x9cf3: sub      rax, qword ptr fs:[0x28]
0x9cfc: jne      0x9d06
0x9cfe: add      rsp, 0x58
0x9d02: pop      rbx
0x9d03: pop      r12
0x9d05: ret
0x9d06: call     0x49d0
0x9d0b: nop      dword ptr [rax + rax]
0x9d10: endbr64
0x9d14: mov      edx, dword ptr [rip + 0x211ee]
0x9d1a: test     edx, edx
0x9d1c: jle      0x9d66
0x9d1e: push     rbp
0x9d1f: xor      ebp, ebp
0x9d21: push     rbx
0x9d22: lea      rbx, [rip + 0x211f7]
0x9d29: sub      rsp, 8
0x9d2d: jmp      0x9d3c
0x9d3c:
0x9d2f: nop
0x9d30: mov      edx, dword ptr [rip + 0x211d2]
0x9d36: add      ebp, eax

```



```

0x9d38: cmp     edx, ebp
0x9d3a: jle     0x9d55
0x9d3c: sub     edx, ebp
0x9d3e: movsxd  rsi, ebp
0x9d41: mov     edi, 1
0x9d46: add     rsi, rbx
0x9d49: movsxd  rdx, edx
0x9d4c: call    0x18da0
0x9d51: test    eax, eax
0x9d53: jns     0x9d30
0x9d55: mov     dword ptr [rip + 0x211a9], 0
0x9d5f: add     rsp, 8
0x9d63: pop     rbx
0x9d64: pop     rbp
0x9d65: ret
0x9d66: mov     dword ptr [rip + 0x21198], 0
0x9d70: ret
0x9d71: nop     word ptr cs:[rax + rax]
0x9d7c: nop     dword ptr [rax]
0x9d80: endbr64
0x9d84: push    rbp
0x9d85: lea     rdx, [rip - 0x31c]
0x9d8c: push    rbx
0x9d8d: sub     rsp, 8
0x9d91: mov     esi, dword ptr [rip + 0x20c05]
0x9d97: mov     rdi, qword ptr [rip + 0x2004a]
0x9d9e: call    0x4820
0x9da3: mov     edx, dword ptr [rip + 0x2115f]
0x9da9: test    edx, edx
0x9dab: jle     0x9de5
0x9dad: xor     ebp, ebp
0x9daf: lea     rbx, [rip + 0x2116a]
0x9db6: jmp     0x9dcc
0x9dcc:
0x9db8: nop     dword ptr [rax + rax]
0x9dc0: mov     edx, dword ptr [rip + 0x21142]
0x9dc6: add     ebp, eax
0x9dc8: cmp     edx, ebp
0x9dca: jle     0x9de5
0x9dcc: sub     edx, ebp
0x9dce: movsxd  rsi, ebp
0x9dd1: mov     edi, 1
0x9dd6: add     rsi, rbx
0x9dd9: movsxd  rdx, edx
0x9ddc: call    0x18da0
0x9de1: test    eax, eax
0x9de3: jns     0x9dc0
0x9de5: mov     dword ptr [rip + 0x21119], 0

```

```

0x9def: add    rsp, 8
0x9df3: pop     rbx
0x9df4: pop     rbp
0x9df5: ret
0x9df6: nop     word ptr cs:[rax + rax]
0x9e00: endbr64
0x9e04: push    rbp
0x9e05: lea     rdx, [rip - 0x39c]
0x9e0c: push    rbx
0x9e0d: sub     rsp, 8
0x9e11: mov     eax, dword ptr [rip + 0x20b85]
0x9e17: mov     rdi, qword ptr [rip + 0x2010a]
0x9e1e: lea     esi, [rax - 1]
0x9e21: call    0x4820
0x9e26: mov     edx, dword ptr [rip + 0x210dc]
0x9e2c: test    edx, edx
0x9e2e: jle     0x9e65
0x9e30: xor     ebp, ebp
0x9e32: lea     rbx, [rip + 0x210e7]
0x9e39: jmp     0x9e4c
0x9e4c:
0x9e3b: nop     dword ptr [rax + rax]
0x9e40: mov     edx, dword ptr [rip + 0x210c2]
0x9e46: add     ebp, eax
0x9e48: cmp     edx, ebp
0x9e4a: jle     0x9e65
0x9e4c: sub     edx, ebp
0x9e4e: movsxd  rsi, ebp
0x9e51: mov     edi, 1
0x9e56: add     rsi, rbx
0x9e59: movsxd  rdx, edx
0x9e5c: call    0x18da0
0x9e61: test    eax, eax
0x9e63: jns     0x9e40
0x9e65: mov     dword ptr [rip + 0x21099], 0
0x9e6f: add     rsp, 8
0x9e73: pop     rbx
0x9e74: pop     rbp
0x9e75: ret
0x9e76: nop     word ptr cs:[rax + rax]
0x9e80: endbr64
0x9e84: push    rbp
0x9e85: lea     rdx, [rip - 0x41c]
0x9e8c: push    rbx
0x9e8d: sub     rsp, 8
0x9e91: mov     eax, dword ptr [rip + 0x20b05]
0x9e97: mov     rdi, qword ptr [rip + 0x200aa]
0x9e9e: lea     esi, [rax - 1]

```

```

0x9ea1: call    0x4820
0x9ea6: mov     edx, dword ptr [rip + 0x2105c]
0x9eac: test    edx, edx
0x9eae: jle     0x9ee5
0x9eb0: xor     ebp, ebp
0x9eb2: lea     rbx, [rip + 0x21067]
0x9eb9: jmp     0x9ecc
0x9ecc:
0x9ebb: nop     dword ptr [rax + rax]
0x9ec0: mov     edx, dword ptr [rip + 0x21042]
0x9ec6: add     ebp, eax
0x9ec8: cmp     edx, ebp
0x9eca: jle     0x9ee5
0x9ecc: sub     edx, ebp
0x9ece: movsxd  rsi, ebp
0x9ed1: mov     edi, 1
0x9ed6: add     rsi, rbx
0x9ed9: movsxd  rdx, edx
0x9edc: call    0x18da0
0x9ee1: test    eax, eax
0x9ee3: jns     0x9ec0
0x9ee5: mov     dword ptr [rip + 0x21019], 0
0x9eef: add     rsp, 8
0x9ef3: pop     rbx
0x9ef4: pop     rbp
0x9ef5: ret
0x9ef6: nop     word ptr cs:[rax + rax]
0x9f00: endbr64
0x9f04: mov     edx, dword ptr [rip + 0x214f2]
0x9f0a: movzx   eax, byte ptr [rdi]
0x9f0d: push    r14
0x9f0f: push    r13
0x9f11: mov     r13, rdi
0x9f14: push    r12
0x9f16: push    rbp
0x9f17: push    rbx
0x9f18: test    edx, edx
0x9f1a: jne     0x9fc8
0x9f20: cmp     al, 0x5e
0x9f22: jne     0x9fb8
0x9f28: lea     rbx, [r13 + 1]
0x9f2c: mov     rbp, r13
0x9f2f: lea     r14, [rip + 0x1fd6a]
0x9f36: jmp     0x9f56
0x9f56:
0x9f38: nop     dword ptr [rax + rax]
0x9f40: add     rbp, 1
0x9f44: mov     byte ptr [rbx - 1], al

```

```

0x9f47: mov     rdx, rbx
0x9f4a: add     rbx, 1
0x9f4e: movzx   eax, byte ptr [rbp]
0x9f52: test    al, al
0x9f54: je      0x9fa6
0x9f56: cmp     al, 0x5e
0x9f58: jne     0x9f40
0x9f5a: movzx   r12d, byte ptr [rbp + 1]
0x9f5f: test    r12b, r12b
0x9f62: je      0x9fb8
0x9f64: call    0x47c0
0x9f69: mov     r8, rax
0x9f6c: movzx   eax, r12b
0x9f70: mov     rdx, qword ptr [r8]
0x9f73: mov     eax, dword ptr [rdx + rax*4]
0x9f76: cmp     eax, 0x47
0x9f79: je      0x9fb8
0x9f7b: cmp     eax, 0x5a
0x9f7e: je      0x9fb8
0x9f80: and     r12d, 0x7f
0x9f84: add     rbp, 2
0x9f88: mov     rdx, rbx
0x9f8b: add     rbx, 1
0x9f8f: sub     r12d, 0x20
0x9f93: movsxd  r12, r12d
0x9f96: movzx   eax, byte ptr [r14 + r12]
0x9f9b: mov     byte ptr [rbx - 2], al
0x9f9e: movzx   eax, byte ptr [rbp]
0x9fa2: test    al, al
0x9fa4: jne     0x9f56
0x9fa6: mov     byte ptr [rdx], 0
0x9fa9: mov     rax, r13
0x9fac: pop     rbx
0x9fad: pop     rbp
0x9fae: pop     r12
0x9fb0: pop     r13
0x9fb2: pop     r14
0x9fb4: ret
0x9fb5: nop     dword ptr [rax]
0x9fb8: pop     rbx
0x9fb9: xor     eax, eax
0x9fbb: pop     rbp
0x9fbc: pop     r12
0x9fbe: pop     r13
0x9fc0: pop     r14
0x9fc2: ret
0x9fc3: nop     dword ptr [rax + rax]
0x9fc8: test    al, al

```

```

0x9fca: jne      0x9f28
0x9fd0: mov      rdx, rdi
0x9fd3: jmp      0x9fa6
0x9fa6:
0x9fd5: nop      word ptr cs:[rax + rax]
0x9fe0: endbr64
0x9fe4: push     r15
0x9fe6: push     r14
0x9fe8: mov      r14, rdi
0x9feb: push     r13
0x9fed: lea      r13, [rip + 0x16c12]
0x9ff4: push     r12
0x9ff6: lea      r12, [rip + 0x209c3]
0x9ffd: push     rbp
0x9ffe: lea      rbp, [rip + 0x16bef]
0xa005: push     rbx
0xa006: mov      rbx, rdi
0xa009: sub      rsp, 0x18
0xa00d: mov      rax, qword ptr fs:[0x28]
0xa016: mov      qword ptr [rsp + 8], rax
0xa01b: xor      eax, eax
0xa01d: mov      byte ptr [rip + 0x2099c], 0
0xa024: movzx    r15d, byte ptr [rdi]
0xa028: test     r15b, r15b
0xa02b: jne      0xa0c2
0xa031: jmp      0xa0f0
0xa0f0:
0xa036: nop      word ptr cs:[rax + rax]
0xa040: cmp      r15b, 0x20
0xa044: je       0xa148
0xa04a: cmp      r15b, 0x7f
0xa04e: je       0xa160
0xa054: cmp      r15b, 0xff
0xa058: je       0xa180
0xa05e: call     0x4f00
0xa063: mov      r8, rax
0xa066: movzx    eax, r15b
0xa06a: mov      rsi, qword ptr [r8]
0xa06d: test     byte ptr [rsi + rax*2], 2
0xa071: je       0xa120
0xa077: lea      edx, [r15 + 0x40]
0xa07b: lea      rsi, [rsp + 5]
0xa080: mov      rdi, r12
0xa083: mov      byte ptr [rsp + 5], 0x5e
0xa088: mov      byte ptr [rsp + 6], dl
0xa08c: mov      edx, 0x12c
0xa091: mov      byte ptr [rsp + 7], 0
0xa096: call     0x47d0

```

```

0xa09b: add    r14, 1
0xa09f: cmp    byte ptr [r14], 0
0xa0a3: je     0xa0f0
0xa0a5: cmp    rbx, r14
0xa0a8: je     0xa0be
0xa0aa: mov    edx, 0x12c
0xa0af: lea    rsi, [rip + 0x1668f]
0xa0b6: mov    rdi, r12
0xa0b9: call   0x47d0
0xa0be: movzx  r15d, byte ptr [r14]
0xa0c2: cmp    r15b, 0x1b
0xa0c6: jne    0xa040
0xa0cc: mov    edx, 0x12c
0xa0d1: mov    rsi, rbp
0xa0d4: mov    rdi, r12
0xa0d7: add    r14, 1
0xa0db: call   0x47d0
0xa0e0: cmp    byte ptr [r14], 0
0xa0e4: jne    0xa0a5
0xa0e6: nop    word ptr cs:[rax + rax]
0xa0f0: mov    rax, qword ptr [rsp + 8]
0xa0f5: sub    rax, qword ptr fs:[0x28]
0xa0fe: jne    0xa199
0xa104: add    rsp, 0x18
0xa108: mov    rax, r12
0xa10b: pop    rbx
0xa10c: pop    rbp
0xa10d: pop    r12
0xa10f: pop    r13
0xa111: pop    r14
0xa113: pop    r15
0xa115: ret
0xa116: nop    word ptr cs:[rax + rax]
0xa120: lea    rsi, [rsp + 5]
0xa125: mov    edx, 0x12c
0xa12a: mov    rdi, r12
0xa12d: mov    byte ptr [rsp + 5], r15b
0xa132: mov    byte ptr [rsp + 6], 0
0xa137: call   0x47d0
0xa13c: jmp    0xa09b
0xa09b:
0xa141: nop    dword ptr [rax]
0xa148: mov    edx, 0x12c
0xa14d: mov    rsi, r13
0xa150: mov    rdi, r12
0xa153: call   0x47d0
0xa158: jmp    0xa09b
0xa09b:

```

```

0xa15d: nop      dword ptr [rax]
0xa160: mov      edx, 0x12c
0xa165: lea      rsi, [rip + 0x16a8f]
0xa16c: mov      rdi, r12
0xa16f: call     0x47d0
0xa174: jmp      0xa09b
0xa09b
0xa179: nop      dword ptr [rax]
0xa180: mov      edx, 0x12c
0xa185: lea      rsi, [rip + 0x16a79]
0xa18c: mov      rdi, r12
0xa18f: call     0x47d0
0xa194: jmp      0xa09b
0xa09b
0xa199: call     0x49d0
0xa19e: nop
0xa1a0: endbr64
0xa1a4: mov      eax, dword ptr [rip + 0x211ae]
0xa1aa: mov      r8, rdi
0xa1ad: mov      edi, dword ptr [rip + 0x207e5]
0xa1b3: push     r13
0xa1b5: push     r12
0xa1b7: xor      r12d, r12d
0xa1ba: lea      edx, [rax + rsi]
0xa1bd: push     rbx
0xa1be: mov      dword ptr [rip + 0x21194], edx
0xa1c4: cmp      edi, eax
0xa1c6: jle      0xa21c
0xa1c8: mov      ecx, dword ptr [rip + 0x21186]
0xa1ce: cmp      ecx, dword ptr [rip + 0x207c8]
0xa1d4: jge      0xa21c
0xa1d6: mov      r12d, edi
0xa1d9: sub      r12d, eax
0xa1dc: cmp      edx, edi
0xa1de: cmovle   r12d, esi
0xa1e2: imul     edi, ecx
0xa1e5: mov      rsi, r8
0xa1e8: movsxd   r13, r12d
0xa1eb: lea      ebx, [rdi + rax]
0xa1ee: mov      rdi, qword ptr [rip + 0x21143]
0xa1f5: mov      rdx, r13
0xa1f8: movsxd   rbx, ebx
0xa1fb: add      rdi, rbx
0xa1fe: call     0x4bb0
0xa203: add      rbx, qword ptr [rip + 0x21126]
0xa20a: movzx    esi, byte ptr [rip + 0x21140]
0xa211: mov      rdx, r13
0xa214: mov      rdi, rbx

```

```

0xa217: call    0x4a50
0xa21c: mov     eax, r12d
0xa21f: pop     rbx
0xa220: pop     r12
0xa222: pop     r13
0xa224: ret
0xa225: nop     word ptr cs:[rax + rax]
0xa230: endbr64
0xa234: push    r13
0xa236: push    r12
0xa238: xor     r12d, r12d
0xa23b: push    rbx
0xa23c: sub     rsp, 0x10
0xa240: mov     rax, qword ptr fs:[0x28]
0xa249: mov     qword ptr [rsp + 8], rax
0xa24e: xor     eax, eax
0xa250: mov     eax, dword ptr [rip + 0x21102]
0xa256: mov     byte ptr [rsp + 7], dil
0xa25b: mov     edi, dword ptr [rip + 0x20737]
0xa261: lea     edx, [rax + 1]
0xa264: mov     dword ptr [rip + 0x210ee], edx
0xa26a: cmp     eax, edi
0xa26c: jge     0xa2c2
0xa26e: mov     ecx, dword ptr [rip + 0x210e0]
0xa274: cmp     ecx, dword ptr [rip + 0x20722]
0xa27a: jge     0xa2c2
0xa27c: cmp     edx, edi
0xa27e: jle     0xa2e0
0xa280: mov     r12d, edi
0xa283: sub     r12d, eax
0xa286: movsxd  r13, r12d
0xa289: imul    edi, ecx
0xa28c: lea     rsi, [rsp + 7]
0xa291: mov     rdx, r13
0xa294: lea     ebx, [rdi + rax]
0xa297: mov     rdi, qword ptr [rip + 0x2109a]
0xa29e: movsxd  rbx, ebx
0xa2a1: add     rdi, rbx
0xa2a4: call    0x4bb0
0xa2a9: add     rbx, qword ptr [rip + 0x21080]
0xa2b0: movzx   esi, byte ptr [rip + 0x2109a]
0xa2b7: mov     rdx, r13
0xa2ba: mov     rdi, rbx
0xa2bd: call    0x4a50
0xa2c2: mov     rax, qword ptr [rsp + 8]
0xa2c7: sub     rax, qword ptr fs:[0x28]
0xa2d0: jne     0xa2ee
0xa2d2: add     rsp, 0x10

```



```

0xa2d6: mov     eax, r12d
0xa2d9: pop     rbx
0xa2da: pop     r12
0xa2dc: pop     r13
0xa2de: ret
0xa2df: nop
0xa2e0: mov     r13d, 1
0xa2e6: mov     r12d, 1
0xa2ec: jmp     0xa289
0xa289:
0xa2ee: call    0x49d0
0xa2f3: nop     word ptr cs:[rax + rax]
0xa2fe: nop
0xa300: endbr64
0xa304: push    rbp
0xa305: lea     rdx, [rip - 0x89c]
0xa30c: push    rbx
0xa30d: sub     rsp, 8
0xa311: mov     esi, dword ptr [rip + 0x20685]
0xa317: mov     rdi, qword ptr [rip + 0x1faca]
0xa31e: call    0x4820
0xa323: mov     edx, dword ptr [rip + 0x20bdf]
0xa329: test    edx, edx
0xa32b: jle     0xa365
0xa32d: xor     ebp, ebp
0xa32f: lea     rbx, [rip + 0x20bea]
0xa336: jmp     0xa34c
0xa34c:
0xa338: nop     dword ptr [rax + rax]
0xa340: mov     edx, dword ptr [rip + 0x20bc2]
0xa346: add     ebp, eax
0xa348: cmp     edx, ebp
0xa34a: jle     0xa365
0xa34c: sub     edx, ebp
0xa34e: movsxd  rsi, ebp
0xa351: mov     edi, 1
0xa356: add     rsi, rbx
0xa359: movsxd  rdx, edx
0xa35c: call    0x18da0
0xa361: test    eax, eax
0xa363: jns     0xa340
0xa365: mov     edx, dword ptr [rip + 0x20631]
0xa36b: xor     esi, esi
0xa36d: imul    edx, dword ptr [rip + 0x20624]
0xa374: mov     dword ptr [rip + 0x20b8a], 0
0xa37e: mov     rdi, qword ptr [rip + 0x20fb3]
0xa385: movsxd  rdx, edx
0xa388: call    0x4a50

```

```

0xa38d: mov     edx, dword ptr [rip + 0x20609]
0xa393: imul    edx, dword ptr [rip + 0x205fe]
0xa39a: xor     esi, esi
0xa39c: mov     rdi, qword ptr [rip + 0x20f8d]
0xa3a3: movsxd  rdx, edx
0xa3a6: call    0x4a50
0xa3ab: mov     edx, dword ptr [rip + 0x205eb]
0xa3b1: imul    edx, dword ptr [rip + 0x205e0]
0xa3b8: xor     esi, esi
0xa3ba: mov     rdi, qword ptr [rip + 0x20f67]
0xa3c1: movsxd  rdx, edx
0xa3c4: call    0x4a50
0xa3c9: mov     edx, dword ptr [rip + 0x205cd]
0xa3cf: imul    edx, dword ptr [rip + 0x205c2]
0xa3d6: xor     esi, esi
0xa3d8: mov     rdi, qword ptr [rip + 0x20f41]
0xa3df: movsxd  rdx, edx
0xa3e2: call    0x4a50
0xa3e7: mov     dword ptr [rip + 0x20f67], 0
0xa3f1: mov     dword ptr [rip + 0x20f59], 0
0xa3fb: add     rsp, 8
0xa3ff: pop     rbx
0xa400: pop     rbp
0xa401: ret
0xa402: nop     word ptr cs:[rax + rax]
0xa40d: nop     dword ptr [rax]
0xa410: endbr64
0xa414: sub     rsp, 8
0xa418: mov     edx, dword ptr [rip + 0x2057e]
0xa41e: imul    edx, dword ptr [rip + 0x20573]
0xa425: mov     esi, 0x20
0xa42a: mov     rdi, qword ptr [rip + 0x20f07]
0xa431: movsxd  rdx, edx
0xa434: call    0x4a50
0xa439: mov     edx, dword ptr [rip + 0x2055d]
0xa43f: imul    edx, dword ptr [rip + 0x20552]
0xa446: movzx   esi, byte ptr [rip + 0x20f04]
0xa44d: mov     rdi, qword ptr [rip + 0x20edc]
0xa454: movsxd  rdx, edx
0xa457: call    0x4a50
0xa45c: mov     edx, dword ptr [rip + 0x2053a]
0xa462: imul    edx, dword ptr [rip + 0x2052f]
0xa469: xor     esi, esi
0xa46b: mov     rdi, qword ptr [rip + 0x20eb6]
0xa472: add     rsp, 8
0xa476: movsxd  rdx, edx
0xa479: jmp     0x4a50
0x4a50

```

```

0xa47e: nop
0xa480: endbr64
0xa484: mov     edx, dword ptr [rip + 0x20512]
0xa48a: imul    edx, dword ptr [rip + 0x20507]
0xa491: xor     esi, esi
0xa493: mov     rdi, qword ptr [rip + 0x20e8e]
0xa49a: movsxd  rdx, edx
0xa49d: jmp     0x4a50
0x4a50
0xa4a2: nop     word ptr cs:[rax + rax]
0xa4ad: nop     dword ptr [rax]
0xa4b0: endbr64
0xa4b4: mov     dword ptr [rip + 0x20e9a], edi
0xa4ba: mov     dword ptr [rip + 0x20e98], esi
0xa4c0: ret
0xa4c1: nop     word ptr cs:[rax + rax]
0xa4cc: nop     dword ptr [rax]
0xa4d0: endbr64
0xa4d4: mov     eax, dword ptr [rip + 0x20e7a]
0xa4da: mov     dword ptr [rdi], eax
0xa4dc: mov     eax, dword ptr [rip + 0x20e76]
0xa4e2: mov     dword ptr [rsi], eax
0xa4e4: ret
0xa4e5: nop     word ptr cs:[rax + rax]
0xa4f0: endbr64
0xa4f4: movzx   eax, byte ptr [rip + 0x20e56]
0xa4fb: and     edi, 7
0xa4fe: and     eax, 0xffffffff8
0xa501: or      eax, edi
0xa503: mov     byte ptr [rip + 0x20e48], al
0xa509: ret
0xa50a: nop     word ptr [rax + rax]
0xa510: endbr64
0xa514: movzx   eax, byte ptr [rip + 0x20e36]
0xa51b: shl     edi, 3
0xa51e: and     edi, 0x38
0xa521: and     eax, 0xffffffffc7
0xa524: or      eax, edi
0xa526: mov     byte ptr [rip + 0x20e25], al
0xa52c: ret
0xa52d: nop     dword ptr [rax]
0xa530: endbr64
0xa534: movzx   eax, byte ptr [rip + 0x20e16]
0xa53b: shl     edi, 6
0xa53e: and     edi, 0x40
0xa541: and     eax, 0xffffffffbf
0xa544: or      eax, edi
0xa546: mov     byte ptr [rip + 0x20e05], al

```

```

0xa54c: ret
0xa54d: nop      dword ptr [rax]
0xa550: endbr64
0xa554: movzx    eax, byte ptr [rip + 0x20df6]
0xa55b: shl      edi, 7
0xa55e: and      eax, 0x7f
0xa561: or       eax, edi
0xa563: mov      byte ptr [rip + 0x20de8], al
0xa569: ret
0xa56a: nop      word ptr [rax + rax]
0xa570: endbr64
0xa574: shl      edi, 6
0xa577: and      esi, 7
0xa57a: shl      edx, 3
0xa57d: mov      eax, edi
0xa57f: movzx    edi, byte ptr [rip + 0x20dcb]
0xa586: and      edx, 0x38
0xa589: and      eax, 0x40
0xa58c: and      edi, 0xffffffffb8
0xa58f: or       edi, eax
0xa591: or       edi, esi
0xa593: and      edi, 0xffffffffc7
0xa596: or       edi, edx
0xa598: mov      byte ptr [rip + 0x20db2], dil
0xa59f: ret
0xa5a0: endbr64
0xa5a4: push     rbp
0xa5a5: push     rbx
0xa5a6: sub      rsp, 8
0xa5aa: movsxd   rax, dword ptr [rip + 0x20957]
0xa5b1: cmp      eax, 0x400
0xa5b6: je       0xa610
0xa5b8: lea      edx, [rax + 1]
0xa5bb: lea      rbx, [rip + 0x2095e]
0xa5c2: mov      dword ptr [rip + 0x20940], edx
0xa5c8: mov      byte ptr [rbx + rax], 7
0xa5cc: test     edx, edx
0xa5ce: jle      0xa5fd
0xa5d0: xor      ebp, ebp
0xa5d2: jmp      0xa5e4
0xa5e4:
0xa5d4: nop      dword ptr [rax]
0xa5d8: mov      edx, dword ptr [rip + 0x2092a]
0xa5de: add      ebp, eax
0xa5e0: cmp      edx, ebp
0xa5e2: jle      0xa5fd
0xa5e4: sub      edx, ebp
0xa5e6: movsxd   rsi, ebp

```

```

0xa5e9: mov     edi, 1
0xa5ee: add     rsi, rbx
0xa5f1: movsxd  rdx, edx
0xa5f4: call    0x18da0
0xa5f9: test    eax, eax
0xa5fb: jns     0xa5d8
0xa5fd: mov     dword ptr [rip + 0x20901], 0
0xa607: add     rsp, 8
0xa60b: pop     rbx
0xa60c: pop     rbp
0xa60d: ret
0xa60e: nop
0xa610: xor     ebp, ebp
0xa612: lea     rbx, [rip + 0x20907]
0xa619: jmp     0xa62c
0xa62c:
0xa61b: nop     dword ptr [rax + rax]
0xa620: add     ebp, eax
0xa622: mov     eax, dword ptr [rip + 0x208e0]
0xa628: cmp     eax, ebp
0xa62a: jle     0xa645
0xa62c: sub     eax, ebp
0xa62e: movsxd  rsi, ebp
0xa631: mov     edi, 1
0xa636: add     rsi, rbx
0xa639: movsxd  rdx, eax
0xa63c: call    0x18da0
0xa641: test    eax, eax
0xa643: jns     0xa620
0xa645: mov     dword ptr [rip + 0x208b9], 1
0xa64f: mov     edx, 1
0xa654: mov     byte ptr [rip + 0x208c5], 7
0xa65b: jmp     0xa5d0
0xa5d0:
0xa660: endbr64
0xa664: test    edi, edi
0xa666: je      0xa688
0xa668: mov     rdi, qword ptr [rip + 0x1f759]
0xa66f: test    rdi, rdi
0xa672: je      0xa694
0xa674: lea     rdx, [rip - 0xc0b]
0xa67b: mov     esi, 1
0xa680: jmp     0x4820
0x4820:
0xa685: nop     dword ptr [rax]
0xa688: mov     rdi, qword ptr [rip + 0x1f719]
0xa68f: test    rdi, rdi
0xa692: jne     0xa674

```

```

0xa694: ret
0xa695: nop      word ptr cs:[rax + rax]
0xa6a0: endbr64
0xa6a4: movzx    eax, byte ptr [rip + 0x20ca6]
0xa6ab: mov      byte ptr [rdi], al
0xa6ad: ret
0xa6ae: nop
0xa6b0: endbr64
0xa6b4: movzx    eax, byte ptr [rdi]
0xa6b7: mov      byte ptr [rip + 0x20c94], al
0xa6bd: ret
0xa6be: nop
0xa6c0: endbr64
0xa6c4: push     rbp
0xa6c5: push     rbx
0xa6c6: sub      rsp, 8
0xa6ca: cmp      dword ptr [rip + 0x1f58f], 1
0xa6d1: je       0xa720
0xa6d3: mov      rdi, qword ptr [rip + 0x1f66e]
0xa6da: test     rdi, rdi
0xa6dd: je       0xa6f0
0xa6df: lea      rdx, [rip - 0xc76]
0xa6e6: mov      esi, 1
0xa6eb: call     0x4820
0xa6f0: mov      dword ptr [rip + 0x20c52], 0
0xa6fa: mov      dword ptr [rip + 0x20c44], 0
0xa704: mov      dword ptr [rip + 0x20c36], 0
0xa70e: mov      dword ptr [rip + 0x20c28], 0
0xa718: add      rsp, 8
0xa71c: pop      rbx
0xa71d: pop      rbp
0xa71e: ret
0xa71f: nop
0xa720: mov      edx, dword ptr [rip + 0x207e2]
0xa726: cmp      edx, 0x3fb
0xa72c: jg       0xa750
0xa72e: movsxd   rax, edx
0xa731: lea      rcx, [rip + 0x207e8]
0xa738: add      edx, 4
0xa73b: add      rax, rcx
0xa73e: mov      ecx, dword ptr [rip + 0x1f5db]
0xa744: mov      dword ptr [rip + 0x207be], edx
0xa74a: mov      dword ptr [rax], ecx
0xa74c: jmp      0xa6d3
0xa6d3
0xa74e: nop
0xa750: xor      ebx, ebx
0xa752: lea      rbp, [rip + 0x207c7]

```

```

0xa759: jmp      0xa76c
0xa76c
0xa75b: nop      dword ptr [rax + rax]
0xa760: mov      edx, dword ptr [rip + 0x207a2]
0xa766: add      ebx, eax
0xa768: cmp      edx, ebx
0xa76a: jle      0xa785
0xa76c: sub      edx, ebx
0xa76e: movsxd   rsi, ebx
0xa771: mov      edi, 1
0xa776: add      rsi, rbp
0xa779: movsxd   rdx, edx
0xa77c: call     0x18da0
0xa781: test     eax, eax
0xa783: jns      0xa760
0xa785: mov      edx, 4
0xa78a: lea      rax, [rip + 0x2078f]
0xa791: jmp      0xa73e
0xa73e
0xa793: nop      word ptr cs:[rax + rax]
0xa79e: nop
0xa7a0: push     r12
0xa7a2: mov      r12d, edi
0xa7a5: push     rbp
0xa7a6: mov      ebp, esi
0xa7a8: sub      rsp, 8
0xa7ac: mov      eax, dword ptr [rip + 0x1f71e]
0xa7b2: test     eax, eax
0xa7b4: je       0xa800
0xa7b6: mov      eax, ebp
0xa7b8: mov      esi, dword ptr [rip + 0x201da]
0xa7be: mov      edx, dword ptr [rip + 0x201d8]
0xa7c4: or       eax, r12d
0xa7c7: mov      rdi, qword ptr [rip + 0x1f63a]
0xa7ce: js       0xa7d4
0xa7d0: cmp      ebp, esi
0xa7d2: jl       0xa810
0xa7d4: sub      edx, 1
0xa7d7: sub      esi, 1
0xa7da: call     0x4960
0xa7df: add      rsp, 8
0xa7e3: mov      esi, 1
0xa7e8: lea      rdx, [rip - 0xd7f]
0xa7ef: pop      rbp
0xa7f0: mov      rdi, rax
0xa7f3: pop      r12
0xa7f5: jmp      0x4820
0x4820

```

```

0xa7fa: nop      word ptr [rax + rax]
0xa800: xor      eax, eax
0xa802: call     0xa6c0
0xa807: jmp      0xa7b6
0xa7b6
0xa809: nop      dword ptr [rax]
0xa810: cmp      r12d, edx
0xa813: jge      0xa7d4
0xa815: mov      edx, r12d
0xa818: mov      esi, ebp
0xa81a: jmp      0xa7da
0xa7da
0xa81c: nop      dword ptr [rax]
0xa820: endbr64
0xa824: push     r15
0xa826: push     r14
0xa828: push     r13
0xa82a: push     r12
0xa82c: push     rbp
0xa82d: push     rbx
0xa82e: sub      rsp, 0x38
0xa832: mov      ebx, dword ptr [rip + 0x20160]
0xa838: imul     ebx, dword ptr [rip + 0x2015d]
0xa83f: mov      rax, qword ptr fs:[0x28]
0xa848: mov      qword ptr [rsp + 0x28], rax
0xa84d: xor      eax, eax
0xa84f: mov      rdi, qword ptr [rip + 0x1f552]
0xa856: cmp      dword ptr [rip + 0x20aff], 1
0xa85d: sbb      ebx, 0
0xa860: test     rdi, rdi
0xa863: je       0xa876
0xa865: lea      rdx, [rip - 0xdfc]
0xa86c: mov      esi, 1
0xa871: call     0x4820
0xa876: test     ebx, ebx
0xa878: jle      0xab70
0xa87e: movsxd   rax, ebx
0xa881: mov      dword ptr [rsp + 4], 0xffffffff
0xa889: xor      ebp, ebp
0xa88b: mov      r13d, 0xffffffff
0xa891: mov      qword ptr [rsp + 8], rax
0xa896: lea      r12, [rip + 0x20683]
0xa89d: jmp      0xa9ce
0xa9ce
0xa8a2: nop      word ptr [rax + rax]
0xa8a8: cdq
0xa8a9: idiv     dword ptr [rip + 0x200e9]
0xa8af: mov      r14d, edx

```



```

0xa8b2: cmp     edx, r13d
0xa8b5: jne     0xaa20
0xa8bb: cmp     eax, dword ptr [rsp + 4]
0xa8bf: jne     0xaa20
0xa8c5: mov     rax, qword ptr [rip + 0x20a64]
0xa8cc: movzx   ebx, byte ptr [rax + rbp]
0xa8d0: mov     edi, ebx
0xa8d2: mov     r14d, ebx
0xa8d5: sar     edi, 6
0xa8d8: and     r14d, 7
0xa8dc: and     edi, 1
0xa8df: call    0x9870
0xa8e4: mov     eax, ebx
0xa8e6: and     eax, 7
0xa8e9: cmp     dword ptr [rip + 0x20a5c], 1
0xa8f0: je      0xaa40
0xa8f6: cmp     dword ptr [rip + 0x1f363], 1
0xa8fd: je      0xaa5f
0xa903: xor     edi, edi
0xa905: cmp     eax, 7
0xa908: setne   dil
0xa90c: call    0x9790
0xa911: movzx   eax, byte ptr [rip + 0x20a38]
0xa918: mov     dword ptr [rip + 0x20a2a], 1
0xa922: and     eax, 0xffffffff8
0xa925: or      eax, r14d
0xa928: mov     byte ptr [rip + 0x20a22], al
0xa92e: sar     ebx, 3
0xa931: and     ebx, 7
0xa934: cmp     dword ptr [rip + 0x20a0d], 1
0xa93b: je      0xaab0
0xa941: cmp     dword ptr [rip + 0x1f318], 1
0xa948: je      0xaad8
0xa94e: xor     edi, edi
0xa950: test    ebx, ebx
0xa952: setne   dil
0xa956: call    0x9790
0xa95b: mov     edx, dword ptr [rip + 0x205a7]
0xa961: movzx   eax, byte ptr [rip + 0x209e8]
0xa968: shl     ebx, 3
0xa96b: mov     dword ptr [rip + 0x209d3], 1
0xa975: and     eax, 0xffffffc7
0xa978: or      ebx, eax
0xa97a: mov     byte ptr [rip + 0x209d0], bl
0xa980: mov     rax, qword ptr [rip + 0x209b1]
0xa987: movzx   ebx, byte ptr [rax + rbp]
0xa98b: cmp     edx, 0x400
0xa991: je      0xab30

```

```

0xa997: lea     eax, [rdx + 1]
0xa99a: movsxd  rdx, edx
0xa99d: add     r13d, 1
0xa9a1: cmp     dword ptr [rip + 0x1fff0], r13d
0xa9a8: mov     dword ptr [rip + 0x2055a], eax
0xa9ae: mov     byte ptr [r12 + rdx], bl
0xa9b2: jne     0xa9bf
0xa9b4: mov     dword ptr [rsp + 4], 0xffffffff
0xa9bc: xor     r13d, r13d
0xa9bf: add     rbp, 1
0xa9c3: cmp     rbp, qword ptr [rsp + 8]
0xa9c8: je      0xab70
0xa9ce: mov     rdx, qword ptr [rip + 0x20953]
0xa9d5: mov     rcx, qword ptr [rip + 0x2095c]
0xa9dc: mov     eax, ebp
0xa9de: movzx   ebx, byte ptr [rdx + rbp]
0xa9e2: cmp     byte ptr [rcx + rbp], bl
0xa9e5: jne     0xa8a8
0xa9eb: mov     rdx, qword ptr [rip + 0x2092e]
0xa9f2: mov     rcx, qword ptr [rip + 0x20937]
0xa9f9: movzx   edi, byte ptr [rdx + rbp]
0xa9fd: cmp     byte ptr [rcx + rbp], dil
0xaa01: je      0xa9bf
0xaa03: cdq
0xaa04: idiv    dword ptr [rip + 0x1ff8e]
0xaa0a: mov     r14d, edx
0xaa0d: cmp     edx, r13d
0xaa10: je      0xa8bb
0xaa16: nop     word ptr cs:[rax + rax]
0xaa20: mov     esi, r14d
0xaa23: mov     edi, eax
0xaa25: mov     dword ptr [rsp + 4], eax
0xaa29: mov     r13d, r14d
0xaa2c: call    0xa7a0
0xaa31: jmp     0xa8c5
0xa8c5:
0xaa36: nop     word ptr cs:[rax + rax]
0xaa40: movzx   edx, byte ptr [rip + 0x20909]
0xaa47: and     edx, 7
0xaa4a: cmp     eax, edx
0xaa4c: je      0xa92e
0xaa52: cmp     dword ptr [rip + 0x1f207], 1
0xaa59: jne     0xa903
0xaa5f: mov     eax, dword ptr [rip + 0x164b4]
0xaa65: mov     edx, dword ptr [rip + 0x2049d]
0xaa6b: mov     dword ptr [rsp + 0x10], eax
0xaa6f: movzx   eax, byte ptr [rip + 0x164a7]
0xaa76: mov     byte ptr [rsp + 0x14], al

```

```

0xaa7a: lea     eax, [r14 + 0x30]
0xaa7e: mov     byte ptr [rsp + 0x13], al
0xaa82: cmp     edx, 0x3fa
0xaa88: jg      0xac18
0xaa8e: movsxd  rax, edx
0xaa91: add     edx, 5
0xaa94: add     rax, r12
0xaa97: mov     ecx, dword ptr [rsp + 0x10]
0xaa9b: mov     dword ptr [rip + 0x20467], edx
0xaaa1: mov     dword ptr [rax], ecx
0xaaa3: movzx   ecx, byte ptr [rsp + 0x14]
0xaaa8: mov     byte ptr [rax + 4], cl
0xaaaab: jmp     0xa911
0xa911
0xaab0: movzx   eax, byte ptr [rip + 0x20899]
0xaab7: mov     edx, dword ptr [rip + 0x2044b]
0xaabd: sar     eax, 3
0xaac0: and     eax, 7
0xaac3: cmp     ebx, eax
0xaac5: je      0xa980
0xaacb: cmp     dword ptr [rip + 0x1f18e], 1
0xaad2: jne     0xa94e
0xaad8: mov     eax, dword ptr [rip + 0x16436]
0xaade: mov     edx, dword ptr [rip + 0x20424]
0xaae4: mov     dword ptr [rsp + 0x10], eax
0xaae8: movzx   eax, byte ptr [rip + 0x16429]
0xaaef: mov     byte ptr [rsp + 0x14], al
0xaaf3: lea     eax, [rbx + 0x30]
0xaaf6: mov     byte ptr [rsp + 0x13], al
0xaafa: cmp     edx, 0x3fa
0xab00: jg      0xac58
0xab06: movsxd  rax, edx
0xab09: add     edx, 5
0xab0c: add     rax, r12
0xab0f: mov     ecx, dword ptr [rsp + 0x10]
0xab13: mov     dword ptr [rip + 0x203ef], edx
0xab19: mov     dword ptr [rax], ecx
0xab1b: movzx   ecx, byte ptr [rsp + 0x14]
0xab20: mov     byte ptr [rax + 4], cl
0xab23: jmp     0xa961
0xa961
0xab28: nop     dword ptr [rax + rax]
0xab30: xor     r14d, r14d
0xab33: jmp     0xab46
0xab46
0xab35: nop     dword ptr [rax]
0xab38: mov     edx, dword ptr [rip + 0x203ca]
0xab3e: add     r14d, eax

```

```

0xab41: cmp     edx, r14d
0xab44: jle     0xab60
0xab46: sub     edx, r14d
0xab49: movsxd  rsi, r14d
0xab4c: mov     edi, 1
0xab51: add     rsi, r12
0xab54: movsxd  rdx, edx
0xab57: call    0x18da0
0xab5c: test    eax, eax
0xab5e: jns     0xab38
0xab60: mov     eax, 1
0xab65: xor     edx, edx
0xab67: jmp     0xa99a
0xa99a:
0xab6c: nop     dword ptr [rax]
0xab70: mov     edi, dword ptr [rip + 0x207de]
0xab76: mov     esi, dword ptr [rip + 0x207dc]
0xab7c: call    0xa7a0
0xab81: mov     rdi, qword ptr [rip + 0x1f240]
0xab88: test    rdi, rdi
0xab8b: je      0xab9e
0xab8d: lea     rdx, [rip - 0x1124]
0xab94: mov     esi, 1
0xab99: call    0x4820
0xab9e: mov     edx, dword ptr [rip + 0x20364]
0xaba4: test    edx, edx
0xaba6: jne     0xac98
0xabac: mov     edx, dword ptr [rip + 0x1fde6]
0xabab2: imul    edx, dword ptr [rip + 0x1fde3]
0xabab9: mov     rsi, qword ptr [rip + 0x20778]
0xabc0: mov     rdi, qword ptr [rip + 0x20761]
0xabc7: movsxd  rdx, edx
0xabca: call    0x4bb0
0xabcf: mov     edx, dword ptr [rip + 0x1fdc3]
0xabd5: imul    edx, dword ptr [rip + 0x1fdc0]
0xabdc: mov     rdi, qword ptr [rip + 0x2073d]
0xabe3: mov     rsi, qword ptr [rip + 0x20746]
0xabea: movsxd  rdx, edx
0xabed: mov     rax, qword ptr [rsp + 0x28]
0xabf2: sub     rax, qword ptr fs:[0x28]
0xabfb: jne     0xace4
0xac01: add     rsp, 0x38
0xac05: pop     rbx
0xac06: pop     rbp
0xac07: pop     r12
0xac09: pop     r13
0xac0b: pop     r14
0xac0d: pop     r15

```

```

0xac0f: jmp     0x4bb0
0x4bb0
0xac14: nop     dword ptr [rax]
0xac18: xor     r15d, r15d
0xac1b: jmp     0xac2e
0xac2e
0xac1d: nop     dword ptr [rax]
0xac20: mov     edx, dword ptr [rip + 0x202e2]
0xac26: add     r15d, eax
0xac29: cmp     edx, r15d
0xac2c: jle     0xac48
0xac2e: sub     edx, r15d
0xac31: movsxd  rsi, r15d
0xac34: mov     edi, 1
0xac39: add     rsi, r12
0xac3c: movsxd  rdx, edx
0xac3f: call    0x18da0
0xac44: test    eax, eax
0xac46: jns     0xac20
0xac48: mov     edx, 5
0xac4d: mov     rax, r12
0xac50: jmp     0xaa97
0xaa97
0xac55: nop     dword ptr [rax]
0xac58: xor     r14d, r14d
0xac5b: jmp     0xac6e
0xac6e
0xac5d: nop     dword ptr [rax]
0xac60: mov     edx, dword ptr [rip + 0x202a2]
0xac66: add     r14d, eax
0xac69: cmp     edx, r14d
0xac6c: jle     0xac88
0xac6e: sub     edx, r14d
0xac71: movsxd  rsi, r14d
0xac74: mov     edi, 1
0xac79: add     rsi, r12
0xac7c: movsxd  rdx, edx
0xac7f: call    0x18da0
0xac84: test    eax, eax
0xac86: jns     0xac60
0xac88: mov     edx, 5
0xac8d: mov     rax, r12
0xac90: jmp     0xab0f
0xab0f
0xac95: nop     dword ptr [rax]
0xac98: mov     ebx, 0
0xac9d: lea     rbp, [rip + 0x2027c]
0xaca4: jg      0xacbc

```

```

0xaca6: jmp      0xacd5
0xacd5:
0xaca8: nop      dword ptr [rax + rax]
0xacb0: mov      edx, dword ptr [rip + 0x20252]
0xacb6: add      ebx, eax
0xacb8: cmp      edx, ebx
0xacba: jle      0xacd5
0xacbc: sub      edx, ebx
0xacbe: movsxd   rsi, ebx
0xacc1: mov      edi, 1
0xacc6: add      rsi, rbp
0xacc9: movsxd   rdx, edx
0xaccc: call     0x18da0
0xacd1: test     eax, eax
0xacd3: jns      0xacb0
0xacd5: mov      dword ptr [rip + 0x20229], 0
0xacdf: jmp      0xabac
0xabac:
0xace4: call     0x49d0
0xace9: nop      dword ptr [rax]
0xacf0: endbr64
0xacf4: push     r12
0xacf6: xor      eax, eax
0xacf8: mov      r12, rdi
0xacfb: push     rbx
0xacfc: mov      ebx, esi
0xacfe: sub      rsp, 8
0xad02: call     0xa820
0xad07: mov      rdx, qword ptr [rip + 0x1fc1a]
0xad0e: test     rdx, rdx
0xad11: je       0xad17
0xad13: xor      eax, eax
0xad15: call     rdx
0xad17: movsxd   rdx, ebx
0xad1a: mov      rsi, r12
0xad1d: xor      edi, edi
0xad1f: call     0x18d30
0xad24: mov      rdx, qword ptr [rip + 0x1fbf5]
0xad2b: mov      r12d, eax
0xad2e: test     rdx, rdx
0xad31: je       0xad37
0xad33: xor      eax, eax
0xad35: call     rdx
0xad37: add      rsp, 8
0xad3b: mov      eax, r12d
0xad3e: pop      rbx
0xad3f: pop      r12
0xad41: ret

```

```

0xad42: nop      word ptr cs:[rax + rax]
0xad4d: nop      dword ptr [rax]
0xad50: push     r12
0xad52: mov      edi, 1
0xad57: push     rbp
0xad58: lea      rbp, [rip + 0x1fda1]
0xad5f: push     rbx
0xad60: sub      rsp, 0x50
0xad64: mov      rax, qword ptr fs:[0x28]
0xad6d: mov      qword ptr [rsp + 0x48], rax
0xad72: xor      eax, eax
0xad74: mov      r12, rsp
0xad77: mov      rsi, r12
0xad7a: call     0x4d60
0xad7f: mov      edx, 0xff
0xad84: mov      esi, 1
0xad89: mov      edi, 1
0xad8e: mov      word ptr [rsp + 0x11], dx
0xad93: mov      rdx, r12
0xad96: call     0x4d70
0xad9b: mov      edi, 1
0xada0: mov      dword ptr [rip + 0x1ef7a], 0xffffffff
0xadaa: call     0x18b70
0xadaf: mov      dword ptr [rip + 0x20147], 0
0xadb9: nop      dword ptr [rax]
0xadc0: xor      eax, eax
0xadc2: call     0xa820
0xadc7: mov      rdx, qword ptr [rip + 0x1fb5a]
0xadce: test     rdx, rdx
0xadd1: je       0xadd7
0xadd3: xor      eax, eax
0xadd5: call     rdx
0xadd7: mov      edx, 0x400
0xaddc: mov      rsi, rbp
0xaddf: xor      edi, edi
0xade1: call     0x18d30
0xade6: mov      rdx, qword ptr [rip + 0x1fb33]
0xaded: mov      ebx, eax
0xadeb: test     rdx, rdx
0xadf2: je       0xadf8
0xadf4: xor      eax, eax
0xadf6: call     rdx
0xadf8: mov      dword ptr [rip + 0x20106], ebx
0xadfe: test     ebx, ebx
0xae00: js       0xadc0
0xae02: xor      edi, edi
0xae04: call     0x18b70
0xae09: mov      rsi, r12

```

```

0xae0c: mov     edi, 1
0xae11: call    0x4d60
0xae16: mov     eax, 7
0xae1b: mov     rdx, r12
0xae1e: mov     esi, 1
0xae23: mov     edi, 1
0xae28: mov     word ptr [rsp + 0x11], ax
0xae2d: call    0x4d70
0xae32: mov     eax, dword ptr [rip + 0x200cc]
0xae38: mov     dword ptr [rip + 0x1eee2], 7
0xae42: test    eax, eax
0xae44: je      0xae9f
0xae46: movsxd  rsi, dword ptr [rip + 0x200b3]
0xae4d: movsx   r8d, byte ptr [rbp + rsi]
0xae53: mov     rdx, rsi
0xae56: mov     ecx, r8d
0xae59: test    r8b, r8b
0xae5c: js      0xae90
0xae5e: sub     eax, 1
0xae61: add     edx, 1
0xae64: mov     dword ptr [rip + 0x2009a], eax
0xae6a: mov     dword ptr [rip + 0x20090], edx
0xae70: mov     rax, qword ptr [rsp + 0x48]
0xae75: sub     rax, qword ptr fs:[0x28]
0xae7e: jne     0xaea7
0xae80: add     rsp, 0x50
0xae84: mov     eax, r8d
0xae87: pop     rbx
0xae88: pop     rbp
0xae89: pop     r12
0xae8b: ret
0xae8c: nop     dword ptr [rax]
0xae90: and     ecx, 0x7f
0xae93: mov     r8d, 0x1b
0xae99: mov     byte ptr [rbp + rsi], cl
0xae9d: jmp     0xae70
0xae70:
0xae9f: mov     r8d, 0xffffffff
0xaea5: jmp     0xae70
0xae70:
0xaea7: call    0x49d0
0xaeac: nop     dword ptr [rax]
0xaeb0: endbr64
0xaeb4: xor     eax, eax
0xaeb6: sub     rsp, 8
0xaeba: call    0x18ab0
0xaebf: mov     eax, dword ptr [rip + 0x2003f]
0xaec5: test    eax, eax

```



```

0xaec7: jne      0xaed8
0xaec9: xor      eax, eax
0xaecb: add      rsp, 8
0xaecf: jmp      0xad50
0xad50:
0xaed4: nop      dword ptr [rax]
0xaed8: movsxd   rsi, dword ptr [rip + 0x20021]
0xaedf: lea      rdi, [rip + 0x1fc1a]
0xae6: movsx    r8d, byte ptr [rdi + rsi]
0xaeeb: mov      rdx, rsi
0xaeee: mov      ecx, r8d
0xaeef1: test     r8b, r8b
0xaeef4: js       0xaf10
0xaeef6: sub      eax, 1
0xaeef9: add      edx, 1
0xaeefc: mov      dword ptr [rip + 0x20002], eax
0xaf02: mov      dword ptr [rip + 0x1fff8], edx
0xaf08: mov      eax, r8d
0xaf0b: add      rsp, 8
0xaf0f: ret
0xaf10: and      ecx, 0x7f
0xaf13: mov      r8d, 0x1b
0xaf19: mov      byte ptr [rdi + rsi], cl
0xaf1c: jmp      0xaf08
0xaf08:
0xaf1e: nop
0xaf20: endbr64
0xaf24: push     r13
0xaf26: mov      r13, rsi
0xaf29: push     r12
0xaf2b: mov      r12, rdx
0xaf2e: push     rbp
0xaf2f: mov      rbp, rdi
0xaf32: mov      edi, 0x18
0xaf37: push     rbx
0xaf38: sub      rsp, 8
0xaf3c: call     0x1d670
0xaf41: mov      rdi, r13
0xaf44: mov      rbx, rax
0xaf47: call     0x1d7a0
0xaf4c: mov      qword ptr [rbx + 0x10], r12
0xaf50: mov      qword ptr [rbx], rax
0xaf53: mov      rax, qword ptr [rbp]
0xaf57: mov      qword ptr [rbx + 8], rax
0xaf5b: mov      qword ptr [rbp], rbx
0xaf5f: add      rsp, 8
0xaf63: pop      rbx
0xaf64: pop      rbp

```

```

0xaf65: pop     r12
0xaf67: pop     r13
0xaf69: ret
0xaf6a: nop     word ptr [rax + rax]
0xaf70: endbr64
0xaf74: cmp     byte ptr [rdi], 0
0xaf77: je      0xb030
0xaf7d: push    r13
0xaf7f: push    r12
0xaf81: mov     r12, rsi
0xaf84: push    rbp
0xaf85: mov     rbp, rdi
0xaf88: push    rbx
0xaf89: sub     rsp, 8
0xaf8d: mov     r13, qword ptr [rip + 0x1fa0c]
0xaf94: test    r13, r13
0xaf97: je      0xb010
0xaf99: mov     rbx, qword ptr [r13]
0xaf9d: mov     rsi, qword ptr [rbx]
0xafa0: call    0x4b30
0xafa5: test    eax, eax
0xafa7: jle     0xb010
0xafa9: nop     dword ptr [rax]
0xafb0: lea     r13, [rbx + 8]
0xafb4: mov     rbx, qword ptr [rbx + 8]
0xafb8: mov     qword ptr [rip + 0x1f9e1], r13
0xafbf: test    rbx, rbx
0xafc2: je      0xafd3
0xafc4: mov     rsi, qword ptr [rbx]
0xafc7: mov     rdi, rbp
0xafca: call    0x4b30
0xafcf: test    eax, eax
0xafd1: jg      0xafb0
0xafd3: mov     edi, 0x18
0xafd8: call    0x1d670
0xafdd: mov     rdi, rbp
0xafe0: mov     rbx, rax
0xafe3: call    0x1d7a0
0xafe8: mov     qword ptr [rbx + 0x10], r12
0xafec: mov     qword ptr [rbx], rax
0xafef: mov     rax, qword ptr [r13]
0xaff3: mov     qword ptr [rbx + 8], rax
0xaff7: mov     qword ptr [r13], rbx
0xaffb: add     rsp, 8
0xafff: pop     rbx
0xb000: pop     rbp
0xb001: pop     r12
0xb003: pop     r13

```

```

0xb005: ret
0xb006: nop      word ptr cs:[rax + rax]
0xb010: mov      rbx, qword ptr [rip + 0x1f969]
0xb017: lea      r13, [rip + 0x1f962]
0xb01e: mov      qword ptr [rip + 0x1f97b], r13
0xb025: test     rbx, rbx
0xb028: jne      0xafc4
0xb02a: jmp      0xafd3
0xafd3
0xb02c: nop      dword ptr [rax]
0xb030: ret
0xb031: nop      word ptr cs:[rax + rax]
0xb03c: nop      dword ptr [rax]
0xb040: endbr64
0xb044: mov      rax, qword ptr [rip + 0x1f935]
0xb04b: mov      qword ptr [rip + 0x1f926], rax
0xb052: ret
0xb053: nop      word ptr cs:[rax + rax]
0xb05e: nop
0xb060: endbr64
0xb064: cmp      qword ptr [rip + 0x1f90c], 0
0xb06c: je       0xb073
0xb06e: jmp      0x99e0
0x99e0
0xb073: xor      eax, eax
0xb075: ret
0xb076: nop      word ptr cs:[rax + rax]
0xb080: endbr64
0xb084: mov      eax, dword ptr [rip + 0x1f912]
0xb08a: push     r12
0xb08c: mov      r12, rdi
0xb08f: push     rbp
0xb090: push     rbx
0xb091: sub      eax, 1
0xb094: movzx    ebx, byte ptr [rip + 0x202b6]
0xb09b: mov      dword ptr [rip + 0x202b3], eax
0xb0a1: mov      dword ptr [rip + 0x202ad], 0
0xb0ab: mov      eax, ebx
0xb0ad: and      eax, 0xffffffffc0
0xb0b0: or       eax, 0x38
0xb0b3: mov      byte ptr [rip + 0x20298], al
0xb0b9: mov      eax, dword ptr [rip + 0x1f8d9]
0xb0bf: lea      edi, [rax + 1]
0xb0c2: movsxd   rdi, edi
0xb0c5: call     0x1d670
0xb0ca: movsxd   rdx, dword ptr [rip + 0x1f8c7]
0xb0d1: mov      esi, 0x20
0xb0d6: mov      rdi, rax

```

```

0xb0d9: mov     rbp, rax
0xb0dc: call    0x4a50
0xb0e1: movsxd  rax, dword ptr [rip + 0x1f8b0]
0xb0e8: mov     rdi, rbp
0xb0eb: mov     byte ptr [rbp + rax], 0
0xb0f0: mov     esi, dword ptr [rip + 0x1f8a2]
0xb0f6: call    0xa1a0
0xb0fb: mov     rdi, rbp
0xb0fe: lea     rbp, [rip + 0x1f8bb]
0xb105: call    0x18d10
0xb10a: mov     eax, dword ptr [rip + 0x1f88c]
0xb110: mov     rdi, r12
0xb113: mov     dword ptr [rip + 0x2023b], 0
0xb11d: sub     eax, 1
0xb120: mov     dword ptr [rip + 0x2022e], eax
0xb126: call    0x9fe0
0xb12b: mov     esi, 0x13
0xb130: lea     rdi, [rip + 0x15ad5]
0xb137: call    0xa1a0
0xb13c: mov     rdi, rbp
0xb13f: call    0x49b0
0xb144: mov     rdi, rbp
0xb147: mov     esi, eax
0xb149: call    0xa1a0
0xb14e: mov     esi, 1
0xb153: lea     rdi, [rip + 0x155eb]
0xb15a: call    0xa1a0
0xb15f: xor     eax, eax
0xb161: call    0xa820
0xb166: mov     byte ptr [rip + 0x201e5], bl
0xb16c: pop     rbx
0xb16d: pop     rbp
0xb16e: pop     r12
0xb170: ret
0xb171: nop     word ptr cs:[rax + rax]
0xb17c: nop     dword ptr [rax]
0xb180: endbr64
0xb184: push    r15
0xb186: push    r14
0xb188: mov     r14, rdi
0xb18b: push    r13
0xb18d: push    r12
0xb18f: push    rbp
0xb190: lea     rbp, [rip + 0x1f969]
0xb197: push    rbx
0xb198: sub     rsp, 0x28
0xb19c: jmp     0xb1ae
0xb1ae:

```

```

0xb19e: nop
0xb1a0: xor     eax, eax
0xb1a2: call    0xad50
0xb1a7: mov     ebx, eax
0xb1a9: cmp     ebx, -1
0xb1ac: jne     0xb1ef
0xb1ae: xor     eax, eax
0xb1b0: call    0x18ab0
0xb1b5: mov     eax, dword ptr [rip + 0x1fd49]
0xb1bb: test    eax, eax
0xb1bd: je      0xb1a0
0xb1bf: movsxd  rsi, dword ptr [rip + 0x1fd3a]
0xb1c6: movsx   ebx, byte ptr [rbp + rsi]
0xb1cb: mov     rdx, rsi
0xb1ce: mov     ecx, ebx
0xb1d0: test    bl, bl
0xb1d2: js      0xb4a0
0xb1d8: sub     eax, 1
0xb1db: add     edx, 1
0xb1de: mov     dword ptr [rip + 0x1fd20], eax
0xb1e4: mov     dword ptr [rip + 0x1fd16], edx
0xb1ea: cmp     ebx, -1
0xb1ed: je      0xb1ae
0xb1ef: test    r14, r14
0xb1f2: je      0xb1fb
0xb1f4: mov     dword ptr [r14], 1
0xb1fb: mov     eax, dword ptr [rip + 0x201fb]
0xb201: test    ebx, ebx
0xb203: jne     0xb445
0xb209: mov     ebx, 0xff
0xb20e: test    eax, eax
0xb210: jne     0xb240
0xb212: xor     r12d, r12d
0xb215: mov     ebp, 0x1fe
0xb21a: mov     ebx, 0xff
0xb21f: call    0x4f00
0xb224: mov     rax, qword ptr [rax]
0xb227: movzx   eax, word ptr [rax + rbp]
0xb22b: test    ah, 0x40
0xb22e: jne     0xb47d
0xb234: test    r12b, r12b
0xb237: jne     0xb47d
0xb23d: nop     dword ptr [rax]
0xb240: mov     rax, qword ptr [rip + 0x1f739]
0xb247: xor     r12d, r12d
0xb24a: lea     r13, [rip + 0x1f8af]
0xb251: mov     dword ptr [rip + 0x1f751], 0
0xb25b: mov     dword ptr [rip + 0x1f743], 0

```

```

0xb265: mov     qword ptr [rip + 0x1f70c], rax
0xb26c: test    ebx, ebx
0xb26e: mov     eax, 0xffffffff
0xb273: movsxd  rbp, r12d
0xb276: cmovne  eax, ebx
0xb279: mov     rbx, qword ptr [rip + 0x200e0]
0xb280: mov     byte ptr [rbx + rbp], al
0xb283: add     rbp, 1
0xb287: mov     byte ptr [rbx + rbp], 0
0xb28b: cmp     qword ptr [rip + 0x1f6e5], 0
0xb293: je      0xb2b3
0xb295: mov     rdi, rbx
0xb298: call    0x99e0
0xb29d: mov     r15, rax
0xb2a0: cmp     rax, -1
0xb2a4: je      0xb33a
0xb2aa: test    rax, rax
0xb2ad: jne     0xb38f
0xb2b3: mov     ecx, dword ptr [rip + 0x1fc4b]
0xb2b9: test    ecx, ecx
0xb2bb: jne     0xb2ee
0xb2bd: mov     edx, dword ptr [rip + 0x1f6e5]
0xb2c3: test    edx, edx
0xb2c5: je      0xb380
0xb2cb: mov     rdi, rbx
0xb2ce: call    0xb080
0xb2d3: mov     dword ptr [rip + 0x1f6cf], 1
0xb2dd: jmp     0xb2ee
0xb2ee
0xb2df: nop
0xb2e0: xor     eax, eax
0xb2e2: call    0xad50
0xb2e7: mov     ebx, eax
0xb2e9: cmp     ebx, -1
0xb2ec: jne     0xb32c
0xb2ee: xor     eax, eax
0xb2f0: call    0x18ab0
0xb2f5: mov     eax, dword ptr [rip + 0x1fc09]
0xb2fb: test    eax, eax
0xb2fd: je      0xb2e0
0xb2ff: movsxd  rsi, dword ptr [rip + 0x1fbfa]
0xb306: movsx   ebx, byte ptr [r13 + rsi]
0xb30c: mov     rdx, rsi
0xb30f: mov     ecx, ebx
0xb311: test    bl, bl
0xb313: js      0xb360
0xb315: sub     eax, 1
0xb318: add     edx, 1

```

```

0xb31b: mov     dword ptr [rip + 0x1fbe3], eax
0xb321: mov     dword ptr [rip + 0x1fbd9], edx
0xb327: cmp     ebx, -1
0xb32a: je      0xb2ee
0xb32c: add     r12d, 1
0xb330: cmp     r12d, 0xf
0xb334: jne     0xb26c
0xb33a: mov     edi, 1
0xb33f: xor     r15d, r15d
0xb342: call    0x4aa0
0xb347: mov     dword ptr [rip + 0x1f65b], 0
0xb351: jmp     0xb433
0xb433:
0xb356: nop     word ptr cs:[rax + rax]
0xb360: and     ecx, 0x7f
0xb363: add     r12d, 1
0xb367: mov     byte ptr [r13 + rsi], cl
0xb36c: cmp     r12d, 0xf
0xb370: je      0xb33a
0xb372: mov     eax, 0x1b
0xb377: jmp     0xb279
0xb279:
0xb37c: nop     dword ptr [rax]
0xb380: mov     edi, 1
0xb385: call    0x4aa0
0xb38a: jmp     0xb2d3
0xb2d3:
0xb38f: test    r14, r14
0xb392: je      0xb41f
0xb398: mov     ecx, dword ptr [rip + 0x1fb66]
0xb39e: cmp     r12d, ecx
0xb3a1: jge     0xb41f
0xb3a3: movsxd  rax, dword ptr [rip + 0x1fb56]
0xb3aa: lea     r9d, [r12 + 1]
0xb3af: lea     rdx, [rip + 0x1f74a]
0xb3b6: movsxd  rdi, r9d
0xb3b9: mov     qword ptr [rsp + 0x18], rdi
0xb3be: lea     r8d, [r9 + rax]
0xb3c2: lea     r13, [rax + rdx]
0xb3c6: jmp     0xb3f1
0xb3f1:
0xb3c8: nop     dword ptr [rax + rax]
0xb3d0: sub     ecx, r9d
0xb3d3: add     dword ptr [r14], 1
0xb3d7: add     r13, qword ptr [rsp + 0x18]
0xb3dc: mov     dword ptr [rip + 0x1fb1d], r8d
0xb3e3: add     r8d, r9d
0xb3e6: mov     dword ptr [rip + 0x1fb18], ecx

```

```

0xb3ec: cmp     r12d, ecx
0xb3ef: jge     0xb41f
0xb3f1: mov     rdx, rbp
0xb3f4: mov     rsi, r13
0xb3f7: mov     rdi, rbx
0xb3fa: mov     dword ptr [rsp + 0x14], r8d
0xb3ff: mov     dword ptr [rsp + 0x10], r9d
0xb404: mov     dword ptr [rsp + 0xc], ecx
0xb408: call    0x4af0
0xb40d: mov     ecx, dword ptr [rsp + 0xc]
0xb411: mov     r9d, dword ptr [rsp + 0x10]
0xb416: test    eax, eax
0xb418: mov     r8d, dword ptr [rsp + 0x14]
0xb41d: je      0xb3d0
0xb41f: mov     edi, 1
0xb424: call    0x4aa0
0xb429: mov     dword ptr [rip + 0x1f579], 0
0xb433: add     rsp, 0x28
0xb437: mov     rax, r15
0xb43a: pop     rbx
0xb43b: pop     rbp
0xb43c: pop     r12
0xb43e: pop     r13
0xb440: pop     r14
0xb442: pop     r15
0xb444: ret
0xb445: test    eax, eax
0xb447: jne     0xb240
0xb44d: cmp     ebx, 0xa
0xb450: sete    bpl
0xb454: cmp     ebx, 0xd
0xb457: sete    al
0xb45a: or      ebp, eax
0xb45c: call    0x4f00
0xb461: mov     rax, qword ptr [rax]
0xb464: test    bpl, bpl
0xb467: je      0xb4c6
0xb469: mov     ebx, 0xa
0xb46e: test    byte ptr [rax + 0x15], 0x40
0xb472: je      0xb240
0xb478: mov     ebx, 0xa
0xb47d: mov     rax, qword ptr [rip + 0x1f4dc]
0xb484: lea     r15, [rip + 0x1f4d5]
0xb48b: mov     byte ptr [rax], bl
0xb48d: mov     rax, qword ptr [rip + 0x1f4cc]
0xb494: mov     byte ptr [rax + 1], 0
0xb498: jmp     0xb433
0xb433

```



```

0xb49a: nop      word ptr [rax + rax]
0xb4a0: and      ecx, 0x7f
0xb4a3: mov      byte ptr [rbp + rsi], cl
0xb4a7: test     r14, r14
0xb4aa: je       0xb4b3
0xb4ac: mov      dword ptr [r14], 1
0xb4b3: mov      eax, dword ptr [rip + 0x1ff43]
0xb4b9: mov      ebx, 0x1b
0xb4be: test     eax, eax
0xb4c0: jne      0xb240
0xb4c6: movsxd   rbp, ebx
0xb4c9: add      rbp, rbp
0xb4cc: cmp      ebx, 7
0xb4cf: sete     r12b
0xb4d3: jmp      0xb21f
0xb21f
0xb4d8: nop      dword ptr [rax + rax]
0xb4e0: endbr64
0xb4e4: mov      eax, dword ptr [rip + 0x1f4c2]
0xb4ea: test     eax, eax
0xb4ec: jne      0xb4f0
0xb4ee: ret
0xb4ef: nop
0xb4f0: sub      rsp, 8
0xb4f4: mov      rdi, qword ptr [rip + 0x1fe65]
0xb4fb: call     0xb080
0xb500: mov      dword ptr [rip + 0x1f49e], 1
0xb50a: add      rsp, 8
0xb50e: ret
0xb50f: nop
0xb510: endbr64
0xb514: mov      rax, qword ptr [rip + 0x1fe45]
0xb51b: ret
0xb51c: nop      dword ptr [rax]
0xb520: endbr64
0xb524: push     r15
0xb526: mov      esi, 0x5413
0xb52b: mov      edi, 1
0xb530: push     r14
0xb532: push     r13
0xb534: push     r12
0xb536: push     rbp
0xb537: push     rbx
0xb538: sub      rsp, 0x58
0xb53c: mov      rax, qword ptr fs:[0x28]
0xb545: mov      qword ptr [rsp + 0x48], rax
0xb54a: xor      eax, eax
0xb54c: lea      rdx, [rsp + 0x18]

```

```

0xb551: mov     dword ptr [rsp + 0x10], 0
0xb559: mov     dword ptr [rsp + 0x14], 0
0xb561: call    0x4a70
0xb566: test    eax, eax
0xb568: jne     0xb5a0
0xb56a: mov     ebx, eax
0xb56c: movzx   eax, word ptr [rsp + 0x1a]
0xb571: test    ax, ax
0xb574: je      0xb790
0xb57a: movzx   ebp, word ptr [rsp + 0x18]
0xb57f: test    bp, bp
0xb582: je      0xb790
0xb588: movzx   ebx, ax
0xb58b: lea     r12d, [rbp - 1]
0xb58f: lea     eax, [rbx - 1]
0xb592: mov     dword ptr [rsp + 0xc], eax
0xb596: jmp     0xb5b2
0xb5b2:
0xb598: nop     dword ptr [rax + rax]
0xb5a0: mov     dword ptr [rsp + 0xc], 0xffffffff
0xb5a8: mov     r12d, 0xffffffff
0xb5ae: xor     ebx, ebx
0xb5b0: xor     ebp, ebp
0xb5b2: lea     r13, [rip + 0x15667]
0xb5b9: lea     r15, [rip + 0x15a13]
0xb5c0: mov     rdi, r13
0xb5c3: call    0x47e0
0xb5c8: mov     rdi, rax
0xb5cb: test    rax, rax
0xb5ce: je      0xb5df
0xb5d0: lea     rdx, [rsp + 0x14]
0xb5d5: mov     rsi, r15
0xb5d8: xor     eax, eax
0xb5da: call    0x4c60
0xb5df: lea     r14, [rip + 0x15642]
0xb5e6: mov     rdi, r14
0xb5e9: call    0x47e0
0xb5ee: mov     rdi, rax
0xb5f1: test    rax, rax
0xb5f4: je      0xb605
0xb5f6: lea     rdx, [rsp + 0x10]
0xb5fb: mov     rsi, r15
0xb5fe: xor     eax, eax
0xb600: call    0x4c60
0xb605: cmp     dword ptr [rsp + 0xc], 0x3ff
0xb60d: mov     eax, dword ptr [rip + 0x1e8fd]
0xb613: jbe     0xb628
0xb615: mov     ebx, dword ptr [rsp + 0x14]

```

```

0xb619: lea     edx, [rbx - 1]
0xb61c: cmp     edx, 0x3ff
0xb622: ja      0xb7c0
0xb628: mov     dword ptr [rip + 0x1f36a], ebx
0xb62e: cmp     r12d, 0x3ff
0xb635: jbe     0xb64a
0xb637: mov     ebp, dword ptr [rsp + 0x10]
0xb63b: lea     edx, [rbp - 1]
0xb63e: cmp     edx, 0x3ff
0xb644: ja      0xb7a8
0xb64a: lea     r12, [rsp + 0x20]
0xb64f: mov     r8d, ebp
0xb652: mov     rcx, r15
0xb655: mov     edx, 0x20
0xb65a: mov     esi, 1
0xb65f: mov     rdi, r12
0xb662: xor     eax, eax
0xb664: mov     dword ptr [rip + 0x1f332], ebp
0xb66a: call    0x4f10
0xb66f: mov     rsi, r12
0xb672: mov     rdi, r14
0xb675: call    0x18380
0xb67a: mov     rcx, r15
0xb67d: mov     edx, 0x20
0xb682: mov     rdi, r12
0xb685: mov     r8d, dword ptr [rip + 0x1f30c]
0xb68c: mov     esi, 1
0xb691: xor     eax, eax
0xb693: call    0x4f10
0xb698: mov     rdi, r13
0xb69b: mov     rsi, r12
0xb69e: call    0x18380
0xb6a3: mov     rdi, qword ptr [rip + 0x1fc8e]
0xb6aa: test    rdi, rdi
0xb6ad: je      0xb6b4
0xb6af: call    0x18d10
0xb6b4: mov     rdi, qword ptr [rip + 0x1fc75]
0xb6bb: test    rdi, rdi
0xb6be: je      0xb6c5
0xb6c0: call    0x18d10
0xb6c5: mov     rdi, qword ptr [rip + 0x1fc5c]
0xb6cc: test    rdi, rdi
0xb6cf: je      0xb6d6
0xb6d1: call    0x18d10
0xb6d6: mov     rdi, qword ptr [rip + 0x1fc43]
0xb6dd: test    rdi, rdi
0xb6e0: je      0xb6e7
0xb6e2: call    0x18d10

```

```

0xb6e7: mov     edi, dword ptr [rip + 0x1f2ab]
0xb6ed: imul     edi, dword ptr [rip + 0x1f2a8]
0xb6f4: mov     esi, 1
0xb6f9: movsxd   rdi, edi
0xb6fc: call     0x1d740
0xb701: mov     edi, dword ptr [rip + 0x1f291]
0xb707: mov     esi, 1
0xb70c: imul     edi, dword ptr [rip + 0x1f289]
0xb713: mov     qword ptr [rip + 0x1fc1e], rax
0xb71a: movsxd   rdi, edi
0xb71d: call     0x1d740
0xb722: mov     edi, dword ptr [rip + 0x1f270]
0xb728: mov     esi, 1
0xb72d: imul     edi, dword ptr [rip + 0x1f268]
0xb734: mov     qword ptr [rip + 0x1fbf5], rax
0xb73b: movsxd   rdi, edi
0xb73e: call     0x1d740
0xb743: mov     edi, dword ptr [rip + 0x1f24f]
0xb749: mov     esi, 1
0xb74e: imul     edi, dword ptr [rip + 0x1f247]
0xb755: mov     qword ptr [rip + 0x1fbcc], rax
0xb75c: movsxd   rdi, edi
0xb75f: call     0x1d740
0xb764: mov     qword ptr [rip + 0x1fbb5], rax
0xb76b: mov     rax, qword ptr [rsp + 0x48]
0xb770: sub     rax, qword ptr fs:[0x28]
0xb779: jne     0xb7dc
0xb77b: add     rsp, 0x58
0xb77f: pop     rbx
0xb780: pop     rbp
0xb781: pop     r12
0xb783: pop     r13
0xb785: pop     r14
0xb787: pop     r15
0xb789: ret
0xb78a: nop     word ptr [rax + rax]
0xb790: mov     dword ptr [rsp + 0xc], 0xffffffff
0xb798: xor     ebp, ebp
0xb79a: mov     r12d, 0xffffffff
0xb7a0: jmp     0xb5b2
0xb5b2:
0xb7a5: nop     dword ptr [rax]
0xb7a8: lea     edx, [rax - 1]
0xb7ab: mov     ebp, 0x18
0xb7b0: cmp     edx, 0x400
0xb7b6: cmovb   ebp, eax
0xb7b9: jmp     0xb64a
0xb64a:

```

```

0xb7be: nop
0xb7c0: mov     ebx, dword ptr [rip + 0x1e72a]
0xb7c6: lea     edx, [rbx - 1]
0xb7c9: cmp     edx, 0x3ff
0xb7cf: mov     edx, 0x50
0xb7d4: cmova   ebx, edx
0xb7d7: jmp     0xb628
0xb628:
0xb7dc: call    0x49d0
0xb7e1: nop     word ptr cs:[rax + rax]
0xb7ec: nop     dword ptr [rax]
0xb7f0: endbr64
0xb7f4: push    r15
0xb7f6: push    r14
0xb7f8: push    r13
0xb7fa: push    r12
0xb7fc: mov     r12, rdi
0xb7ff: mov     edi, 0x5858
0xb804: push    rbp
0xb805: push    rbx
0xb806: mov     ebx, esi
0xb808: sub     rsp, 0x38
0xb80c: mov     rax, qword ptr fs:[0x28]
0xb815: mov     qword ptr [rsp + 0x28], rax
0xb81a: movabs  rax, 0x7363762f7665642f
0xb824: mov     word ptr [rsp + 8], di
0xb829: mov     byte ptr [rsp + 0xa], 0
0xb82e: mov     dword ptr [rsp + 0x18], 0x585861
0xb836: mov     qword ptr [rsp], rax
0xb83a: mov     qword ptr [rsp + 0x10], rax
0xb83f: cmp     esi, 1
0xb842: je      0xb978
0xb848: mov     esi, dword ptr [rip + 0x1fbb2]
0xb84e: mov     r13, qword ptr [rip + 0x1d783]
0xb855: mov     r14d, 1
0xb85b: test    esi, esi
0xb85d: jne     0xb90c
0xb863: mov     rax, qword ptr [rip + 0x1f126]
0xb86a: lea     r15, [rsp + 0x10]
0xb86f: mov     esi, r14d
0xb872: mov     dword ptr [rip + 0x1fb84], 0
0xb87c: mov     rdi, r15
0xb87f: movzx   edx, byte ptr [rax + 8]
0xb883: mov     byte ptr [rsp + 0x19], dl
0xb887: movzx   eax, byte ptr [rax + 9]
0xb88b: mov     byte ptr [rsp + 0x1a], al
0xb88f: call    0x4930
0xb894: mov     ebp, eax

```

```

0xb896: cmp     eax, -1
0xb899: je      0xb8f0
0xb89b: mov     eax, dword ptr [rip + 0x1f0fb]
0xb8a1: imul    eax, dword ptr [rip + 0x1f0f0]
0xb8a8: mov     edi, ebp
0xb8aa: mov     rsi, r12
0xb8ad: lea     edx, [rax + rax + 4]
0xb8b1: xor     eax, eax
0xb8b3: call    r13
0xb8b6: mov     edi, ebp
0xb8b8: call    0x4ab0
0xb8bd: cmp     ebx, 2
0xb8c0: je      0xb990
0xb8c6: mov     eax, 1
0xb8cb: mov     rdx, qword ptr [rsp + 0x28]
0xb8d0: sub     rdx, qword ptr fs:[0x28]
0xb8d9: jne     0xbb07
0xb8df: add     rsp, 0x38
0xb8e3: pop     rbx
0xb8e4: pop     rbp
0xb8e5: pop     r12
0xb8e7: pop     r13
0xb8e9: pop     r14
0xb8eb: pop     r15
0xb8ed: ret
0xb8ee: nop
0xb8f0: mov     ecx, 0x30
0xb8f5: mov     esi, r14d
0xb8f8: mov     rdi, r15
0xb8fb: mov     word ptr [rsp + 0x19], cx
0xb900: call    0x4930
0xb905: mov     ebp, eax
0xb907: cmp     eax, -1
0xb90a: jne     0xb89b
0xb90c: mov     dword ptr [rip + 0x1faea], 1
0xb916: cmp     ebx, 2
0xb919: je      0xba30
0xb91f: mov     rax, qword ptr [rip + 0x1f06a]
0xb926: mov     r15, rsp
0xb929: mov     esi, r14d
0xb92c: mov     rdi, r15
0xb92f: movzx   edx, byte ptr [rax + 8]
0xb933: movzx   eax, byte ptr [rax + 9]
0xb937: mov     byte ptr [rsp + 8], dl
0xb93b: mov     byte ptr [rsp + 9], al
0xb93f: call    0x4930
0xb944: mov     ebp, eax
0xb946: cmp     eax, -1

```

```

0xb949: je      0xba00
0xb94f: mov     edx, dword ptr [rip + 0x1f043]
0xb955: mov     edi, ebp
0xb957: mov     rsi, r12
0xb95a: xor     eax, eax
0xb95c: imul    edx, dword ptr [rip + 0x1f039]
0xb963: add     edx, 4
0xb966: call    r13
0xb969: mov     edi, ebp
0xb96b: call    0x4ab0
0xb970: jmp     0xb8c6
0xb8c6:
0xb975: nop     dword ptr [rax]
0xb978: mov     r13, qword ptr [rip + 0x1d661]
0xb97f: xor     r14d, r14d
0xb982: jmp     0xb863
0xb863:
0xb987: nop     word ptr [rax + rax]
0xb990: movsx   esi, byte ptr [r12 + 2]
0xb996: movsx   edi, byte ptr [r12 + 3]
0xb99c: call    0xa7a0
0xb9a1: mov     edx, dword ptr [rip + 0x1f561]
0xb9a7: test    edx, edx
0xb9a9: jle     0xb9e5
0xb9ab: xor     ebp, ebp
0xb9ad: lea     rbx, [rip + 0x1f56c]
0xb9b4: jmp     0xb9cc
0xb9cc:
0xb9b6: nop     word ptr cs:[rax + rax]
0xb9c0: mov     edx, dword ptr [rip + 0x1f542]
0xb9c6: add     ebp, eax
0xb9c8: cmp     edx, ebp
0xb9ca: jle     0xb9e5
0xb9cc: sub     edx, ebp
0xb9ce: movsxd  rsi, ebp
0xb9d1: mov     edi, 1
0xb9d6: add     rsi, rbx
0xb9d9: movsxd  rdx, edx
0xb9dc: call    0x18da0
0xb9e1: test    eax, eax
0xb9e3: jns     0xb9c0
0xb9e5: mov     dword ptr [rip + 0x1f519], 0
0xb9ef: mov     eax, 1
0xb9f4: jmp     0xb8cb
0xb8cb:
0xb9f9: nop     dword ptr [rax]
0xba00: mov     edx, 0x30
0xba05: mov     esi, r14d

```

```

0xba08: mov     rdi, r15
0xba0b: mov     word ptr [rsp + 8], dx
0xba10: call    0x4930
0xba15: mov     ebp, eax
0xba17: cmp     eax, -1
0xba1a: jne     0xb94f
0xba20: xor     eax, eax
0xba22: jmp     0xb8cb
0xb8cb
0xba27: nop     word ptr [rax + rax]
0xba30: xor     eax, eax
0xba32: mov     r15, rsp
0xba35: call    0xa300
0xba3a: mov     rax, qword ptr [rip + 0x1ef4f]
0xba41: mov     esi, r14d
0xba44: mov     rdi, r15
0xba47: movzx   edx, byte ptr [rax + 8]
0xba4b: movzx   eax, byte ptr [rax + 9]
0xba4f: mov     byte ptr [rsp + 8], dl
0xba53: mov     byte ptr [rsp + 9], al
0xba57: call    0x4930
0xba5c: mov     ebp, eax
0xba5e: cmp     eax, -1
0xba61: je      0xbae0
0xba63: mov     edx, dword ptr [rip + 0x1ef2f]
0xba69: mov     rsi, r12
0xba6c: mov     edi, ebp
0xba6e: xor     eax, eax
0xba70: imul    edx, dword ptr [rip + 0x1ef25]
0xba77: add     edx, 4
0xba7a: call    r13
0xba7d: mov     edi, ebp
0xba7f: call    0x4ab0
0xba84: mov     eax, dword ptr [rip + 0x1ef12]
0xba8a: xor     esi, esi
0xba8c: lea     edi, [rax - 1]
0xba8f: call    0xa7a0
0xba94: mov     edx, dword ptr [rip + 0x1f46e]
0xba9a: test    edx, edx
0xba9c: jle     0xb9e5
0xbaa2: xor     ebp, ebp
0xbaa4: lea     rbx, [rip + 0x1f475]
0xbaab: jmp     0xbac0
0xbac0
0xbaad: nop     dword ptr [rax]
0xbab0: mov     edx, dword ptr [rip + 0x1f452]
0xbab6: add     ebp, eax
0xbab8: cmp     edx, ebp

```



```

0xbaba: jle      0xb9e5
0xbac0: sub      edx, ebp
0xbac2: movsxd   rsi, ebp
0xbac5: mov      edi, 1
0xbaca: add      rsi, rbx
0xbacd: movsxd   rdx, edx
0xbad0: call     0x18da0
0xbad5: test     eax, eax
0xbad7: jns      0xbab0
0xbad9: jmp      0xb9e5
0xb9e5:
0xbade: nop
0xbae0: mov      eax, 0x30
0xbae5: mov      esi, r14d
0xbae8: mov      rdi, r15
0xbaeb: mov      word ptr [rsp + 8], ax
0xbaf0: call     0x4930
0xbaf5: mov      ebp, eax
0xbaf7: cmp      eax, -1
0xbafa: jne      0xba63
0xbb00: xor      eax, eax
0xbb02: jmp      0xb8cb
0xb8cb:
0xbb07: call     0x49d0
0xbb0c: nop      dword ptr [rax]
0xbb10: endbr64
0xbb14: mov      eax, dword ptr [rip + 0x1ee6e]
0xbb1a: test     eax, eax
0xbb1c: jne      0xbb20
0xbb1e: ret
0xbb1f: nop
0xbb20: sub      rsp, 8
0xbb24: mov      esi, 1
0xbb29: xor      eax, eax
0xbb2b: call     0xb7f0
0xbb30: mov      dword ptr [rip + 0x1f8ce], eax
0xbb36: add      rsp, 8
0xbb3a: ret
0xbb3b: nop      dword ptr [rax + rax]
0xbb40: endbr64
0xbb44: push     rbp
0xbb45: xor      eax, eax
0xbb47: mov      rbp, rdi
0xbb4a: call     0xa6c0
0xbb4f: mov      edx, dword ptr [rip + 0x1ee33]
0xbb55: test     edx, edx
0xbb57: je       0xbb94
0xbb59: mov      eax, dword ptr [rip + 0x1f8a5]

```

```

0xbb5f: test    eax, eax
0xbb61: je      0xbb94
0xbb63: mov     edx, dword ptr [rip + 0x1ee33]
0xbb69: imul    edx, dword ptr [rip + 0x1ee28]
0xbb70: xor     esi, esi
0xbb72: mov     rdi, qword ptr [rip + 0x1f7af]
0xbb79: movsxd  rdx, edx
0xbb7c: call    0x4a50
0xbb81: xor     eax, eax
0xbb83: mov     esi, 2
0xbb88: mov     rdi, rbp
0xbb8b: call    0xb7f0
0xbb90: test    eax, eax
0xbb92: jne     0xbba0
0xbb94: xor     eax, eax
0xbb96: pop     rbp
0xbb97: jmp     0xa300
0xa300:
0xbb9c: nop     dword ptr [rax]
0xbba0: mov     edx, dword ptr [rip + 0x1edf6]
0xbba6: imul    edx, dword ptr [rip + 0x1edeb]
0xbbad: xor     esi, esi
0xbbae: mov     rdi, qword ptr [rip + 0x1f782]
0xbbb6: pop     rbp
0xbbb7: movsxd  rdx, edx
0xbbba: jmp     0x4a50
0x4a50:
0xbbbf: nop
0xbbc0: endbr64
0xbbc4: push    rbp
0xbbc5: mov     rbp, rdi
0xbbc8: push    rbx
0xbbc9: sub     rsp, 8
0xbbcd: cmp     dword ptr [rip + 0x1ed84], 1
0xbbd4: je      0xbcd0
0xbbda: xor     eax, eax
0xbbdc: call    0xa6c0
0xbbe1: test    rbp, rbp
0xbbe4: je      0xbbf4
0xbbe6: mov     eax, dword ptr [rip + 0x1ed9c]
0xbbec: test    eax, eax
0xbbee: jne     0xbcc0
0xbbf4: mov     eax, dword ptr [rip + 0x1eda2]
0xbbfa: mov     rdi, qword ptr [rip + 0x1e347]
0xbc01: lea     rdx, [rip - 0x2198]
0xbc08: lea     esi, [rax - 1]
0xbc0b: call    0x4820
0xbc10: mov     edx, dword ptr [rip + 0x1f2f2]

```

```

0xbc16: test    edx, edx
0xbc18: jle      0xbc4d
0xbc1a: xor      ebp, ebp
0xbc1c: lea      rbx, [rip + 0x1f2fd]
0xbc23: jmp      0xbc34
0xbc34:
0xbc25: nop      dword ptr [rax]
0xbc28: mov      edx, dword ptr [rip + 0x1f2da]
0xbc2e: add      ebp, eax
0xbc30: cmp      edx, ebp
0xbc32: jle      0xbc4d
0xbc34: sub      edx, ebp
0xbc36: movsxd   rsi, ebp
0xbc39: mov      edi, 1
0xbc3e: add      rsi, rbx
0xbc41: movsxd   rdx, edx
0xbc44: call     0x18da0
0xbc49: test     eax, eax
0xbc4b: jns      0xbc28
0xbc4d: mov      edi, dword ptr [rip + 0x1ed49]
0xbc53: xor      esi, esi
0xbc55: mov      dword ptr [rip + 0x1f2a9], 0
0xbc5f: call     0xa7a0
0xbc64: mov      edx, dword ptr [rip + 0x1f29e]
0xbc6a: test     edx, edx
0xbc6c: jle      0xbca5
0xbc6e: xor      ebp, ebp
0xbc70: lea      rbx, [rip + 0x1f2a9]
0xbc77: jmp      0xbc8c
0xbc8c:
0xbc79: nop      dword ptr [rax]
0xbc80: mov      edx, dword ptr [rip + 0x1f282]
0xbc86: add      ebp, eax
0xbc88: cmp      edx, ebp
0xbc8a: jle      0xbca5
0xbc8c: sub      edx, ebp
0xbc8e: movsxd   rsi, ebp
0xbc91: mov      edi, 1
0xbc96: add      rsi, rbx
0xbc99: movsxd   rdx, edx
0xbc9c: call     0x18da0
0xbca1: test     eax, eax
0xbca3: jns      0xbc80
0xbca5: mov      dword ptr [rip + 0x1f259], 0
0xbcaf: add      rsp, 8
0xbcb3: mov      edi, 0xa
0xbcb8: pop      rbx
0xbcb9: pop      rbp

```

```

0xbcba: jmp      0x4810
0x4810
0xbcbf: nop
0xbcc0: mov     rdi, rbp
0xbcc3: call    0xbb40
0xbcc8: jmp     0xbbf4
0bbf4
0xbccd: nop     dword ptr [rax]
0xbcd0: lea     rdx, [rip + 0x1f6e9]
0xbcd7: mov     esi, 1
0xbcdc: mov     edi, 1
0xbce1: call    0x4d70
0xbce6: mov     dword ptr [rip + 0x1ec68], 0
0xbcf0: jmp     0xbbda
0bbda
0xbcf5: nop     word ptr cs:[rax + rax]
0xbd00: endbr64
0xbd04: push    r13
0xbd06: lea     r13, [rip + 0x1cc33]
0xbd0d: push    r12
0xbd0f: push    rbp
0xbd10: mov     rbp, rdi
0xbd13: lea     rdi, [rip + 0x14f14]
0xbd1a: push    rbx
0xbd1b: xor     ebx, ebx
0xbd1d: sub     rsp, 8
0xbd21: jmp     0xbd2d
0xbd2d
0xbd23: nop     dword ptr [rax + rax]
0xbd28: mov     rdi, qword ptr [r13 + rbx*8]
0xbd2d: mov     rsi, rbp
0xbd30: mov     r12d, ebx
0xbd33: call    0x4b30
0xbd38: test    eax, eax
0xbd3a: je      0xbd60
0xbd3c: add     rbx, 1
0xbd40: cmp     rbx, 0xa
0xbd44: jne     0xbd28
0xbd46: add     rsp, 8
0xbd4a: mov     r12d, 0xffffffff
0xbd50: pop     rbx
0xbd51: mov     eax, r12d
0xbd54: pop     rbp
0xbd55: pop     r12
0xbd57: pop     r13
0xbd59: ret
0xbd5a: nop     word ptr [rax + rax]
0xbd60: lea     eax, [rbx - 8]

```

```

0xbd63: cmp     ebx, 7
0xbd66: cmovg    r12d, eax
0xbd6a: add     rsp, 8
0xbd6e: pop     rbx
0xbd6f: pop     rbp
0xbd70: mov     eax, r12d
0xbd73: pop     r12
0xbd75: pop     r13
0xbd77: ret
0xbd78: nop     dword ptr [rax + rax]
0xbd80: endbr64
0xbd84: push    r12
0xbd86: mov     r12, rdi
0xbd89: push    rbp
0xbd8a: lea     rbp, [rip + 0x1e1e7]
0xbd91: push    rbx
0xbd92: mov     ebx, 0x11
0xbd97: jmp     0xbdac
0xbdac:
0xbd99: nop     dword ptr [rax]
0xbda0: add     ebx, 1
0xbda3: add     rbp, 0x20
0xbda7: cmp     ebx, 0x26
0xbdaa: je      0xbdd8
0xbdac: mov     rdi, qword ptr [rbp]
0xbdb0: mov     rsi, r12
0xbdb3: call    0x4b30
0xbdb8: test    eax, eax
0xbdba: jne     0xbda0
0xbdbc: movsxd  rbx, ebx
0xbdbf: lea     rax, [rip + 0x1df7a]
0xbdc6: shl     rbx, 5
0xbdca: mov     rax, qword ptr [rax + rbx + 8]
0xbdcf: pop     rbx
0xbdd0: pop     rbp
0xbdd1: pop     r12
0xbdd3: ret
0xbdd4: nop     dword ptr [rax]
0xbdd8: pop     rbx
0xbdd9: xor     eax, eax
0xbddb: pop     rbp
0xbddc: pop     r12
0xbdde: ret
0xbddf: nop
0xbde0: endbr64
0xbde4: push    r14
0xbde6: lea     rdi, [rip + 0x1479d]
0xbded: push    r13

```

```

0xbdef: push    r12
0xbdf1: push    rbp
0xbdf2: push    rbx
0xbdf3: sub     rsp, 0xb0
0xbdfa: mov     rax, qword ptr fs:[0x28]
0xbe03: mov     qword ptr [rsp + 0xa8], rax
0xbe0b: xor     eax, eax
0xbe0d: call    0x47e0
0xbe12: lea     rsi, [rsp + 0x10]
0xbe17: mov     edi, 1
0xbe1c: mov     rbp, rax
0xbe1f: call    0x4eb0
0xbe24: mov     rax, qword ptr [rsp + 0x38]
0xbe29: shr     rax, 8
0xbe2d: cmp     rax, 4
0xbe31: sete    al
0xbe34: movzx   eax, al
0xbe37: mov     dword ptr [rip + 0x1eb4b], eax
0xbe3d: test    rbp, rbp
0xbe40: je      0xc36d
0xbe46: mov     rdi, rbp
0xbe49: call    0x49b0
0xbe4e: cmp     rax, 0x3f
0xbe52: ja      0xc308
0xbe58: lea     r12, [rip + 0x14ff1]
0xbe5f: lea     rbx, [rip + 0x14d6f]
0xbe66: lea     r14, [rip + 0x14e02]
0xbe6d: lea     r13, [rip + 0x14e07]
0xbe74: nop     dword ptr [rax]
0xbe78: xor     edi, edi
0xbe7a: mov     rsi, rbp
0xbe7d: call    0x4e90
0xbe82: cmp     eax, -1
0xbe85: je      0xbf38
0xbe8b: test    eax, eax
0xbe8d: jne     0xbff8
0xbe93: mov     rdi, qword ptr [rip + 0x1e9e6]
0xbe9a: mov     esi, 1
0xbe9f: mov     r8, rbp
0xbea2: mov     rdx, r12
0xbea5: mov     rcx, qword ptr [rip + 0x1ea3c]
0xbeac: lea     r9, [rip + 0x1de5d]
0xbeb3: call    0x4e60
0xbeb8: mov     rsi, rbx
0xbebb: mov     rdi, rbp
0xbebe: call    0x4b30
0xbec3: test    eax, eax
0xbec5: je      0xbf98

```

```

0xbeeb: mov     rdi, rbp
0xbece: call    0x9940
0xbed3: test    eax, eax
0xbed5: jne     0xbfc8
0xbedb: mov     rsi, r14
0xbede: mov     rdi, rbp
0xbee1: call    0x4b30
0xbee6: test    eax, eax
0xbee8: je      0xbf01
0xbeea: lea     rsi, [rip + 0x14d84]
0xbef1: mov     rdi, rbp
0xbef4: call    0x4b30
0xbef9: test    eax, eax
0xbefb: jne     0xc3b4
0xbf01: mov     rdi, qword ptr [rip + 0x1e978]
0xbf08: mov     rcx, qword ptr [rip + 0x1e9d9]
0xbf0f: mov     esi, 1
0xbf14: xor     eax, eax
0xbf16: mov     rdx, r13
0xbf19: lea     rbp, [rip + 0x14cba]
0xbf20: call    0x4e60
0xbf25: xor     edi, edi
0xbf27: mov     rsi, rbp
0xbf2a: call    0x4e90
0xbf2f: cmp     eax, -1
0xbf32: jne     0xbe8b
0xbf38: mov     rcx, qword ptr [rip + 0x1e9a9]
0xbf3f: mov     rdi, qword ptr [rip + 0x1e93a]
0xbf46: xor     eax, eax
0xbf48: lea     r8, [rip + 0x1ddc1]
0xbf4f: lea     rdx, [rip + 0x14ea2]
0xbf56: mov     esi, 1
0xbf5b: call    0x4e60
0xbf60: mov     rdi, qword ptr [rip + 0x1e919]
0xbf67: xor     eax, eax
0xbf69: mov     rcx, qword ptr [rip + 0x1e978]
0xbf70: lea     r8, [rip + 0x1dd89]
0xbf77: lea     rdx, [rip + 0x14ea2]
0xbf7e: mov     esi, 1
0xbf83: call    0x4e60
0xbf88: mov     edi, 1
0xbf8d: call    0x4e20
0xbf92: nop     word ptr [rax + rax]
0xbf98: mov     rcx, qword ptr [rip + 0x1e949]
0xbf9f: mov     rdi, qword ptr [rip + 0x1e8da]
0xbfa6: lea     rdx, [rip + 0x14c99]
0xbfad: mov     esi, 1
0xbfb2: lea     rbp, [rip + 0x14c21]

```

```

0xbfb9: call    0x4e60
0xbfbe: jmp     0xbe78
0xbe78:
0xbfc3: nop     dword ptr [rax + rax]
0xbfc8: mov     rcx, qword ptr [rip + 0x1e919]
0xbfcf: mov     rdi, qword ptr [rip + 0x1e8aa]
0xbfd6: lea     rdx, [rip + 0x14c7d]
0xbfdd: xor     eax, eax
0xbfdf: mov     esi, 1
0xbfe4: lea     rbp, [rip + 0x14bf6]
0xbfeb: call    0x4e60
0xbff0: jmp     0xbe78
0xbe78:
0xbff5: nop     dword ptr [rax]
0xbff8: mov     rdi, rbp
0xbffb: lea     r13, [rsp + 8]
0xc000: call    0x1d7a0
0xc005: mov     edi, 0x800
0xc00a: mov     qword ptr [rip + 0x1e93f], rax
0xc011: call    0x1d670
0xc016: mov     rdi, qword ptr [rip + 0x1de03]
0xc01d: mov     rsi, r13
0xc020: mov     qword ptr [rsp + 8], rax
0xc025: call    0x4ca0
0xc02a: xor     edx, edx
0xc02c: test    rax, rax
0xc02f: je      0xc034
0xc031: movzx   edx, byte ptr [rax]
0xc034: mov     rdi, qword ptr [rip + 0x1de25]
0xc03b: mov     rsi, r13
0xc03e: mov     byte ptr [rip + 0x1e904], dl
0xc044: call    0x4ca0
0xc049: mov     rdi, qword ptr [rip + 0x1ddf0]
0xc050: mov     rsi, r13
0xc053: mov     qword ptr [rip + 0x1e8e6], rax
0xc05a: call    0x4ca0
0xc05f: cmp     qword ptr [rip + 0x1e8d9], 0
0xc067: mov     qword ptr [rip + 0x1e8ca], rax
0xc06e: je      0xc2c0
0xc074: test    rax, rax
0xc077: je      0xc2c0
0xc07d: mov     rdi, qword ptr [rip + 0x1dcbc]
0xc084: mov     rsi, r13
0xc087: call    0x4ca0
0xc08c: mov     rdi, qword ptr [rip + 0x1dced]
0xc093: mov     rsi, r13
0xc096: mov     qword ptr [rip + 0x1dcab], rax
0xc09d: call    0x4ca0

```



```

0xc0a2: mov     rdi, qword ptr [rip + 0x1dcb7]
0xc0a9: mov     rsi, r13
0xc0ac: mov     qword ptr [rip + 0x1dcd5], rax
0xc0b3: call    0x4ca0
0xc0b8: cmp     qword ptr [rip + 0x1dc88], 0
0xc0c0: mov     qword ptr [rip + 0x1dca1], rax
0xc0c7: je      0xc352
0xc0cd: mov     rdi, qword ptr [rip + 0x1ddac]
0xc0d4: mov     rsi, r13
0xc0d7: call    0x4ca0
0xc0dc: xor     edx, edx
0xc0de: mov     qword ptr [rip + 0x1dda3], rax
0xc0e5: test    rax, rax
0xc0e8: je      0xc0ff
0xc0ea: mov     rdi, qword ptr [rip + 0x1ddcf]
0xc0f1: mov     qword ptr [rip + 0x1dc70], rax
0xc0f8: call    0x4c90
0xc0fd: mov     edx, eax
0xc0ff: mov     rdi, qword ptr [rip + 0x1dd9a]
0xc106: mov     dword ptr [rip + 0x1ddc4], edx
0xc10c: call    0x4c50
0xc111: mov     dword ptr [rip + 0x1dd99], eax
0xc117: test    eax, eax
0xc119: js      0xc13c
0xc11b: mov     qword ptr [rip + 0x1dc62], 0
0xc126: mov     qword ptr [rip + 0x1dc37], 0
0xc131: mov     qword ptr [rip + 0x1dc0c], 0
0xc13c: mov     rdi, qword ptr [rip + 0x1dd9d]
0xc143: call    0x4c50
0xc148: mov     rdi, qword ptr [rip + 0x1ddb1]
0xc14f: mov     dword ptr [rip + 0x1dd9b], eax
0xc155: call    0x4c50
0xc15a: mov     rdi, qword ptr [rip + 0x1dc3f]
0xc161: mov     rsi, r13
0xc164: mov     dword ptr [rip + 0x1dda6], eax
0xc16a: call    0x4ca0
0xc16f: mov     rdi, qword ptr [rip + 0x1dc4a]
0xc176: mov     rsi, r13
0xc179: mov     qword ptr [rip + 0x1dc28], rax
0xc180: call    0x4ca0
0xc185: cmp     qword ptr [rip + 0x1dc1b], 0
0xc18d: mov     qword ptr [rip + 0x1dc34], rax
0xc194: je      0xc2a0
0xc19a: test    rax, rax
0xc19d: je      0xc2a0
0xc1a3: mov     rdi, qword ptr [rip + 0x1dc36]
0xc1aa: mov     rsi, r13
0xc1ad: lea     rbp, [rip + 0x1ddac]

```

```

0xc1b4: lea    rbx, [rbp - 0x220]
0xc1bb: call   0x4ca0
0xc1c0: mov     rdi, qword ptr [rip + 0x1dc39]
0xc1c7: mov     rsi, r13
0xc1ca: lea     r12, [rbx + 0x4c0]
0xc1d1: mov     qword ptr [rip + 0x1dc10], rax
0xc1d8: call   0x4ca0
0xc1dd: mov     rdi, qword ptr [rip + 0x1dd3c]
0xc1e4: mov     rsi, r13
0xc1e7: mov     qword ptr [rip + 0x1dc1a], rax
0xc1ee: call   0x4ca0
0xc1f3: mov     rdi, qword ptr [rip + 0x1dd46]
0xc1fa: mov     rsi, r13
0xc1fd: mov     qword ptr [rip + 0x1dd24], rax
0xc204: call   0x4ca0
0xc209: mov     qword ptr [rip + 0x1dd38], rax
0xc210: mov     rdi, qword ptr [rbp]
0xc214: mov     rsi, r13
0xc217: add     rbp, 0x20
0xc21b: call   0x4ca0
0xc220: mov     qword ptr [rbp - 0x18], rax
0xc224: cmp     rbp, r12
0xc227: jne     0xc210
0xc229: xor     ebp, ebp
0xc22b: lea     r13, [rip + 0x161b3]
0xc232: lea     r14, [rip + 0x14c57]
0xc239: jmp     0xc249
0xc249:
0xc23b: nop     dword ptr [rax + rax]
0xc240: add     rbx, 0x20
0xc244: cmp     rbx, r12
0xc247: je      0xc268
0xc249: cmp     qword ptr [rbx + 8], 0
0xc24e: jne     0xc240
0xc250: mov     eax, dword ptr [rbx + 0x14]
0xc253: test    eax, eax
0xc255: jne     0xc2e0
0xc25b: mov     qword ptr [rbx + 8], r13
0xc25f: add     rbx, 0x20
0xc263: cmp     rbx, r12
0xc266: jne     0xc249
0xc268: test    ebp, ebp
0xc26a: jne     0xc390
0xc270: mov     rax, qword ptr [rsp + 0xa8]
0xc278: sub     rax, qword ptr fs:[0x28]
0xc281: jne     0xc3be
0xc287: add     rsp, 0xb0
0xc28e: pop     rbx

```

```

0xc28f: pop     rbp
0xc290: pop     r12
0xc292: pop     r13
0xc294: pop     r14
0xc296: ret
0xc297: nop     word ptr [rax + rax]
0xc2a0: mov     qword ptr [rip + 0x1dafd], 0
0xc2ab: mov     qword ptr [rip + 0x1db12], 0
0xc2b6: jmp     0xc1a3
0xc1a3
0xc2bb: nop     dword ptr [rax + rax]
0xc2c0: mov     qword ptr [rip + 0x1e66d], 0
0xc2cb: mov     qword ptr [rip + 0x1e66a], 0
0xc2d6: jmp     0xc07d
0xc07d
0xc2db: nop     dword ptr [rax + rax]
0xc2e0: mov     r8, qword ptr [rbx]
0xc2e3: mov     rcx, qword ptr [rip + 0x1e5fe]
0xc2ea: mov     rdx, r14
0xc2ed: xor     eax, eax
0xc2ef: mov     rdi, qword ptr [rip + 0x1e58a]
0xc2f6: mov     esi, 1
0xc2fb: add     ebp, 1
0xc2fe: call    0x4e60
0xc303: jmp     0xc240
0xc240
0xc308: mov     rcx, qword ptr [rip + 0x1e5d9]
0xc30f: mov     rdi, qword ptr [rip + 0x1e56a]
0xc316: lea     rdx, [rip + 0x14aab]
0xc31d: xor     eax, eax
0xc31f: mov     esi, 1
0xc324: call    0x4e60
0xc329: mov     rcx, qword ptr [rip + 0x1e550]
0xc330: mov     edx, 0x11
0xc335: mov     esi, 1
0xc33a: lea     rdi, [rip + 0x148f3]
0xc341: lea     rbp, [rip + 0x1d9d1]
0xc348: call    0x4e40
0xc34d: jmp     0xbe58
0xbe58
0xc352: mov     qword ptr [rip + 0x1da2b], 0
0xc35d: mov     qword ptr [rip + 0x1da00], 0
0xc368: jmp     0xc0cd
0xc0cd
0xc36d: mov     rcx, qword ptr [rip + 0x1e574]
0xc374: mov     rdi, qword ptr [rip + 0x1e505]
0xc37b: lea     rdx, [rip + 0x14a16]
0xc382: xor     eax, eax

```

```

0xc384: mov     esi, 1
0xc389: call    0x4e60
0xc38e: jmp     0xc329
0xc329
0xc390: mov     rcx, qword ptr [rip + 0x1e551]
0xc397: mov     rdi, qword ptr [rip + 0x1e4e2]
0xc39e: mov     r8d, ebp
0xc3a1: xor     eax, eax
0xc3a3: lea     rdx, [rip + 0x14b16]
0xc3aa: mov     esi, 1
0xc3af: call    0x4e60
0xc3b4: mov     edi, 1
0xc3b9: call    0x4e20
0xc3be: call    0x49d0
0xc3c3: nop     word ptr cs:[rax + rax]
0xc3ce: nop
0xc3d0: endbr64
0xc3d4: push    rbx
0xc3d5: mov     ebx, edi
0xc3d7: xor     edi, edi
0xc3d9: call    0x48f0
0xc3de: test    eax, eax
0xc3e0: je      0xc453
0xc3e2: mov     edi, 1
0xc3e7: call    0x48f0
0xc3ec: test    eax, eax
0xc3ee: je      0xc453
0xc3f0: mov     edi, 1
0xc3f5: call    0x4cd0
0xc3fa: mov     qword ptr [rip + 0x1e58f], rax
0xc401: test    rax, rax
0xc404: je      0xc47e
0xc406: lea     rsi, [rip + 0x1efb3]
0xc40d: mov     edi, 1
0xc412: call    0x4d60
0xc417: mov     edi, 0x40
0xc41c: call    0x1d670
0xc421: mov     dword ptr [rip + 0x1efd5], ebx
0xc427: pop     rbx
0xc428: mov     qword ptr [rip + 0x1ef31], rax
0xc42f: mov     qword ptr [rip + 0x1e52a], rax
0xc436: xor     eax, eax
0xc438: mov     qword ptr [rip + 0x1e52d], 0
0xc443: mov     qword ptr [rip + 0x1e51a], 0
0xc44e: jmp     0xbde0
0xbde0
0xc453: mov     rdi, qword ptr [rip + 0x1e426]
0xc45a: mov     rcx, qword ptr [rip + 0x1e487]

```

```

0xc461: lea     rdx, [rip + 0x14a88]
0xc468: xor     eax, eax
0xc46a: mov     esi, 1
0xc46f: call    0x4e60
0xc474: mov     edi, 1
0xc479: call    0x4e20
0xc47e: mov     rdi, qword ptr [rip + 0x1e3fb]
0xc485: mov     rcx, qword ptr [rip + 0x1e45c]
0xc48c: mov     esi, 1
0xc491: lea     rdx, [rip + 0x147f8]
0xc498: call    0x4e60
0xc49d: mov     edi, 1
0xc4a2: call    0x4e20
0xc4a7: nop     word ptr [rax + rax]
0xc4b0: endbr64
0xc4b4: push    r12
0xc4b6: push    rbp
0xc4b7: mov     rbp, rdi
0xc4ba: mov     rdi, qword ptr [rip + 0x1e48f]
0xc4c1: push    rbx
0xc4c2: call    0x9940
0xc4c7: test    eax, eax
0xc4c9: jne     0xc4d0
0xc4cb: pop     rbx
0xc4cc: pop     rbp
0xc4cd: pop     r12
0xc4cf: ret
0xc4d0: mov     rdi, rbp
0xc4d3: call    0x49b0
0xc4d8: lea     rdi, [rax + 0x81]
0xc4df: mov     rbx, rax
0xc4e2: call    0x1d670
0xc4e7: mov     rdi, rbp
0xc4ea: mov     r12, rax
0xc4ed: call    0x1d7a0
0xc4f2: mov     rsi, rbx
0xc4f5: add     ebx, 0x7f
0xc4f8: mov     rbp, rax
0xc4fb: mov     rdi, rax
0xc4fe: call    0x184d0
0xc503: push    7
0xc505: lea     rcx, [rip + 0x147a2]
0xc50c: xor     eax, eax
0xc50e: push    rbp
0xc50f: mov     rdx, -1
0xc516: mov     esi, 1
0xc51b: mov     rdi, r12
0xc51e: lea     r9, [rip + 0x14040]

```

```

0xc525: mov     r8d, 0x1b
0xc52b: call    0x4f10
0xc530: mov     eax, dword ptr [rip + 0x1e462]
0xc536: pop     rdx
0xc537: pop     rcx
0xc538: add     eax, eax
0xc53a: cmp     ebx, eax
0xc53c: jl      0xc555
0xc53e: cdqe
0xc540: mov     byte ptr [r12 + rax], 7
0xc545: mov     eax, dword ptr [rip + 0x1e44d]
0xc54b: add     eax, eax
0xc54d: cdqe
0xc54f: mov     byte ptr [r12 + rax + 1], 0
0xc555: mov     rdi, r12
0xc558: call    0x49b0
0xc55d: mov     rsi, r12
0xc560: mov     edi, 1
0xc565: mov     rdx, rax
0xc568: call    0x18da0
0xc56d: mov     rdi, rbp
0xc570: call    0x18d10
0xc575: mov     rdi, r12
0xc578: call    0x18d10
0xc57d: mov     rdi, qword ptr [rip + 0x1e2bc]
0xc584: pop     rbx
0xc585: pop     rbp
0xc586: pop     r12
0xc588: jmp     0x4c40
0x4c40:
0xc58d: nop     dword ptr [rax]
0xc590: endbr64
0xc594: sub     rsp, 8
0xc598: mov     edi, 0x18
0xc59d: call    0x1d670
0xc5a2: mov     qword ptr [rax], 0
0xc5a9: mov     qword ptr [rax + 8], 0
0xc5b1: add     rsp, 8
0xc5b5: ret
0xc5b6: nop     word ptr cs:[rax + rax]
0xc5c0: endbr64
0xc5c4: test    rdi, rdi
0xc5c7: je      0xc5d0
0xc5c9: jmp     0x18d10
0x18d10:
0xc5ce: nop
0xc5d0: ret
0xc5d1: nop     word ptr cs:[rax + rax]

```

```

0xc5dc: nop      dword ptr [rax]
0xc5e0: endbr64
0xc5e4: mov      dword ptr [rdi], esi
0xc5e6: mov      dword ptr [rdi + 4], edx
0xc5e9: mov      dword ptr [rdi + 8], ecx
0xc5ec: mov      dword ptr [rdi + 0xc], r8d
0xc5f0: ret
0xc5f1: nop      word ptr cs:[rax + rax]
0xc5fc: nop      dword ptr [rax]
0xc600: endbr64
0xc604: mov      rax, rdi
0xc607: mov      rdi, rsi
0xc60a: mov      esi, edx
0xc60c: mov      ecx, dword ptr [rax + 0x10]
0xc60f: lea      r8d, [rcx + rdx]
0xc613: mov      edx, dword ptr [rax + 0xc]
0xc616: mov      dword ptr [rax + 0x10], r8d
0xc61a: cmp      edx, ecx
0xc61c: jle      0xc640
0xc61e: mov      r9d, dword ptr [rax + 8]
0xc622: cmp      dword ptr [rax + 0x14], r9d
0xc626: jge      0xc640
0xc628: cmp      r8d, edx
0xc62b: jle      0xc631
0xc62d: sub      edx, ecx
0xc62f: mov      esi, edx
0xc631: jmp      0xa1a0
0xa1a0
0xc636: nop      word ptr cs:[rax + rax]
0xc640: xor      eax, eax
0xc642: ret
0xc643: nop      word ptr cs:[rax + rax]
0xc64e: nop
0xc650: endbr64
0xc654: mov      ecx, dword ptr [rdi + 0x10]
0xc657: lea      edx, [rcx + 1]
0xc65a: mov      dword ptr [rdi + 0x10], edx
0xc65d: cmp      edx, dword ptr [rdi + 0xc]
0xc660: jg       0xc66a
0xc662: mov      ecx, dword ptr [rdi + 8]
0xc665: cmp      dword ptr [rdi + 0x14], ecx
0xc668: jl       0xc670
0xc66a: xor      eax, eax
0xc66c: ret
0xc66d: nop      dword ptr [rax]
0xc670: mov      edi, esi
0xc672: jmp      0xa230
0xa230

```

```

0xc677: nop      word ptr [rax + rax]
0xc680: endbr64
0xc684: mov      rax, rdi
0xc687: mov      edi, esi
0xc689: mov      dword ptr [rax + 0x10], edx
0xc68c: add      edx, dword ptr [rax]
0xc68e: mov      dword ptr [rax + 0x14], esi
0xc691: add      edi, dword ptr [rax + 4]
0xc694: mov      esi, edx
0xc696: jmp      0xa4b0
0xa4b0
0xc69b: nop      dword ptr [rax + rax]
0xc6a0: endbr64
0xc6a4: mov      eax, dword ptr [rdi]
0xc6a6: ret
0xc6a7: nop      word ptr [rax + rax]
0xc6b0: endbr64
0xc6b4: mov      eax, dword ptr [rdi + 4]
0xc6b7: ret
0xc6b8: nop      dword ptr [rax + rax]
0xc6c0: endbr64
0xc6c4: mov      eax, dword ptr [rdi + 8]
0xc6c7: ret
0xc6c8: nop      dword ptr [rax + rax]
0xc6d0: endbr64
0xc6d4: mov      eax, dword ptr [rdi + 0xc]
0xc6d7: ret
0xc6d8: nop      dword ptr [rax + rax]
0xc6e0: push     r14
0xc6e2: push     r13
0xc6e4: push     r12
0xc6e6: mov      r12, qword ptr [rip + 0x1ed1b]
0xc6ed: push     rbp
0xc6ee: push     rbx
0xc6ef: mov      r13, qword ptr [r12 + 0x20]
0xc6f4: mov      rbx, qword ptr [r12 + 0x18]
0xc6f9: cmp      r13, rbx
0xc6fc: je       0xc761
0xc6fe: mov      ebp, edi
0xc700: jbe      0xc770
0xc702: test     bpl, 1
0xc706: je       0xc75c
0xc708: mov      rdi, qword ptr [r12 + 0x60]
0xc70d: test     rdi, rdi
0xc710: je       0xc71e
0xc712: call     0x18d10
0xc717: mov      r12, qword ptr [rip + 0x1ecea]
0xc71e: mov      r14, r13

```



```

0xc721: sub    r14, rbx
0xc724: lea     rdi, [r14 + 1]
0xc728: call    0x1d670
0xc72d: mov     rdx, r14
0xc730: mov     qword ptr [r12 + 0x60], rax
0xc735: mov     rax, qword ptr [rip + 0x1eccc]
0xc73c: mov     rsi, qword ptr [rax + 0x58]
0xc740: mov     rdi, qword ptr [rax + 0x60]
0xc744: add     rsi, rbx
0xc747: call    0x4bb0
0xc74c: mov     rax, qword ptr [rip + 0x1ecb5]
0xc753: mov     rax, qword ptr [rax + 0x60]
0xc757: mov     byte ptr [rax + r14], 0
0xc75c: and     ebp, 2
0xc75f: jne     0xc788
0xc761: pop     rbx
0xc762: pop     rbp
0xc763: pop     r12
0xc765: pop     r13
0xc767: pop     r14
0xc769: ret
0xc76a: nop     word ptr [rax + rax]
0xc770: mov     rax, rbx
0xc773: mov     rbx, r13
0xc776: mov     r13, rax
0xc779: test    bpl, 1
0xc77d: je      0xc75c
0xc77f: jmp     0xc708
0xc708:
0xc781: nop     dword ptr [rax]
0xc788: mov     rax, qword ptr [rip + 0x1ec79]
0xc78f: mov     rdx, rbx
0xc792: sub     rdx, r13
0xc795: add     qword ptr [rax + 0x48], rdx
0xc799: mov     rdi, qword ptr [rax + 0x58]
0xc79d: add     qword ptr [rax + 0x38], rdx
0xc7a1: mov     qword ptr [rax + 0x18], rbx
0xc7a5: lea     rsi, [rdi + r13]
0xc7a9: add     rdi, rbx
0xc7ac: mov     qword ptr [rax + 0x20], rbx
0xc7b0: call    0x48b0
0xc7b5: mov     rbx, qword ptr [rip + 0x1ec4c]
0xc7bc: mov     rax, qword ptr [rbx + 0x38]
0xc7c0: mov     rdi, qword ptr [rbx + 0x58]
0xc7c4: lea     rsi, [rax + 1]
0xc7c8: mov     qword ptr [rbx + 0x50], rsi
0xc7cc: call    0x1d690
0xc7d1: mov     qword ptr [rbx + 0x58], rax

```

```

0xc7d5: pop    rbx
0xc7d6: pop    rbp
0xc7d7: pop    r12
0xc7d9: pop    r13
0xc7db: pop    r14
0xc7dd: ret
0xc7de: nop
0xc7e0: endbr64
0xc7e4: test   rdi, rdi
0xc7e7: je      0xc818
0xc7e9: push   rbp
0xc7ea: mov    rbp, rdi
0xc7ed: mov    rdi, qword ptr [rdi + 0x58]
0xc7f1: test   rdi, rdi
0xc7f4: je      0xc7fb
0xc7f6: call   0x18d10
0xc7fb: mov    rdi, qword ptr [rbp + 0x60]
0xc7ff: test   rdi, rdi
0xc802: je      0xc809
0xc804: call   0x18d10
0xc809: mov    rdi, rbp
0xc80c: pop    rbp
0xc80d: jmp    0x18d10
0x18d10
0xc812: nop    word ptr [rax + rax]
0xc818: ret
0xc819: nop    dword ptr [rax]
0xc820: endbr64
0xc824: mov    rax, qword ptr [rip + 0x1ebdd]
0xc82b: push   rbp
0xc82c: mov    rdi, qword ptr [rax + 0x68]
0xc830: xor    eax, eax
0xc832: call   0x1aac0
0xc837: mov    rax, qword ptr [rip + 0x1ebca]
0xc83e: mov    rdi, qword ptr [rax]
0xc841: call   0xc5c0
0xc846: mov    rbp, qword ptr [rip + 0x1ebbb]
0xc84d: test   rbp, rbp
0xc850: je      0xc876
0xc852: mov    rdi, qword ptr [rbp + 0x58]
0xc856: test   rdi, rdi
0xc859: je      0xc860
0xc85b: call   0x18d10
0xc860: mov    rdi, qword ptr [rbp + 0x60]
0xc864: test   rdi, rdi
0xc867: je      0xc86e
0xc869: call   0x18d10
0xc86e: mov    rdi, rbp

```

```

0xc871: call    0x18d10
0xc876: mov     qword ptr [rip + 0x1eb87], 0
0xc881: pop     rbp
0xc882: ret
0xc883: nop     word ptr cs:[rax + rax]
0xc88e: nop
0xc890: endbr64
0xc894: mov     rax, qword ptr [rip + 0x1eb6d]
0xc89b: movsxd  rcx, edi
0xc89e: mov     edx, esi
0xc8a0: mov     r8, rcx
0xc8a3: mov     qword ptr [rax + 0x28], rcx
0xc8a7: movsxd  rcx, esi
0xc8aa: mov     rdi, qword ptr [rax]
0xc8ad: xor     esi, esi
0xc8af: mov     qword ptr [rax + 0x30], rcx
0xc8b3: mov     ecx, 1
0xc8b8: jmp     0xc5e0
0xc5e0:
0xc8bd: nop     dword ptr [rax]
0xc8c0: endbr64
0xc8c4: push    r12
0xc8c6: mov     edi, 0x70
0xc8cb: push    rbx
0xc8cc: sub     rsp, 8
0xc8d0: call    0x1d670
0xc8d5: mov     rbx, qword ptr [rip + 0x1eb2c]
0xc8dc: mov     r12, rax
0xc8df: movdqu  xmm0, xmmword ptr [rbx]
0xc8e3: movups  xmmword ptr [rax], xmm0
0xc8e6: movdqu  xmm1, xmmword ptr [rbx + 0x10]
0xc8eb: movups  xmmword ptr [rax + 0x10], xmm1
0xc8ef: movdqu  xmm2, xmmword ptr [rbx + 0x20]
0xc8f4: movups  xmmword ptr [rax + 0x20], xmm2
0xc8f8: movdqu  xmm3, xmmword ptr [rbx + 0x30]
0xc8fd: movups  xmmword ptr [rax + 0x30], xmm3
0xc901: movdqu  xmm4, xmmword ptr [rbx + 0x40]
0xc906: movups  xmmword ptr [rax + 0x40], xmm4
0xc90a: movdqu  xmm5, xmmword ptr [rbx + 0x50]
0xc90f: movups  xmmword ptr [rax + 0x50], xmm5
0xc913: movdqu  xmm6, xmmword ptr [rbx + 0x60]
0xc918: cmp     qword ptr [rax + 0x58], 0
0xc91d: movups  xmmword ptr [rax + 0x60], xmm6
0xc921: je      0xc94a
0xc923: mov     rdi, qword ptr [rax + 0x50]
0xc927: call    0x1d670
0xc92c: mov     rdx, qword ptr [r12 + 0x50]
0xc931: mov     qword ptr [rbx + 0x58], rax

```

```

0xc935: mov     rax, qword ptr [rip + 0x1eacc]
0xc93c: mov     rsi, qword ptr [r12 + 0x58]
0xc941: mov     rdi, qword ptr [rax + 0x58]
0xc945: call    0x4bb0
0xc94a: mov     rdi, qword ptr [r12 + 0x60]
0xc94f: test    rdi, rdi
0xc952: je      0xc964
0xc954: mov     rbx, qword ptr [rip + 0x1eaad]
0xc95b: call    0x1d7a0
0xc960: mov     qword ptr [rbx + 0x60], rax
0xc964: add     rsp, 8
0xc968: mov     rax, r12
0xc96b: pop     rbx
0xc96c: pop     r12
0xc96e: ret
0xc96f: nop
0xc970: endbr64
0xc974: test    rdi, rdi
0xc977: je      0xc9f0
0xc979: push    r13
0xc97b: push    r12
0xc97d: push    rbp
0xc97e: push    rbx
0xc97f: mov     rbx, rdi
0xc982: sub     rsp, 8
0xc986: mov     rbp, qword ptr [rip + 0x1ea7b]
0xc98d: mov     rdi, qword ptr [rbp + 0x58]
0xc991: mov     r13, qword ptr [rbp + 0x28]
0xc995: mov     r12, qword ptr [rbp + 0x30]
0xc999: test    rdi, rdi
0xc99c: je      0xc9a3
0xc99e: call    0x18d10
0xc9a3: mov     rdi, qword ptr [rbp + 0x60]
0xc9a7: test    rdi, rdi
0xc9aa: je      0xc9b1
0xc9ac: call    0x18d10
0xc9b1: mov     rdi, rbp
0xc9b4: call    0x18d10
0xc9b9: movsxd  rax, r13d
0xc9bc: mov     edx, r12d
0xc9bf: movsxd  r12, r12d
0xc9c2: mov     qword ptr [rbx + 0x28], rax
0xc9c6: mov     rdi, qword ptr [rbx]
0xc9c9: mov     r8d, r13d
0xc9cc: mov     ecx, 1
0xc9d1: mov     qword ptr [rbx + 0x30], r12
0xc9d5: xor     esi, esi
0xc9d7: mov     qword ptr [rip + 0x1ea2a], rbx

```

```

0xc9de: add     rsp, 8
0xc9e2: pop     rbx
0xc9e3: pop     rbp
0xc9e4: pop     r12
0xc9e6: pop     r13
0xc9e8: jmp     0xc5e0
0xc5e0
0xc9ed: nop     dword ptr [rax]
0xc9f0: ret
0xc9f1: nop     word ptr cs:[rax + rax]
0xc9fc: nop     dword ptr [rax]
0xca00: endbr64
0xca04: mov     rdx, qword ptr [rip + 0x1e9fd]
0xca0b: mov     rax, qword ptr [rdx + 0x18]
0xca0f: sub     rax, qword ptr [rdx + 0x40]
0xca13: ret
0xca14: nop     word ptr cs:[rax + rax]
0xca1f: nop
0xca20: endbr64
0xca24: mov     rax, qword ptr [rip + 0x1e9dd]
0xca2b: mov     r8d, dword ptr [rax + 8]
0xca2f: mov     dword ptr [rax + 8], edi
0xca32: mov     eax, r8d
0xca35: ret
0xca36: nop     word ptr cs:[rax + rax]
0xca40: endbr64
0xca44: mov     rax, qword ptr [rip + 0x1e9bd]
0xca4b: cmp     qword ptr [rax + 0x48], 0
0xca50: sete    al
0xca53: movzx   eax, al
0xca56: ret
0xca57: nop     word ptr [rax + rax]
0xca60: endbr64
0xca64: mov     rax, qword ptr [rip + 0x1e99d]
0xca6b: mov     rdx, qword ptr [rax + 0x18]
0xca6f: mov     qword ptr [rax + 0x20], rdx
0xca73: ret
0xca74: nop     word ptr cs:[rax + rax]
0xca7f: nop
0xca80: endbr64
0xca84: sub     rsp, 8
0xca88: xor     eax, eax
0xca8a: mov     edi, 3
0xca8f: call    0xc6e0
0xca94: mov     rax, qword ptr [rip + 0x1e96d]
0xca9b: mov     dword ptr [rax + 0x10], 0x18
0xcaa2: add     rsp, 8
0xcaa6: ret

```

```

0xcaa7: nop      word ptr [rax + rax]
0xcab0: endbr64
0xcab4: sub      rsp, 8
0xcab8: xor      eax, eax
0xcaba: mov      edi, 1
0xcabf: call     0xc6e0
0xcac4: mov      rax, qword ptr [rip + 0x1e93d]
0xcacb: mov      dword ptr [rax + 0x10], 0x19
0xcad2: add      rsp, 8
0xcad6: ret
0xcad7: nop      word ptr [rax + rax]
0xcae0: endbr64
0xcae4: mov      rax, qword ptr [rip + 0x1e91d]
0xcaeb: mov      rdx, qword ptr [rax + 0x18]
0xcaef: mov      rcx, qword ptr [rax + 0x20]
0xcaf3: movsxd   rdx, edx
0xcaf6: mov      qword ptr [rax + 0x18], rcx
0xcafa: mov      qword ptr [rax + 0x20], rdx
0xcafe: ret
0xcaff: nop
0xcb00: endbr64
0xcb04: mov      rax, qword ptr [rip + 0x1e8fd]
0xcb0b: mov      rdx, qword ptr [rax + 0x18]
0xcb0f: cmp      rdx, qword ptr [rax + 0x40]
0xcb13: jbe      0xcb24
0xcb15: sub      rdx, 1
0xcb19: mov      dword ptr [rax + 0x10], 1
0xcb20: mov      qword ptr [rax + 0x18], rdx
0xcb24: ret
0xcb25: nop      word ptr cs:[rax + rax]
0xcb30: endbr64
0xcb34: mov      rax, qword ptr [rip + 0x1e8cd]
0xcb3b: mov      rdx, qword ptr [rax + 0x18]
0xcb3f: cmp      rdx, qword ptr [rax + 0x38]
0xcb43: jae      0xcb54
0xcb45: add      rdx, 1
0xcb49: mov      dword ptr [rax + 0x10], 2
0xcb50: mov      qword ptr [rax + 0x18], rdx
0xcb54: ret
0xcb55: nop      word ptr cs:[rax + rax]
0xcb60: endbr64
0xcb64: mov      rdx, qword ptr [rip + 0x1e89d]
0xcb6b: mov      rax, qword ptr [rdx + 0x18]
0xcb6f: mov      rdi, qword ptr [rdx + 0x40]
0xcb73: cmp      rax, rdi
0xcb76: jbe      0xcc20
0xcb7c: mov      r8, qword ptr [rdx + 0x58]
0xcb80: jmp      0xcba9

```

```

0xcba9
0xcb82: nop        word ptr [rax + rax]
0xcb88: lea          ecx, [rsi - 0x30]
0xcb8b: cmp          cl, 9
0xcb8e: jbe          0xcbbf
0xcb90: sub          esi, 0x24
0xcb93: cmp          sil, 1
0xcb97: jbe          0xcbbf
0xcb99: mov          qword ptr [rdx + 0x18], rax
0xcb9d: mov          dword ptr [rdx + 0x10], 1
0xcba4: cmp          rdi, rax
0xcba7: jae          0xcbef
0xcba9: sub          rax, 1
0xcbad: movzx        esi, byte ptr [r8 + rax]
0xcbb2: mov          ecx, esi
0xcbb4: and          ecx, 0xffffffff
0xcbb7: sub          ecx, 0x41
0xcbb8: cmp          cl, 0x19
0xcbbd: ja           0xcb88
0xcbbf: mov          rax, qword ptr [rdx + 0x18]
0xcbc3: cmp          rdi, rax
0xcbc6: jae          0xcbef
0xcbc8: sub          rax, 1
0xcbcc: movzx        esi, byte ptr [r8 + rax]
0xcbd1: mov          ecx, esi
0xcbd3: and          ecx, 0xffffffff
0xcbd6: sub          ecx, 0x41
0xcbd9: cmp          cl, 0x19
0xcbd9: jbe          0xcc00
0xcbde: lea          ecx, [rsi - 0x30]
0xcbe1: cmp          cl, 9
0xcbe4: jbe          0xcc00
0xcbe6: sub          esi, 0x24
0xcbe9: cmp          sil, 1
0xcbed: jbe          0xcc00
0xcbef: mov          dword ptr [rdx + 0x10], 3
0xcbf6: ret
0xcbf7: nop          word ptr [rax + rax]
0xcc00: mov          qword ptr [rdx + 0x18], rax
0xcc04: mov          dword ptr [rdx + 0x10], 1
0xcc0b: cmp          rdi, rax
0xcc0e: jb           0xcbc8
0xcc10: mov          dword ptr [rdx + 0x10], 3
0xcc17: ret
0xcc18: nop          dword ptr [rax + rax]
0xcc20: ret
0xcc21: nop          word ptr cs:[rax + rax]
0xcc2c: nop          dword ptr [rax]

```

```

0xcc30: endbr64
0xcc34: mov     rdx, qword ptr [rip + 0x1e7cd]
0xcc3b: mov     rax, qword ptr [rdx + 0x18]
0xcc3f: mov     rdi, qword ptr [rdx + 0x38]
0xcc43: cmp     rax, rdi
0xcc46: jae     0xcce8
0xcc4c: mov     r8, qword ptr [rdx + 0x58]
0xcc50: jmp     0xcc7d
0xcc7d:
0xcc52: nop     word ptr [rax + rax]
0xcc58: lea     ecx, [rsi - 0x30]
0xcc5b: cmp     cl, 9
0xcc5e: jbe     0xcc8f
0xcc60: sub     esi, 0x24
0xcc63: cmp     sil, 1
0xcc67: jbe     0xcc8f
0xcc69: add     rax, 1
0xcc6d: mov     dword ptr [rdx + 0x10], 2
0xcc74: mov     qword ptr [rdx + 0x18], rax
0xcc78: cmp     rdi, rax
0xcc7b: je      0xccbb
0xcc7d: movzx   esi, byte ptr [r8 + rax]
0xcc82: mov     ecx, esi
0xcc84: and     ecx, 0xffffffff
0xcc87: sub     ecx, 0x41
0xcc8a: cmp     cl, 0x19
0xcc8d: ja      0xcc58
0xcc8f: mov     rax, qword ptr [rdx + 0x18]
0xcc93: cmp     rdi, rax
0xcc96: jbe     0xccbb
0xcc98: movzx   esi, byte ptr [r8 + rax]
0xcc9d: mov     ecx, esi
0xcc9f: and     ecx, 0xffffffff
0xcca2: sub     ecx, 0x41
0xcca5: cmp     cl, 0x19
0xcca8: jbe     0xccc8
0xccaa: lea     ecx, [rsi - 0x30]
0ccad: cmp     cl, 9
0ccb0: jbe     0xccc8
0ccb2: sub     esi, 0x24
0ccb5: cmp     sil, 1
0ccb9: jbe     0xccc8
0ccbb: mov     dword ptr [rdx + 0x10], 4
0ccc2: ret
0ccc3: nop     dword ptr [rax + rax]
0ccc8: add     rax, 1
0cccc: mov     dword ptr [rdx + 0x10], 2
0ccd3: mov     qword ptr [rdx + 0x18], rax

```



```

0xccd7: cmp     rdi, rax
0xccda: jne     0xcc98
0xccdc: mov     dword ptr [rdx + 0x10], 4
0xcce3: ret
0xcce4: nop     dword ptr [rax]
0xcce8: ret
0xcce9: nop     dword ptr [rax]
0xccf0: endbr64
0xccf4: mov     rax, qword ptr [rip + 0x1e70d]
0xccfb: mov     rdx, qword ptr [rax + 0x40]
0xccff: mov     dword ptr [rax + 0x10], 5
0xcd06: mov     qword ptr [rax + 0x18], rdx
0xcd0a: ret
0xcd0b: nop     dword ptr [rax + rax]
0xcd10: endbr64
0xcd14: mov     rax, qword ptr [rip + 0x1e6ed]
0xcd1b: mov     rdx, qword ptr [rax + 0x38]
0xcd1f: mov     dword ptr [rax + 0x10], 6
0xcd26: mov     qword ptr [rax + 0x18], rdx
0xcd2a: ret
0xcd2b: nop     dword ptr [rax + rax]
0xcd30: endbr64
0xcd34: push    rbp
0xcd35: push    rbx
0xcd36: mov     ebx, edi
0xcd38: sub     rsp, 8
0xcd3c: call    0x4f00
0xcd41: movsxd  rdx, ebx
0xcd44: mov     rax, qword ptr [rax]
0xcd47: test    byte ptr [rax + rdx*2 + 1], 0x40
0xcd4c: je      0xcdab
0xcd4e: mov     rbp, qword ptr [rip + 0x1e6b3]
0xcd55: mov     rsi, qword ptr [rbp + 0x38]
0xcd59: lea     rdx, [rsi + 1]
0xcd5d: cmp     rdx, qword ptr [rbp + 0x50]
0xcd61: jae     0xcdb8
0xcd63: mov     rax, qword ptr [rbp + 0x18]
0xcd67: mov     rcx, qword ptr [rbp + 0x58]
0xcd6b: sub     rdx, rax
0xcd6e: lea     rsi, [rcx + rax]
0xcd72: lea     rdi, [rcx + rax + 1]
0xcd77: call    0x4d40
0xcd7c: mov     rax, qword ptr [rip + 0x1e685]
0xcd83: mov     rdx, qword ptr [rax + 0x58]
0xcd87: mov     rax, qword ptr [rax + 0x18]
0xcd8b: mov     byte ptr [rdx + rax], bl
0xcd8e: mov     rax, qword ptr [rip + 0x1e673]
0xcd95: add     qword ptr [rax + 0x18], 1

```

```

0xcd9a: add    qword ptr [rax + 0x38], 1
0xcd9f: add    qword ptr [rax + 0x48], 1
0xcda4: mov    dword ptr [rax + 0x10], 7
0xcdab: add    rsp, 8
0xcdaf: pop    rbx
0xcdb0: pop    rbp
0xcdb1: ret
0xcdb2: nop    word ptr [rax + rax]
0xcdb8: add    rsi, 0x21
0xcdbc: mov    rdi, qword ptr [rbp + 0x58]
0xcdc0: mov    qword ptr [rbp + 0x50], rsi
0xcdc4: call   0x1d690
0xcdc9: mov    qword ptr [rbp + 0x58], rax
0xcdcd: mov    rbp, qword ptr [rip + 0x1e634]
0xcdd4: mov    rax, qword ptr [rbp + 0x38]
0xcdd8: lea    rdx, [rax + 1]
0xcddc: jmp    0xcd63
0xcd63:
0xcdde: nop
0xcde0: endbr64
0xcde4: mov    rax, qword ptr [rip + 0x1e61d]
0xcdeb: mov    rcx, qword ptr [rax + 0x18]
0xcdef: mov    rdx, qword ptr [rax + 0x38]
0xcdf3: cmp    rcx, rdx
0xcdf6: jb     0xce00
0xcdf8: ret
0xcdf9: nop    dword ptr [rax]
0xce00: push   rbx
0xce01: mov    rax, qword ptr [rax + 0x58]
0xce05: add    rdx, 1
0xce09: sub    rdx, rcx
0xce0c: lea    rsi, [rax + rcx + 1]
0xce11: lea    rdi, [rax + rcx]
0xce15: call   0x4bb0
0xce1a: mov    rbx, qword ptr [rip + 0x1e5e7]
0xce21: mov    rsi, qword ptr [rbx + 0x38]
0xce25: sub    qword ptr [rbx + 0x48], 1
0xce2a: lea    rax, [rsi - 1]
0xce2e: mov    qword ptr [rbx + 0x38], rax
0xce32: test   al, 0xf
0xce34: je     0xce40
0xce36: mov    dword ptr [rbx + 0x10], 8
0xce3d: pop    rbx
0xce3e: ret
0xce3f: nop
0xce40: mov    qword ptr [rbx + 0x50], rsi
0xce44: mov    rdi, qword ptr [rbx + 0x58]
0xce48: call   0x1d690

```

```

0xce4d: mov     qword ptr [rbx + 0x58], rax
0xce51: mov     rbx, qword ptr [rip + 0x1e5b0]
0xce58: mov     dword ptr [rbx + 0x10], 8
0xce5f: pop     rbx
0xce60: ret
0xce61: nop     word ptr cs:[rax + rax]
0xce6c: nop     dword ptr [rax]
0xce70: endbr64
0xce74: mov     rax, qword ptr [rip + 0x1e58d]
0xce7b: mov     rcx, qword ptr [rax + 0x18]
0xce7f: cmp     rcx, qword ptr [rax + 0x40]
0xce83: ja      0xce90
0xce85: ret
0xce86: nop     word ptr cs:[rax + rax]
0xce90: push    rbx
0xce91: mov     rdx, qword ptr [rax + 0x38]
0xce95: mov     rsi, qword ptr [rax + 0x58]
0xce99: add     rdx, 1
0xce9d: lea     rdi, [rsi + rcx - 1]
0xcea2: sub     rdx, rcx
0xcea5: add     rsi, rcx
0xcea8: call    0x4bb0
0xcead: mov     rbx, qword ptr [rip + 0x1e554]
0xceb4: mov     rsi, qword ptr [rbx + 0x38]
0xceb8: sub     qword ptr [rbx + 0x18], 1
0xcebd: sub     qword ptr [rbx + 0x48], 1
0xcec2: lea     rax, [rsi - 1]
0xcec6: mov     qword ptr [rbx + 0x38], rax
0xceca: test    al, 0xf
0xcecc: je      0xcee0
0xcece: mov     dword ptr [rbx + 0x10], 9
0xcd5: pop     rbx
0xcd6: ret
0xcd7: nop     word ptr [rax + rax]
0xcee0: mov     qword ptr [rbx + 0x50], rsi
0xcee4: mov     rdi, qword ptr [rbx + 0x58]
0xcee8: call    0x1d690
0xceed: mov     qword ptr [rbx + 0x58], rax
0xcef1: mov     rbx, qword ptr [rip + 0x1e510]
0xcef8: mov     dword ptr [rbx + 0x10], 9
0xceff: pop     rbx
0xcf00: ret
0xcf01: nop     word ptr cs:[rax + rax]
0xcf0c: nop     dword ptr [rax]
0xcf10: endbr64
0xcf14: push    rbp
0xcf15: push    rbx
0xcf16: sub     rsp, 8

```

```

0xcf1a: mov     r9, qword ptr [rip + 0x1e4e7]
0xcf21: mov     rax, qword ptr [r9 + 0x18]
0xcf25: mov     rbx, qword ptr [r9 + 0x20]
0xcf29: mov     qword ptr [r9 + 0x20], rax
0xcf2d: xor     eax, eax
0xcf2f: call    0xcc30
0xcf34: xor     eax, eax
0xcf36: mov     edi, 3
0xcf3b: mov     rbp, qword ptr [r9 + 0x18]
0xcf3f: call    0xc6e0
0xcf44: mov     rax, qword ptr [rip + 0x1e4bd]
0xcf4b: mov     rdx, qword ptr [rax + 0x18]
0xcf4f: cmp     rdx, rbx
0xcf52: jae     0xcf65
0xcf54: lea     rcx, [rdx + rbx]
0xcf58: sub     rcx, rbp
0xcf5b: cmp     rbx, rbp
0xcf5e: cmovae rdx, rcx
0xcf62: mov     rbx, rdx
0xcf65: mov     qword ptr [rax + 0x20], rbx
0xcf69: mov     dword ptr [rax + 0x10], 0xa
0xcf70: add     rsp, 8
0xcf74: pop     rbx
0xcf75: pop     rbp
0xcf76: ret
0xcf77: nop     word ptr [rax + rax]
0xcf80: endbr64
0xcf84: push    rbp
0xcf85: push    rbx
0xcf86: sub     rsp, 8
0xcf8a: mov     rax, qword ptr [rip + 0x1e477]
0xcf91: mov     rbp, qword ptr [rax + 0x18]
0xcf95: mov     rbx, qword ptr [rax + 0x20]
0xcf99: mov     qword ptr [rax + 0x20], rbp
0xcf9d: xor     eax, eax
0xcf9f: call    0xcb60
0xcfa4: xor     eax, eax
0xcfa6: mov     edi, 3
0xcfab: call    0xc6e0
0xcfb0: mov     rax, qword ptr [rip + 0x1e451]
0xcfb7: mov     rdx, qword ptr [rax + 0x18]
0xcfb9: cmp     rdx, rbx
0xcfbe: jae     0xcfd1
0xcfc0: lea     rcx, [rdx + rbx]
0xcfc4: sub     rcx, rbp
0xcfc7: cmp     rbx, rbp
0xcfc9: cmovae rdx, rcx
0xcfce: mov     rbx, rdx

```

```

0xcfd1: mov     qword ptr [rax + 0x20], rbx
0xcfd5: mov     dword ptr [rax + 0x10], 0xb
0xcfdc: add     rsp, 8
0xcfe0: pop     rbx
0xcfe1: pop     rbp
0xcfe2: ret
0xcfe3: nop     word ptr cs:[rax + rax]
0xcfee: nop
0xcff0: endbr64
0xcff4: push    rbx
0xcff5: mov     rbx, qword ptr [rip + 0x1e40c]
0xcffc: mov     esi, 1
0xd001: mov     qword ptr [rbx + 0x18], 0
0xd009: mov     rdi, qword ptr [rbx + 0x58]
0xd00d: mov     qword ptr [rbx + 0x20], 0
0xd015: mov     qword ptr [rbx + 0x38], 0
0xd01d: mov     qword ptr [rbx + 0x40], 0
0xd025: mov     qword ptr [rbx + 0x48], 0
0xd02d: mov     qword ptr [rbx + 0x50], 1
0xd035: call    0x1d690
0xd03a: mov     qword ptr [rbx + 0x58], rax
0xd03e: mov     rax, qword ptr [rip + 0x1e3c3]
0xd045: mov     rax, qword ptr [rax + 0x58]
0xd049: mov     byte ptr [rax], 0
0xd04c: mov     rax, qword ptr [rip + 0x1e3b5]
0xd053: mov     dword ptr [rax + 0x10], 0xc
0xd05a: pop     rbx
0xd05b: ret
0xd05c: nop     dword ptr [rax]
0xd060: endbr64
0xd064: push    rbx
0xd065: mov     edi, 0x70
0xd06a: sub     rsp, 0x10
0xd06e: mov     rax, qword ptr fs:[0x28]
0xd077: mov     qword ptr [rsp + 8], rax
0xd07c: xor     eax, eax
0xd07e: call    0x1d670
0xd083: mov     qword ptr [rax + 8], 1
0xd08b: mov     qword ptr [rax + 0x58], 0
0xd093: mov     qword ptr [rax + 0x60], 0
0xd09b: mov     qword ptr [rax + 0x28], 0
0xd0a3: mov     qword ptr [rax + 0x30], 0
0xd0ab: mov     qword ptr [rip + 0x1e356], rax
0xd0b2: xor     eax, eax
0xd0b4: call    0xcff0
0xd0b9: mov     rbx, qword ptr [rip + 0x1e348]
0xd0c0: xor     eax, eax
0xd0c2: call    0xc590

```

```

0xd0c7: lea    rdi, [rip + 0x135e4]
0xd0ce: mov     qword ptr [rbx], rax
0xd0d1: call    0x17c50
0xd0d6: mov     ecx, 1
0xd0db: mov     rsi, rsp
0xd0de: mov     edx, 1
0xd0e3: lea     rdi, [rip + 0x13e42]
0xd0ea: call    0x17320
0xd0ef: mov     rdi, qword ptr [rsp]
0xd0f3: lea     rax, [rip + 0x13e24]
0xd0fa: mov     rbx, qword ptr [rip + 0x1e307]
0xd101: test    rdi, rdi
0xd104: cmov    rdi, rax
0xd108: xor     eax, eax
0xd10a: call    0x19940
0xd10f: lea     rdi, [rip + 0x1cb32]
0xd116: mov     qword ptr [rbx + 0x68], rax
0xd11a: mov     eax, dword ptr [rip + 0x1cb40]
0xd120: test    eax, eax
0xd122: lea     rax, [rip + 0x1cb07]
0xd129: cmov    rdi, rax
0xd12d: call    0x17c50
0xd132: lea     rdi, [rip + 0x1d0c7]
0xd139: mov     edx, 6
0xd13e: lea     rsi, [rip + 0x1d0db]
0xd145: call    0x17f70
0xd14a: xor     eax, eax
0xd14c: call    0x1a310
0xd151: mov     rax, qword ptr [rip + 0x1e2b0]
0xd158: mov     rdi, qword ptr [rax + 0x68]
0xd15c: xor     eax, eax
0xd15e: call    0x1a900
0xd163: nop     dword ptr [rax + rax]
0xd168: xor     eax, eax
0xd16a: call    0x1ab40
0xd16f: test    rax, rax
0xd172: jne     0xd168
0xd174: mov     edi, 0x200
0xd179: call    0x1a620
0xd17e: mov     rax, qword ptr [rsp + 8]
0xd183: sub     rax, qword ptr fs:[0x28]
0xd18c: jne     0xd194
0xd18e: add     rsp, 0x10
0xd192: pop     rbx
0xd193: ret
0xd194: call    0x49d0
0xd199: nop     dword ptr [rax]
0xd1a0: endbr64

```

```

0xd1a4: sub     rsp, 8
0xd1a8: mov     rax, qword ptr [rip + 0x1e259]
0xd1af: or      edi, 2
0xd1b2: mov     rdx, qword ptr [rax + 0x40]
0xd1b6: mov     dword ptr [rax + 0x10], 6
0xd1bd: mov     qword ptr [rax + 0x20], rdx
0xd1c1: mov     rdx, qword ptr [rax + 0x38]
0xd1c5: mov     qword ptr [rax + 0x18], rdx
0xd1c9: xor     eax, eax
0xd1cb: call    0xc6e0
0xd1d0: mov     rax, qword ptr [rip + 0x1e231]
0xd1d7: mov     rdx, qword ptr [rax + 0x18]
0xd1db: mov     dword ptr [rax + 0x10], 0xd
0xd1e2: mov     qword ptr [rax + 0x20], rdx
0xd1e6: add     rsp, 8
0xd1ea: ret
0xd1eb: nop     dword ptr [rax + rax]
0xd1f0: endbr64
0xd1f4: mov     rax, qword ptr [rip + 0x1e20d]
0xd1fb: push    rbx
0xd1fc: mov     edi, 3
0xd201: mov     rdx, qword ptr [rax + 0x18]
0xd205: mov     rsi, qword ptr [rax + 0x20]
0xd209: mov     dword ptr [rax + 0x10], 5
0xd210: mov     rcx, qword ptr [rax + 0x40]
0xd214: mov     qword ptr [rax + 0x20], rdx
0xd218: mov     rbx, rsi
0xd21b: mov     qword ptr [rax + 0x18], rcx
0xd21f: sub     rbx, rdx
0xd222: cmp     rsi, rdx
0xd225: cmovbe  rbx, rcx
0xd229: xor     eax, eax
0xd22b: call    0xc6e0
0xd230: mov     rdx, qword ptr [rip + 0x1e1d1]
0xd237: mov     rax, qword ptr [rdx + 0x38]
0xd23b: mov     dword ptr [rdx + 0x10], 0xe
0xd242: cmp     rax, rbx
0xd245: cmova   rax, rbx
0xd249: mov     qword ptr [rdx + 0x20], rax
0xd24d: pop     rbx
0xd24e: ret
0xd24f: nop
0xd250: endbr64
0xd254: mov     rax, qword ptr [rip + 0x1e1ad]
0xd25b: push    rbx
0xd25c: mov     edi, 3
0xd261: mov     rdx, qword ptr [rax + 0x18]
0xd265: mov     rbx, qword ptr [rax + 0x20]

```

```

0xd269: mov     dword ptr [rax + 0x10], 6
0xd270: mov     qword ptr [rax + 0x20], rdx
0xd274: mov     rdx, qword ptr [rax + 0x38]
0xd278: mov     qword ptr [rax + 0x18], rdx
0xd27c: xor     eax, eax
0xd27e: call    0xc6e0
0xd283: mov     rdx, qword ptr [rip + 0x1e17e]
0xd28a: mov     rax, qword ptr [rdx + 0x38]
0xd28e: mov     dword ptr [rdx + 0x10], 0xf
0xd295: cmp     rax, rbx
0xd298: cmova   rax, rbx
0xd29c: mov     qword ptr [rdx + 0x20], rax
0xd2a0: pop     rbx
0xd2a1: ret
0xd2a2: nop     word ptr cs:[rax + rax]
0xd2ad: nop     dword ptr [rax]
0xd2b0: endbr64
0xd2b4: mov     rax, qword ptr [rip + 0x1e14d]
0xd2bb: mov     rdx, qword ptr [rax + 0x58]
0xd2bf: mov     rax, qword ptr [rax + 0x18]
0xd2c3: cmp     byte ptr [rdx + rax], 0x20
0xd2c7: je      0xd2d0
0xd2c9: ret
0xd2ca: nop     word ptr [rax + rax]
0xd2d0: sub     rsp, 8
0xd2d4: nop     dword ptr [rax]
0xd2d8: xor     eax, eax
0xd2da: call    0xcde0
0xd2df: mov     rax, qword ptr [rip + 0x1e122]
0xd2e6: mov     rdx, qword ptr [rax + 0x58]
0xd2ea: mov     rcx, qword ptr [rax + 0x18]
0xd2ee: cmp     byte ptr [rdx + rcx], 0x20
0xd2f2: je      0xd2d8
0xd2f4: cmp     qword ptr [rax + 0x48], 0
0xd2f9: jne     0xd31d
0xd2fb: jmp     0xd324
0xd324:
0xd2fd: nop     dword ptr [rax]
0xd300: xor     eax, eax
0xd302: call    0xce70
0xd307: mov     rax, qword ptr [rip + 0x1e0fa]
0xd30e: cmp     qword ptr [rax + 0x48], 0
0xd313: je      0xd324
0xd315: mov     rdx, qword ptr [rax + 0x58]
0xd319: mov     rcx, qword ptr [rax + 0x18]
0xd31d: cmp     byte ptr [rdx + rcx - 1], 0x20
0xd322: je      0xd300
0xd324: mov     dword ptr [rax + 0x10], 0x11

```



```

0xd32b: add     rsp, 8
0xd32f: ret
0xd330: endbr64
0xd334: mov     rax, qword ptr [rip + 0x1e0cd]
0xd33b: mov     rdx, qword ptr [rax + 0x58]
0xd33f: mov     rax, qword ptr [rax + 0x18]
0xd343: cmp     byte ptr [rdx + rax], 0x20
0xd347: je      0xd350
0xd349: ret
0xd34a: nop     word ptr [rax + rax]
0xd350: sub     rsp, 8
0xd354: xor     eax, eax
0xd356: call    0xd2b0
0xd35b: mov     edi, 0x20
0xd360: call    0xcd30
0xd365: mov     rax, qword ptr [rip + 0x1e09c]
0xd36c: mov     dword ptr [rax + 0x10], 0x10
0xd373: add     rsp, 8
0xd377: ret
0xd378: nop     dword ptr [rax + rax]
0xd380: endbr64
0xd384: push    rbp
0xd385: push    rbx
0xd386: sub     rsp, 8
0xd38a: mov     rbp, qword ptr [rip + 0x1e077]
0xd391: mov     rbx, qword ptr [rbp + 0x18]
0xd395: cmp     rbx, qword ptr [rbp + 0x38]
0xd399: jb      0xd3a8
0xd39b: add     rsp, 8
0xd39f: pop     rbx
0xd3a0: pop     rbp
0xd3a1: ret
0xd3a2: nop     word ptr [rax + rax]
0xd3a8: xor     eax, eax
0xd3aa: call    0xcc30
0xd3af: cmp     rbx, qword ptr [rbp + 0x18]
0xd3b3: jae     0xd3e4
0xd3b5: call    0x4ef0
0xd3ba: nop     word ptr [rax + rax]
0xd3c0: mov     rdx, qword ptr [rbp + 0x58]
0xd3c4: mov     rcx, qword ptr [rax]
0xd3c7: add     rdx, rbx
0xd3ca: add     rbx, 1
0xd3ce: movsx   rsi, byte ptr [rdx]
0xd3d2: mov     ecx, dword ptr [rcx + rsi*4]
0xd3d5: mov     byte ptr [rdx], cl
0xd3d7: mov     rbp, qword ptr [rip + 0x1e02a]
0xd3de: cmp     qword ptr [rbp + 0x18], rbx

```

```

0xd3e2: ja      0xd3c0
0xd3e4: mov     dword ptr [rbp + 0x10], 0x12
0xd3eb: add     rsp, 8
0xd3ef: pop     rbx
0xd3f0: pop     rbp
0xd3f1: ret
0xd3f2: nop     word ptr cs:[rax + rax]
0xd3fd: nop     dword ptr [rax]
0xd400: endbr64
0xd404: push    rbp
0xd405: push    rbx
0xd406: sub     rsp, 8
0xd40a: mov     rbp, qword ptr [rip + 0x1dff7]
0xd411: mov     rbx, qword ptr [rbp + 0x18]
0xd415: cmp     rbx, qword ptr [rbp + 0x38]
0xd419: jb      0xd428
0xd41b: add     rsp, 8
0xd41f: pop     rbx
0xd420: pop     rbp
0xd421: ret
0xd422: nop     word ptr [rax + rax]
0xd428: xor     eax, eax
0xd42a: call    0xcc30
0xd42f: cmp     rbx, qword ptr [rbp + 0x18]
0xd433: jae     0xd464
0xd435: call    0x47c0
0xd43a: nop     word ptr [rax + rax]
0xd440: mov     rdx, qword ptr [rbp + 0x58]
0xd444: mov     rcx, qword ptr [rax]
0xd447: add     rdx, rbx
0xd44a: add     rbx, 1
0xd44e: movsx   rsi, byte ptr [rdx]
0xd452: mov     ecx, dword ptr [rcx + rsi*4]
0xd455: mov     byte ptr [rdx], cl
0xd457: mov     rbp, qword ptr [rip + 0x1dfaa]
0xd45e: cmp     qword ptr [rbp + 0x18], rbx
0xd462: ja      0xd440
0xd464: mov     dword ptr [rbp + 0x10], 0x13
0xd46b: add     rsp, 8
0xd46f: pop     rbx
0xd470: pop     rbp
0xd471: ret
0xd472: nop     word ptr cs:[rax + rax]
0xd47d: nop     dword ptr [rax]
0xd480: endbr64
0xd484: push    r14
0xd486: push    r13
0xd488: push    r12

```

```

0xd48a: push    rbp
0xd48b: mov     rbp, qword ptr [rip + 0x1df76]
0xd492: push    rbx
0xd493: mov     rbx, qword ptr [rbp + 0x18]
0xd497: cmp     rbx, qword ptr [rbp + 0x38]
0xd49b: jb      0xd4b0
0xd49d: pop     rbx
0xd49e: pop     rbp
0xd49f: pop     r12
0xd4a1: pop     r13
0xd4a3: pop     r14
0xd4a5: ret
0xd4a6: nop     word ptr cs:[rax + rax]
0xd4b0: xor     eax, eax
0xd4b2: call    0xcc30
0xd4b7: cmp     rbx, qword ptr [rbp + 0x18]
0xd4bb: jae     0xd528
0xd4bd: call    0x4f00
0xd4c2: mov     r14d, 1
0xd4c8: mov     r13, rax
0xd4cb: jmp     0xd4f3
0xd4f3:
0xd4cd: nop     dword ptr [rax]
0xd4d0: call    0x47c0
0xd4d5: xor     r14d, r14d
0xd4d8: mov     rax, qword ptr [rax]
0xd4db: mov     eax, dword ptr [rax + r12*4]
0xd4df: mov     byte ptr [rbp], al
0xd4e2: mov     rbp, qword ptr [rip + 0x1df1f]
0xd4e9: add     rbx, 1
0xd4ed: cmp     qword ptr [rbp + 0x18], rbx
0xd4f1: jbe     0xd528
0xd4f3: mov     rax, qword ptr [rbp + 0x58]
0xd4f7: add     rax, rbx
0xd4fa: movsx   r12, byte ptr [rax]
0xd4fe: mov     rbp, rax
0xd501: mov     rax, qword ptr [r13]
0xd505: test    byte ptr [rax + r12*2], 8
0xd50a: je      0xd4e2
0xd50c: test    r14d, r14d
0xd50f: jne     0xd4d0
0xd511: call    0x4ef0
0xd516: mov     rax, qword ptr [rax]
0xd519: mov     eax, dword ptr [rax + r12*4]
0xd51d: mov     byte ptr [rbp], al
0xd520: jmp     0xd4e2
0xd4e2:
0xd522: nop     word ptr [rax + rax]

```

```

0xd528: mov     dword ptr [rbp + 0x10], 0x14
0xd52f: pop     rbx
0xd530: pop     rbp
0xd531: pop     r12
0xd533: pop     r13
0xd535: pop     r14
0xd537: ret
0xd538: nop     dword ptr [rax + rax]
0xd540: endbr64
0xd544: push    r12
0xd546: mov     r12, rdi
0xd549: push    rbp
0xd54a: push    rbx
0xd54b: call    0x49b0
0xd550: mov     rbp, qword ptr [rip + 0x1deb1]
0xd557: mov     rbx, rax
0xd55a: mov     rsi, qword ptr [rbp + 0x40]
0xd55e: sub     rax, rsi
0xd561: add     qword ptr [rbp + 0x18], rax
0xd565: add     qword ptr [rbp + 0x20], rax
0xd569: mov     rax, qword ptr [rbp + 0x48]
0xd56d: lea     rdx, [rax + 1]
0xd571: lea     r8, [rbx + rdx]
0xd575: cmp     r8, qword ptr [rbp + 0x50]
0xd579: ja      0xd5d8
0xd57b: mov     rdi, qword ptr [rbp + 0x58]
0xd57f: add     rsi, rdi
0xd582: add     rdi, rbx
0xd585: call    0x4d40
0xd58a: mov     rax, qword ptr [rip + 0x1de77]
0xd591: mov     rdx, rbx
0xd594: mov     rsi, r12
0xd597: mov     rdi, qword ptr [rax + 0x58]
0xd59b: call    0x4bb0
0xd5a0: mov     rax, qword ptr [rip + 0x1de61]
0xd5a7: mov     rsi, rbx
0xd5aa: mov     rdi, qword ptr [rax + 0x58]
0xd5ae: call    0x184d0
0xd5b3: mov     rax, qword ptr [rip + 0x1de4e]
0xd5ba: mov     qword ptr [rax + 0x40], rbx
0xd5be: add     rbx, qword ptr [rax + 0x48]
0xd5c2: mov     qword ptr [rax + 0x38], rbx
0xd5c6: mov     dword ptr [rax + 0x10], 0x15
0xd5cd: pop     rbx
0xd5ce: pop     rbp
0xd5cf: pop     r12
0xd5d1: ret
0xd5d2: nop     word ptr [rax + rax]

```

```

0xd5d8: mov     qword ptr [rbp + 0x50], r8
0xd5dc: mov     rdi, qword ptr [rbp + 0x58]
0xd5e0: mov     rsi, r8
0xd5e3: call    0x1d690
0xd5e8: mov     qword ptr [rbp + 0x58], rax
0xd5ec: mov     rbp, qword ptr [rip + 0x1de15]
0xd5f3: mov     rax, qword ptr [rbp + 0x48]
0xd5f7: mov     rsi, qword ptr [rbp + 0x40]
0xd5fb: lea     rdx, [rax + 1]
0xd5ff: jmp     0xd57b
0xd57b:
0xd604: nop     word ptr cs:[rax + rax]
0xd60f: nop
0xd610: endbr64
0xd614: test    rdi, rdi
0xd617: je      0xd6f0
0xd61d: push    r12
0xd61f: push    rbp
0xd620: mov     rbp, rdi
0xd623: push    rbx
0xd624: call    0x49b0
0xd629: mov     r12, qword ptr [rip + 0x1ddd8]
0xd630: mov     rbx, rax
0xd633: mov     rax, qword ptr [r12 + 0x38]
0xd638: lea     rdx, [rax + 1]
0xd63c: mov     rax, qword ptr [r12 + 0x50]
0xd641: lea     rcx, [rbx + rdx]
0xd645: cmp     rcx, rax
0xd648: ja      0xd6c0
0xd64a: mov     rax, qword ptr [r12 + 0x18]
0xd64f: mov     rdi, qword ptr [r12 + 0x58]
0xd654: sub     rdx, rax
0xd657: lea     rsi, [rdi + rax]
0xd65b: add     rax, rbx
0xd65e: add     rdi, rax
0xd661: call    0x4d40
0xd666: mov     rax, qword ptr [rip + 0x1dd9b]
0xd66d: mov     rdx, rbx
0xd670: mov     rsi, rbp
0xd673: mov     rdi, qword ptr [rax + 0x18]
0xd677: add     rdi, qword ptr [rax + 0x58]
0xd67b: call    0x4bb0
0xd680: mov     rax, qword ptr [rip + 0x1dd81]
0xd687: mov     rsi, rbx
0xd68a: mov     rdi, qword ptr [rax + 0x18]
0xd68e: add     rdi, qword ptr [rax + 0x58]
0xd692: call    0x184d0
0xd697: mov     rax, qword ptr [rip + 0x1dd6a]

```

```

0xd69e: add    qword ptr [rax + 0x18], rbx
0xd6a2: add    qword ptr [rax + 0x38], rbx
0xd6a6: add    qword ptr [rax + 0x48], rbx
0xd6aa: mov    dword ptr [rax + 0x10], 0x16
0xd6b1: pop    rbx
0xd6b2: pop    rbp
0xd6b3: pop    r12
0xd6b5: ret
0xd6b6: nop    word ptr cs:[rax + rax]
0xd6c0: lea    rsi, [rax + rbx + 0x21]
0xd6c5: mov    rdi, qword ptr [r12 + 0x58]
0xd6ca: mov    qword ptr [r12 + 0x50], rsi
0xd6cf: call   0xd690
0xd6d4: mov    qword ptr [r12 + 0x58], rax
0xd6d9: mov    r12, qword ptr [rip + 0x1dd28]
0xd6e0: mov    rax, qword ptr [r12 + 0x38]
0xd6e5: lea    rdx, [rax + 1]
0xd6e9: jmp    0xd64a
0xd64a:
0xd6ee: nop
0xd6f0: ret
0xd6f1: nop    word ptr cs:[rax + rax]
0xd6fc: nop    dword ptr [rax]
0xd700: endbr64
0xd704: mov    rax, qword ptr [rip + 0x1dcfd]
0xd70b: mov    rdi, qword ptr [rax + 0x60]
0xd70f: test   rdi, rdi
0xd712: je     0xd730
0xd714: sub    rsp, 8
0xd718: call   0xd610
0xd71d: mov    rax, qword ptr [rip + 0x1dce4]
0xd724: mov    dword ptr [rax + 0x10], 0x1a
0xd72b: add    rsp, 8
0xd72f: ret
0xd730: ret
0xd731: nop    word ptr cs:[rax + rax]
0xd73c: nop    dword ptr [rax]
0xd740: endbr64
0xd744: mov    rax, qword ptr [rip + 0x1dcbd]
0xd74b: push   r12
0xd74d: push   rbp
0xd74e: push   rbx
0xd74f: mov    rbx, qword ptr [rax + 0x40]
0xd753: mov    r12, qword ptr [rax + 0x28]
0xd757: mov    eax, dword ptr [rax + 0xc]
0xd75a: lea    rbp, [rbx + 3]
0xd75e: test   eax, eax
0xd760: je     0xd7f0

```

```

0xd766: mov     edx, dword ptr [rip + 0x1caa4]
0xd76c: mov     esi, dword ptr [rip + 0x1ca9a]
0xd772: mov     edi, dword ptr [rip + 0x1ca9c]
0xd778: call    0xa570
0xd77d: mov     rcx, qword ptr [rip + 0x1dc84]
0xd784: cmp     r12, rbp
0xd787: jb      0xd810
0xd78d: mov     rsi, qword ptr [rcx + 0x18]
0xd791: mov     rdx, qword ptr [rcx + 0x40]
0xd795: xor     ebx, ebx
0xd797: mov     rdi, qword ptr [rcx + 0x28]
0xd79b: mov     rax, rdi
0xd79e: sub     rax, rdx
0xd7a1: mov     edx, 1
0xd7a6: mov     r8, rax
0xd7a9: shr     r8, 2
0xd7ad: cmp     rax, 3
0xd7b1: cmova   edx, r8d
0xd7b5: xor     eax, eax
0xd7b7: cmp     rdi, rsi
0xd7ba: ja      0xd7d3
0xd7bc: mov     rax, rsi
0xd7bf: sub     rax, rdi
0xd7c2: movsxd  rdi, edx
0xd7c5: lea     r8d, [rdx + rax]
0xd7c9: xor     edx, edx
0xd7cb: div     rdi
0xd7ce: mov     eax, r8d
0xd7d1: sub     eax, edx
0xd7d3: mov     edx, esi
0xd7d5: mov     rdi, qword ptr [rcx]
0xd7d8: xor     esi, esi
0xd7da: sub     edx, eax
0xd7dc: call    0xc680
0xd7e1: cmp     r12, rbp
0xd7e4: jb      0xd840
0xd7e6: pop     rbx
0xd7e7: pop     rbp
0xd7e8: pop     r12
0xd7ea: ret
0xd7eb: nop     dword ptr [rax + rax]
0xd7f0: mov     edx, dword ptr [rip + 0x1ca0e]
0xd7f6: mov     esi, dword ptr [rip + 0x1ca04]
0xd7fc: mov     edi, dword ptr [rip + 0x1ca06]
0xd802: jmp     0xd778
0xd778
0xd807: nop     word ptr [rax + rax]
0xd810: mov     rbx, qword ptr [rcx + 0x40]

```

```

0xd814: mov     rsi, qword ptr [rcx + 0x18]
0xd818: mov     qword ptr [rcx + 0x40], 0
0xd820: xor     edx, edx
0xd822: add     qword ptr [rcx + 0x58], rbx
0xd826: sub     rsi, rbx
0xd829: sub     qword ptr [rcx + 0x38], rbx
0xd82d: sub     qword ptr [rcx + 0x20], rbx
0xd831: movsxd  rbx, ebx
0xd834: mov     qword ptr [rcx + 0x18], rsi
0xd838: jmp     0xd797
0xd797:
0xd83d: nop     dword ptr [rax]
0xd840: mov     rax, qword ptr [rip + 0x1dbc1]
0xd847: movsxd  rbx, ebx
0xd84a: mov     qword ptr [rax + 0x40], rbx
0xd84e: sub     qword ptr [rax + 0x58], rbx
0xd852: add     qword ptr [rax + 0x38], rbx
0xd856: add     qword ptr [rax + 0x18], rbx
0xd85a: add     qword ptr [rax + 0x20], rbx
0xd85e: pop     rbx
0xd85f: pop     rbp
0xd860: pop     r12
0xd862: ret
0xd863: nop     word ptr cs:[rax + rax]
0xd86e: nop
0xd870: endbr64
0xd874: push    r15
0xd876: push    r14
0xd878: push    r13
0xd87a: push    r12
0xd87c: push    rbp
0xd87d: push    rbx
0xd87e: sub     rsp, 0x28
0xd882: mov     rax, qword ptr fs:[0x28]
0xd88b: mov     qword ptr [rsp + 0x18], rax
0xd890: mov     rax, qword ptr [rip + 0x1db71]
0xd897: lea     r14, [rsp + 0x17]
0xd89c: mov     rdi, r14
0xd89f: mov     rbx, qword ptr [rax + 0x40]
0xd8a3: mov     r13, qword ptr [rax + 0x28]
0xd8a7: call    0xa6a0
0xd8ac: mov     rax, qword ptr [rip + 0x1db55]
0xd8b3: add     rbx, 3
0xd8b7: mov     ecx, dword ptr [rax + 0xc]
0xd8ba: test    ecx, ecx
0xd8bc: je      0xda20
0xd8c2: mov     edx, dword ptr [rip + 0x1c948]
0xd8c8: mov     esi, dword ptr [rip + 0x1c93e]

```



```

0xd8ce: mov     edi, dword ptr [rip + 0x1c940]
0xd8d4: call    0xa570
0xd8d9: mov     rdx, qword ptr [rip + 0x1db28]
0xd8e0: cmp     r13, rbx
0xd8e3: jb      0xda90
0xd8e9: mov     qword ptr [rsp + 8], 0
0xd8f2: mov     rax, qword ptr [rdx + 0x18]
0xd8f6: mov     rcx, qword ptr [rdx + 0x40]
0xd8fa: mov     rdi, qword ptr [rdx + 0x28]
0xd8fe: mov     r15, rdi
0xd901: sub     r15, rcx
0xd904: cmp     r15, 3
0xd908: jbe     0xda80
0xd90e: shr     r15, 2
0xd912: movsxd  rcx, r15d
0xd915: cmp     rdi, rax
0xd918: ja      0xda70
0xd91e: mov     edx, eax
0xd920: sub     rax, rdi
0xd923: sub     edx, edi
0xd925: add     r15d, edx
0xd928: xor     edx, edx
0xd92a: div     rcx
0xd92d: sub     r15d, edx
0xd930: mov     r12d, r15d
0xd933: call    0x1d670
0xd938: mov     esi, 0x20
0xd93d: mov     rbp, rax
0xd940: mov     rax, qword ptr [rip + 0x1dac1]
0xd947: mov     rdi, rbp
0xd94a: mov     rdx, qword ptr [rax + 0x28]
0xd94e: call    0x4a50
0xd953: mov     rax, qword ptr [rip + 0x1daae]
0xd95a: mov     edx, dword ptr [rax + 8]
0xd95d: test    edx, edx
0xd95f: je      0xda40
0xd965: mov     rdx, qword ptr [rax + 0x38]
0xd969: sub     rdx, qword ptr [rax + 0x40]
0xd96d: mov     rdi, rbp
0xd970: sub     rdx, r12
0xd973: mov     rsi, qword ptr [rax + 0x28]
0xd977: sub     rsi, qword ptr [rax + 0x40]
0xd97b: cmp     rdx, rsi
0xd97e: cmova   rdx, rsi
0xd982: mov     rsi, qword ptr [rax + 0x40]
0xd986: add     rsi, r12
0xd989: add     rsi, qword ptr [rax + 0x58]
0xd98d: call    0x4bb0

```

```

0xd992: mov     rax, qword ptr [rip + 0x1da6f]
0xd999: xor     edx, edx
0xd99b: xor     esi, esi
0xd99d: mov     rdi, qword ptr [rax]
0xd9a0: call    0xc680
0xd9a5: cmp     r13, rbx
0xd9a8: jae     0xdaf0
0xd9ae: mov     rax, qword ptr [rip + 0x1da53]
0xd9b5: mov     rsi, rbp
0xd9b8: mov     rdi, qword ptr [rax]
0xd9bb: mov     rdx, qword ptr [rax + 0x28]
0xd9bf: sub     edx, dword ptr [rax + 0x40]
0xd9c2: call    0xc600
0xd9c7: mov     rax, qword ptr [rip + 0x1da3a]
0xd9ce: xor     esi, esi
0xd9d0: mov     edx, dword ptr [rax + 0x18]
0xd9d3: mov     rdi, qword ptr [rax]
0xd9d6: sub     edx, r15d
0xd9d9: call    0xc680
0xd9de: cmp     r13, rbx
0xd9e1: jnb     0xdac8
0xd9e7: mov     rdi, rbp
0xd9ea: call    0x18d10
0xd9ef: mov     rdi, r14
0xd9f2: call    0xa6b0
0xd9f7: mov     rax, qword ptr [rsp + 0x18]
0xd9fc: sub     rax, qword ptr fs:[0x28]
0xda05: jne     0xdb0b
0xda0b: add     rsp, 0x28
0xda0f: pop     rbx
0xda10: pop     rbp
0xda11: pop     r12
0xda13: pop     r13
0xda15: pop     r14
0xda17: pop     r15
0xda19: ret
0xda1a: nop     word ptr [rax + rax]
0xda20: mov     edx, dword ptr [rip + 0x1c7de]
0xda26: mov     esi, dword ptr [rip + 0x1c7d4]
0xda2c: mov     edi, dword ptr [rip + 0x1c7d6]
0xda32: jmp     0xd8d4
0xd8d4:
0xda37: nop     word ptr [rax + rax]
0xda40: mov     rsi, qword ptr [rax + 0x40]
0xda44: mov     rdx, qword ptr [rax + 0x38]
0xda48: mov     rdi, rbp
0xda4b: mov     rax, qword ptr [rax + 0x28]
0xda4f: sub     rdx, rsi

```

```

0xda52: sub     rdx, r12
0xda55: sub     rax, rsi
0xda58: mov     esi, 0x2a
0xda5d: cmp     rdx, rax
0xda60: cmova   rdx, rax
0xda64: call    0x4a50
0xda69: jmp     0xd992
0xd992:
0xda6e: nop
0xda70: xor     r12d, r12d
0xda73: xor     r15d, r15d
0xda76: jmp     0xd933
0xd933:
0xda7b: nop     dword ptr [rax + rax]
0xda80: mov     ecx, 1
0xda85: mov     r15d, 1
0xda8b: jmp     0xd915
0xd915:
0xda90: mov     r12, qword ptr [rdx + 0x40]
0xda94: mov     rax, qword ptr [rdx + 0x18]
0xda98: mov     qword ptr [rdx + 0x40], 0
0xdaa0: add     qword ptr [rdx + 0x58], r12
0xdaa4: movsxd  rcx, r12d
0xdaa7: sub     rax, r12
0xdaaa: sub     qword ptr [rdx + 0x38], r12
0xdaae: mov     qword ptr [rsp + 8], rcx
0xdab3: xor     ecx, ecx
0xdab5: sub     qword ptr [rdx + 0x20], r12
0xdab9: mov     qword ptr [rdx + 0x18], rax
0xdabd: jmp     0xd8fa
0xd8fa:
0xdac2: nop     word ptr [rax + rax]
0xdac8: mov     rax, qword ptr [rip + 0x1d939]
0xdacf: movsxd  r12, dword ptr [rsp + 8]
0xdad4: sub     qword ptr [rax + 0x58], r12
0xdad8: add     qword ptr [rax + 0x38], r12
0xdadc: add     qword ptr [rax + 0x18], r12
0xdae0: add     qword ptr [rax + 0x20], r12
0xdae4: mov     qword ptr [rax + 0x40], r12
0xdae8: jmp     0xd9e7
0xd9e7:
0xdaed: nop     dword ptr [rax]
0xdaf0: mov     rax, qword ptr [rip + 0x1d911]
0xdaf7: mov     rsi, qword ptr [rax + 0x58]
0xdafb: mov     edx, dword ptr [rax + 0x40]
0xdafe: mov     rdi, qword ptr [rax]
0xdb01: call    0xc600
0xdb06: jmp     0xd9ae

```

```

0xd9ae
0xdb0b: call    0x49d0
0xdb10: endbr64
0xdb14: mov     rax, qword ptr [rip + 0x1d8ed]
0xdb1b: push    rbx
0xdb1c: mov     rbx, rdi
0xdb1f: mov     rdi, qword ptr [rdi]
0xdb22: mov     rsi, qword ptr [rax + 0x48]
0xdb26: add     rsi, 1
0xdb2a: call    0x1d690
0xdb2f: mov     qword ptr [rbx], rax
0xdb32: mov     rdi, rax
0xdb35: mov     rax, qword ptr [rip + 0x1d8cc]
0xdb3c: mov     rcx, qword ptr [rax + 0x48]
0xdb40: mov     rsi, qword ptr [rax + 0x40]
0xdb44: add     rsi, qword ptr [rax + 0x58]
0xdb48: lea     rdx, [rcx + 1]
0xdb4c: call    0x4bb0
0xdb51: mov     rax, qword ptr [rip + 0x1d8b0]
0xdb58: pop     rbx
0xdb59: mov     eax, dword ptr [rax + 0x48]
0xdb5c: ret
0xdb5d: nop     dword ptr [rax]
0xdb60: endbr64
0xdb64: push    rbp
0xdb65: xor     eax, eax
0xdb67: mov     rbp, rdi
0xdb6a: call    0xcff0
0xdb6f: test    rbp, rbp
0xdb72: lea     rax, [rip + 0x133bf]
0xdb79: cmov    rbp, rax
0xdb7d: mov     rdi, rbp
0xdb80: call    0xd540
0xdb85: xor     eax, eax
0xdb87: call    0xd870
0xdb8c: xor     eax, eax
0xdb8e: pop     rbp
0xdb8f: jmp     0xd740
0xd740
0xdb94: nop     word ptr cs:[rax + rax]
0xdb9f: nop
0xdba0: endbr64
0xdba4: mov     rax, qword ptr [rip + 0x1d85d]
0xdbab: mov     dword ptr [rax + 0xc], edi
0xdbae: ret
0xdbaf: nop
0xdbb0: endbr64
0xdbb4: push    r12

```

```

0xdbb6: push    rbp
0xdbb7: push    rbx
0xdbb8: cmp     edi, 1
0xdbbb: je      0xdbd0
0xdbbd: cmp     edi, 2
0xdbc0: je      0xdc90
0xdbc6: test    edi, edi
0xdbc8: je      0xdc40
0xdbca: pop     rbx
0xdbc b: pop     rbp
0xdbcc: pop     r12
0xdbce: ret
0xdbcf: nop
0xdbd0: mov     ecx, dword ptr [rip + 0x1d83e]
0xdbd6: test    ecx, ecx
0xdbd8: je      0xdd38
0xdbde: xor     eax, eax
0xdbe0: call    0x1ab10
0xdbe5: mov     rbp, rax
0xdbe8: test    rax, rax
0xdbeb: je      0xdbca
0xdbed: mov     r12, qword ptr [rbp]
0xdbf1: mov     rbx, qword ptr [rip + 0x1d810]
0xdbf8: mov     rdi, r12
0xdbfb: call    0x49b0
0xdc00: mov     rdi, qword ptr [rbx + 0x40]
0xdc04: mov     qword ptr [rbx + 0x48], rax
0xdc08: lea     rsi, [rax + rdi]
0xdc0c: mov     qword ptr [rbx + 0x38], rsi
0xdc10: mov     qword ptr [rbx + 0x18], rsi
0xdc14: add     rsi, 1
0xdc18: cmp     rsi, qword ptr [rbx + 0x50]
0xdc1c: ja      0xdd70
0xdc22: add     rdi, qword ptr [rbx + 0x58]
0xdc26: mov     rsi, r12
0xdc29: call    0x48b0
0xdc2e: pop     rbx
0xdc2f: xor     eax, eax
0xdc31: pop     rbp
0xdc32: pop     r12
0xdc34: jmp     0xd870
0xd870:
0xdc39: nop     dword ptr [rax]
0xdc40: mov     eax, dword ptr [rip + 0x1d7ce]
0xdc46: test    eax, eax
0xdc48: jne     0xdd18
0xdc4e: xor     eax, eax
0xdc50: call    0x1ab10

```

```

0xdc55: test    rax, rax
0xdc58: je      0xdd98
0xdc5e: mov     rdx, qword ptr [rip + 0x1d7a3]
0xdc65: mov     rsi, qword ptr [rax]
0xdc68: mov     rbp, qword ptr [rdx + 0x40]
0xdc6c: add     rbp, qword ptr [rdx + 0x58]
0xdc70: mov     rdi, rbp
0xdc73: call    0x4b30
0xdc78: test    eax, eax
0xdc7a: jne     0xdd58
0xdc80: pop     rbx
0xdc81: xor     eax, eax
0xdc83: pop     rbp
0xdc84: pop     r12
0xdc86: jmp     0x1ab40
0x1ab40
0xdc8b: nop     dword ptr [rax + rax]
0xdc90: mov     edx, dword ptr [rip + 0x1d77e]
0xdc96: test    edx, edx
0xdc98: je      0xdcf8
0xdc9a: xor     eax, eax
0xdc9c: call    0x1ab40
0xdca1: mov     rbp, rax
0xdca4: test    rax, rax
0xdca7: jne     0xdbed
0xdcad: mov     rax, qword ptr [rip + 0x1d754]
0xdcb4: mov     edi, 2
0xdcb9: mov     rdx, qword ptr [rax + 0x40]
0xdcbd: mov     dword ptr [rax + 0x10], 6
0xdcc4: mov     qword ptr [rax + 0x20], rdx
0xdcc8: mov     rdx, qword ptr [rax + 0x38]
0xdccc: mov     qword ptr [rax + 0x18], rdx
0xdcd0: xor     eax, eax
0xdcd2: call    0xc6e0
0xdcd7: mov     rax, qword ptr [rip + 0x1d72a]
0xdcde: mov     rdx, qword ptr [rax + 0x18]
0xdce2: mov     dword ptr [rax + 0x10], 0xd
0xdce9: mov     qword ptr [rax + 0x20], rdx
0xdced: jmp     0xdc2e
0xdc2e
0xdcf2: nop     word ptr [rax + rax]
0xdcf8: mov     dword ptr [rip + 0x1d712], 1
0xdd02: xor     eax, eax
0xdd04: call    0x1a570
0xdd09: mov     dword ptr [rip + 0x1d701], eax
0xdd0f: jmp     0xdc9a
0xdc9a
0xdd11: nop     dword ptr [rax]

```

```

0xdd18: mov     edi, dword ptr [rip + 0x1d6f2]
0xdd1e: xor     eax, eax
0xdd20: call    0x1ac30
0xdd25: mov     dword ptr [rip + 0x1d6e5], 0
0xdd2f: jmp     0xdc4e
0xdc4e:
0xdd34: nop     dword ptr [rax]
0xdd38: mov     dword ptr [rip + 0x1d6d2], 1
0xdd42: xor     eax, eax
0xdd44: call    0x1a570
0xdd49: mov     dword ptr [rip + 0x1d6c1], eax
0xdd4f: jmp     0xdbde
0xdbde:
0xdd54: nop     dword ptr [rax]
0xdd58: xor     eax, eax
0xdd5a: mov     rdi, rbp
0xdd5d: call    0x1a380
0xdd62: xor     eax, eax
0xdd64: call    0x1ab40
0xdd69: jmp     0xdc80
0xdc80:
0xdd6e: nop
0xdd70: mov     qword ptr [rbx + 0x50], rsi
0xdd74: mov     rdi, qword ptr [rbx + 0x58]
0xdd78: call    0x1d690
0xdd7d: mov     qword ptr [rbx + 0x58], rax
0xdd81: mov     rbx, qword ptr [rip + 0x1d680]
0xdd88: mov     r12, qword ptr [rbp]
0xdd8c: mov     rdi, qword ptr [rbx + 0x40]
0xdd90: jmp     0xdc22
0xdc22:
0xdd95: nop     dword ptr [rax]
0xdd98: mov     rax, qword ptr [rip + 0x1d669]
0xdd9f: mov     rdi, qword ptr [rax + 0x40]
0xdda3: add     rdi, qword ptr [rax + 0x58]
0xdda7: xor     eax, eax
0xdda9: call    0x1a380
0xddae: jmp     0xdc80
0xdc80:
0xddb3: nop     word ptr cs:[rax + rax]
0xddbd: nop     dword ptr [rax]
0dddc0: endbr64
0dddc4: mov     eax, dword ptr [rip + 0x1be96]
0xddca: push    rbp
0ddccb: mov     rbp, rdi
0ddcce: lea     rdi, [rip + 0x1be73]
0ddd5: test    eax, eax
0ddd7: lea     rax, [rip + 0x1be52]

```

```

0xddde: cmove    rdi, rax
0xdde2: call     0x17c50
0dde7: mov      edx, 9
0dddec: lea      rsi, [rip + 0x1c4ad]
0dddf3: lea      rdi, [rip + 0x1c466]
0dddfa: call     0x17f70
0dddff: mov      rdi, rbp
0xde02: call     0x1d7a0
0xde07: mov      rdi, rax
0xde0a: mov      rbp, rax
0xde0d: mov      qword ptr [rip + 0x1d794], rax
0xde14: call     0x49b0
0xde19: mov      rdi, rbp
0xde1c: mov      rsi, rax
0xde1f: call     0x184d0
0xde24: xor      eax, eax
0xde26: call     0xc590
0xde2b: lea      rdi, [rip + 0x1d5ee]
0xde32: pop      rbp
0xde33: mov      qword ptr [rip + 0x1d78e], rax
0xde3a: jmp      0x4a30
0x4a30
0xde3f: nop
0xde40: endbr64
0xde44: mov      rdi, qword ptr [rip + 0x1d77d]
0xde4b: jmp      0xc5c0
0xc5c0
0xde50: endbr64
0xde54: push     rbp
0xde55: mov      rbp, rdi
0xde58: push     rbx
0xde59: mov      rbx, rsi
0xde5c: sub      rsp, 8
0xde60: mov      rdi, qword ptr [rip + 0x1d749]
0xde67: test     rdi, rdi
0xde6a: je       0xde71
0xde6c: call     0x18d10
0xde71: mov      rdi, rbp
0xde74: call     0x1d670
0xde79: mov      rdi, qword ptr [rip + 0x1d748]
0xde80: mov      edx, ebx
0xde82: xor      esi, esi
0xde84: mov      qword ptr [rip + 0x1d725], rax
0xde8b: add      rsp, 8
0xde8f: mov      r8d, ebp
0xde92: mov      ecx, 1
0xde97: pop      rbx
0xde98: pop      rbp

```



```

0xde99: jmp      0xc5e0
0xc5e0
0xde9e: nop
0xdea0: endbr64
0xdea4: push     rbp
0xdea5: mov      rbp, rsp
0xdea8: push     r15
0xdeaa: push     r14
0xdeac: push     r13
0xdeae: push     r12
0xdeb0: push     rbx
0xdeb1: sub      rsp, 0x68
0xdeb5: mov      rax, qword ptr fs:[0x28]
0xdebe: mov      qword ptr [rbp - 0x38], rax
0xdec2: xor      eax, eax
0xdec4: lea      rax, [rbp - 0x61]
0xdec8: mov      rdi, rax
0xdecb: mov      qword ptr [rbp - 0x80], rax
0xdecf: call     0xa6a0
0xded4: mov      rax, qword ptr [rip + 0x1d6ed]
0xdedb: mov      esi, 0x20
0xdee0: mov      rdi, qword ptr [rip + 0x1d6c9]
0xdee7: movsxd   rdx, dword ptr [rax + 0xc]
0xdeeb: call     0x4a50
0xdef0: mov      rdi, qword ptr [rip + 0x1d6c9]
0xdef7: call     0x49b0
0xdefc: lea      r14, [rax + 1]
0xdf00: mov      rdi, r14
0xdf03: call     0x1d670
0xdf08: mov      rcx, qword ptr [rip + 0x1d6b1]
0xdf0f: mov      r13, rax
0xdf12: movzx    eax, byte ptr [rcx]
0xdf15: test     al, al
0xdf17: je       0xe430
0xdf1d: xor      esi, esi
0xdf1f: xor      ebx, ebx
0xdf21: lea      r15, [rip + 0x13188]
0xdf28: xor      edx, edx
0xdf2a: mov      dword ptr [rbp - 0x74], esi
0xdf2d: jmp      0xdf48
0xdf48
0xdf2f: nop
0xdf30: cmp      al, 9
0xdf32: je       0xdf80
0xdf34: add      dword ptr [rbp - 0x74], 1
0xdf38: mov      byte ptr [r13 + r12], al
0xdf3d: movsxd   rdx, ebx
0xdf40: movzx    eax, byte ptr [rcx + rdx]

```

```

0xdf44: test    al, al
0xdf46: je       0xdfc0
0xdf48: movsxd   r12, dword ptr [rbp - 0x74]
0xdf4c: add      ebx, 1
0xdf4f: cmp      al, 0x5c
0xdf51: jne      0xdf30
0xdf53: lea      rdx, [rcx + rdx + 1]
0xdf58: movzx    eax, byte ptr [rdx]
0xdf5b: test     al, al
0xdf5d: je       0xe320
0xdf63: sub      eax, 0x5c
0xdf66: cmp      al, 0x17
0xdf68: ja       0xe0d8
0xdf6e: movzx    eax, al
0xdf71: movsxd   rax, dword ptr [r15 + rax*4]
0xdf75: add      rax, r15
0xdf78: jmp      rax
rax
->->->                                indirect JMP

0x1d958: movsx    esi, dl
0x1d95b: mov      rdi, r12
0x1d95e: mov      byte ptr [rsp + 8], dl
0x1d962: call     0x4a00
0x1d967: movzx    edx, byte ptr [rsp + 8]
0x1d96c: test     rax, rax
0x1d96f: jne      0x1d905
0x1d971: mov      qword ptr [r14], rbx
0x1d974: or       ebp, 2
0x1d977: jmp      0x1d896
0x1d896
0x1d97c: mov      esi, 1
0x1d981: mov      ecx, 0x400
0x1d986: sub      edx, 0x42
0x1d989: cmp      dl, 0x35
0x1d98c: ja       0x1d971
0x1d98e: lea      rdi, [rip + 0x4c93]
0x1d995: movzx    edx, dl
0x1d998: movsxd   rax, dword ptr [rdi + rdx*4]
0x1d99c: add      rax, rdi
0x1d99f: jmp      rax
rax
->->->                                indirect JMP

0x1dbf9: call     0x49d0
0x1dbfe: mov      eax, 1
0x1dc03: or       rbx, 0xffffffffffffffff
0x1dc07: or       ebp, eax
0x1dc09: jmp      0x1d9d7
0x1d9d7

```

```

0x1dc0e:      mov     r8d, 1
0x1dc14:      or      rbx, 0xffffffffffffffff
0x1dc18:      jmp     0x1d9cf
0x1d9cf:
0x1dc1d:      mov     r8d, 1
0x1dc23:      or      rbx, 0xffffffffffffffff
0x1dc27:      jmp     0x1da67
0x1da67:
0x1dc2c:      mov     r8d, 1
0x1dc32:      or      rbx, 0xffffffffffffffff
0x1dc36:      jmp     0x1da8f
0x1da8f:
0x1dc3b:      mov     r8d, 1
0x1dc41:      or      rbx, 0xffffffffffffffff
0x1dc45:      jmp     0x1dacf
0x1dacf:
0x1dc4a:      mov     r8d, 1
0x1dc50:      or      rbx, 0xffffffffffffffff
0x1dc54:      jmp     0x1db27
0x1db27:
0x1dc59:      nop     dword ptr [rax]
0x1dc60:      endbr64
0x1dc64:      push    rax
0x1dc65:      pop     rax
0x1dc66:      sub     rsp, 8
0x1dc6a:      mov     edi, dword ptr [rip + 0xcb90]
0x1dc70:      call    0x4e20
0x1dc75:      nop     word ptr cs:[rax + rax]
0x1dc80:      endbr64
0x1dc84:      push    r15
0x1dc86:      push    r14
0x1dc88:      mov     r14, rdi
0x1dc8b:      push    r13
0x1dc8d:      mov     r13, rcx
0x1dc90:      push    r12
0x1dc92:      push    rbp
0x1dc93:      mov     rbp, rdx
0x1dc96:      push    rbx
0x1dc97:      mov     rbx, rsi
0x1dc9a:      sub     rsp, 0x28
0x1dc9e:      mov     qword ptr [rsp + 8], rsi
0x1dca3:      mov     qword ptr [rsp + 0x18], rdx
0x1dca8:      call    0x49b0
0x1dcad:      mov     r15, qword ptr [rbx]
0x1dcb0:      test    r15, r15
0x1dcb3:      je      0x1dd86
0x1dcb9:      mov     byte ptr [rsp + 0x17], 0
0x1dcbe:      mov     r12, rax

```

```

0x1dcc1:      xor     ebx, ebx
0x1dcc3:      mov     qword ptr [rsp], -1
0x1dccb:      jmp     0x1dd18
0x1dd18:
0x1dccd:      nop     dword ptr [rax]
0x1dcd0:      mov     rax, qword ptr [rsp + 0x18]
0x1dcd5:      test    rax, rax
0x1dcd8:      je      0x1dd48
0x1dcda:      mov     rdi, qword ptr [rsp]
0x1dcde:      mov     rdx, r13
0x1dce1:      mov     rsi, rbp
0x1dce4:      imul    rdi, r13
0x1dce8:      add     rdi, rax
0x1dceb:      call    0x4af0
0x1dcf0:      mov     ecx, 1
0x1dcf5:      test    eax, eax
0x1dcf7:      movzx   eax, byte ptr [rsp + 0x17]
0x1dcfc:      cmovne  eax, ecx
0x1dcff:      mov     byte ptr [rsp + 0x17], al
0x1dd03:      mov     rax, qword ptr [rsp + 8]
0x1dd08:      add     rbx, 1
0x1dd0c:      add     rbp, r13
0x1dd0f:      mov     r15, qword ptr [rax + rbx*8]
0x1dd13:      test    r15, r15
0x1dd16:      je      0x1dd50
0x1dd18:      mov     rdx, r12
0x1dd1b:      mov     rsi, r14
0x1dd1e:      mov     rdi, r15
0x1dd21:      call    0x4890
0x1dd26:      test    eax, eax
0x1dd28:      jne     0x1dd03
0x1dd2a:      mov     rdi, r15
0x1dd2d:      call    0x49b0
0x1dd32:      cmp     rax, r12
0x1dd35:      je      0x1dd80
0x1dd37:      cmp     qword ptr [rsp], -1
0x1dd3c:      jne     0x1dcd0
0x1dd3e:      mov     qword ptr [rsp], rbx
0x1dd42:      jmp     0x1dd03
0x1dd03:
0x1dd44:      nop     dword ptr [rax]
0x1dd48:      mov     byte ptr [rsp + 0x17], 1
0x1dd4d:      jmp     0x1dd03
0x1dd03:
0x1dd4f:      nop
0x1dd50:      cmp     byte ptr [rsp + 0x17], 0
0x1dd55:      mov     rax, -2
0x1dd5c:      cmov    rax, qword ptr [rsp]

```

```

0x1dd61:      mov     qword ptr [rsp], rax
0x1dd65:      mov     rax, qword ptr [rsp]
0x1dd69:      add     rsp, 0x28
0x1dd6d:      pop     rbx
0x1dd6e:      pop     rbp
0x1dd6f:      pop     r12
0x1dd71:      pop     r13
0x1dd73:      pop     r14
0x1dd75:      pop     r15
0x1dd77:      ret
0x1dd78:      nop     dword ptr [rax + rax]
0x1dd80:      mov     qword ptr [rsp], rbx
0x1dd84:      jmp     0x1dd65
0x1dd65:
0x1dd86:      mov     qword ptr [rsp], -1
0x1dd8e:      jmp     0x1dd65
0x1dd65:
0x1dd90:      endbr64
0x1dd94:      push    r13
0x1dd96:      cmp     rdx, -1
0x1dd9a:      mov     r13, rsi
0x1dd9d:      lea     rax, [rip + 0x4977]
0x1dda4:      push    r12
0x1dda6:      mov     rsi, rdi
0x1dda9:      lea     r12, [rip + 0x4950]
0x1ddb0:      mov     edi, 1
0x1ddb5:      push    rbx
0x1ddb6:      cmovne  r12, rax
0x1ddba:      call    0x1e000
0x1ddbf:      mov     rdx, r13
0x1ddc2:      mov     esi, 5
0x1ddc7:      xor     edi, edi
0x1ddc9:      mov     rbx, rax
0x1ddcc:      call    0x1ee20
0x1ddd1:      mov     r8, rbx
0x1ddd4:      mov     rdx, r12
0x1ddd7:      pop     rbx
0x1ddd8:      mov     rcx, rax
0x1ddd9:      pop     r12
0x1dddd:      xor     esi, esi
0x1dddf:      xor     edi, edi
0x1dde1:      xor     eax, eax
0x1dde3:      pop     r13
0x1dde5:      jmp     0x4d50
0x4d50:
0x1ddea:      nop     word ptr [rax + rax]
0x1ddf0:      endbr64
0x1ddf4:      push    r15

```

```

0x1ddf6:    push    r14
0x1ddf8:    lea     r14, [rip + 0x494e]
0x1ddff:    push    r13
0x1de01:    xor     r13d, r13d
0x1de04:    push    r12
0x1de06:    mov     r12, rdx
0x1de09:    mov     edx, 0x14
0x1de0e:    push    rbp
0x1de0f:    mov     rbp, rsi
0x1de12:    mov     esi, 1
0x1de17:    push    rbx
0x1de18:    mov     rbx, rdi
0x1de1b:    sub     rsp, 0x18
0x1de1f:    mov     rcx, qword ptr [rip + 0xca5a]
0x1de26:    mov     qword ptr [rsp + 8], rdi
0x1de2b:    lea     rdi, [rip + 0x4906]
0x1de32:    call    0x4e40
0x1de37:    mov     r15, qword ptr [rbx]
0x1de3a:    xor     ebx, ebx
0x1de3c:    test    r15, r15
0x1de3f:    jne     0x1de75
0x1de41:    jmp     0x1dec0
0x1dec0:
0x1de43:    nop     dword ptr [rax + rax]
0x1de48:    mov     rcx, r15
0x1de4b:    mov     rdx, r14
0x1de4e:    mov     esi, 1
0x1de53:    mov     rdi, r8
0x1de56:    xor     eax, eax
0x1de58:    mov     r13, rbp
0x1de5b:    call    0x4e60
0x1de60:    mov     rax, qword ptr [rsp + 8]
0x1de65:    add     rbx, 1
0x1de69:    add     rbp, r12
0x1de6c:    mov     r15, qword ptr [rax + rbx*8]
0x1de70:    test    r15, r15
0x1de73:    je      0x1dec0
0x1de75:    mov     r8, qword ptr [rip + 0xca04]
0x1de7c:    test    rbx, rbx
0x1de7f:    je      0x1de48
0x1de81:    mov     rdx, r12
0x1de84:    mov     rsi, rbp
0x1de87:    mov     rdi, r13
0x1de8a:    mov     qword ptr [rsp], r8
0x1de8e:    call    0x4af0
0x1de93:    mov     r8, qword ptr [rsp]
0x1de97:    test    eax, eax
0x1de99:    jne     0x1de48

```

```

0x1de9b:      mov     rcx, r15
0x1de9e:      lea     rdx, [rip + 0x48b2]
0x1dea5:      mov     rdi, r8
0x1dea8:      xor     eax, eax
0x1deaa:      mov     esi, 1
0x1deaf:      call    0x4e60
0x1deb4:      jmp     0x1de60
0x1de60:
0x1deb6:      nop     word ptr cs:[rax + rax]
0x1dec0:      mov     rsi, qword ptr [rip + 0xc9b9]
0x1dec7:      add     rsp, 0x18
0x1decb:      mov     edi, 0xa
0x1ded0:      pop     rbx
0x1ded1:      pop     rbp
0x1ded2:      pop     r12
0x1ded4:      pop     r13
0x1ded6:      pop     r14
0x1ded8:      pop     r15
0x1deda:      jmp     0x4b40
0x4b40:
0x1dedf:      nop
0x1dee0:      endbr64
0x1dee4:      push    r15
0x1dee6:      mov     r15, rsi
0x1dee9:      push    r14
0x1deeb:      mov     r14, r8
0x1deee:      push    r13
0x1def0:      mov     r13, rdi
0x1def3:      mov     rdi, r15
0x1def6:      push    r12
0x1def8:      mov     r12, rcx
0x1defb:      mov     rcx, r8
0x1defe:      push    rbp
0x1deff:      mov     rbp, rdx
0x1df02:      mov     rdx, r12
0x1df05:      push    rbx
0x1df06:      mov     rsi, rbp
0x1df09:      mov     rbx, r9
0x1df0c:      sub     rsp, 0x18
0x1df10:      call    0x1dc80
0x1df15:      test    rax, rax
0x1df18:      js      0x1df30
0x1df1a:      add     rsp, 0x18
0x1df1e:      pop     rbx
0x1df1f:      pop     rbp
0x1df20:      pop     r12
0x1df22:      pop     r13
0x1df24:      pop     r14

```

```

0x1df26:    pop     r15
0x1df28:    ret
0x1df29:    nop     dword ptr [rax]
0x1df30:    cmp     rax, -1
0x1df34:    lea     r9, [rip + 0x47c5]
0x1df3b:    mov     rsi, r13
0x1df3e:    mov     edi, 1
0x1df43:    lea     rax, [rip + 0x47d1]
0x1df4a:    cmovne  r9, rax
0x1df4e:    mov     qword ptr [rsp + 8], r9
0x1df53:    call    0x1e000
0x1df58:    mov     rdx, r15
0x1df5b:    mov     esi, 5
0x1df60:    xor     edi, edi
0x1df62:    mov     r13, rax
0x1df65:    call    0x1ee20
0x1df6a:    mov     rdx, qword ptr [rsp + 8]
0x1df6f:    mov     r8, r13
0x1df72:    xor     esi, esi
0x1df74:    mov     rcx, rax
0x1df77:    xor     edi, edi
0x1df79:    xor     eax, eax
0x1df7b:    call    0x4d50
0x1df80:    mov     rdx, r14
0x1df83:    mov     rsi, r12
0x1df86:    mov     rdi, rbp
0x1df89:    call    0x1ddf0
0x1df8e:    call    rbx
0x1df90:    mov     rax, -1
0x1df97:    jmp     0x1df1a
0x1df1a:
0x1df99:    nop     dword ptr [rax]
0x1dfa0:    endbr64
0x1dfa4:    push    r14
0x1dfa6:    push    r13
0x1dfa8:    push    r12
0x1dfaa:    push    rbp
0x1dfab:    push    rbx
0x1dfac:    mov     r14, qword ptr [rsi]
0x1dfaf:    test    r14, r14
0x1dfb2:    je      0x1dfe9
0x1dfb4:    mov     r12, rdi
0x1dfb7:    mov     r13, rcx
0x1dfba:    lea     rbx, [rsi + 8]
0x1dfbe:    mov     rbp, rdx
0x1dfc1:    jmp     0x1dfd7
0x1dfd7:
0x1dfc3:    nop     dword ptr [rax + rax]

```



```

0x1dfc8:    mov     r14, qword ptr [rbx]
0x1dfcb:    add     rbp, r13
0x1dfce:    add     rbx, 8
0x1dfd2:    test    r14, r14
0x1dfd5:    je      0x1dfe9
0x1dfd7:    mov     rdx, r13
0x1dfda:    mov     rsi, rbp
0x1dfdd:    mov     rdi, r12
0x1dfe0:    call    0x4af0
0x1dfe5:    test    eax, eax
0x1dfe7:    jne     0x1dfc8
0x1dfe9:    pop     rbx
0x1dfea:    mov     rax, r14
0x1dfed:    pop     rbp
0x1dfee:    pop     r12
0x1dff0:    pop     r13
0x1dff2:    pop     r14
0x1dff4:    ret
0x1dff5:    nop     word ptr cs:[rax + rax]
0x1dfff:    nop
0x1e000:    endbr64
0x1e004:    mov     rdx, rsi
0x1e007:    mov     esi, 5
0x1e00c:    jmp     0x1ee20
0x1ee20:
0x1e011:    nop     word ptr cs:[rax + rax]
0x1e01c:    nop     dword ptr [rax]
0x1e020:    endbr64
0x1e024:    mov     rdx, rdi
0x1e027:    mov     esi, 5
0x1e02c:    xor     edi, edi
0x1e02e:    jmp     0x1ee20
0x1ee20:
0x1e033:    nop     word ptr cs:[rax + rax]
0x1e03d:    nop     dword ptr [rax]
0x1e040:    push    r15
0x1e042:    push    r14
0x1e044:    push    r13
0x1e046:    mov     r13, rcx
0x1e049:    push    r12
0x1e04b:    push    rbp
0x1e04c:    push    rbx
0x1e04d:    mov     ebx, r8d
0x1e050:    sub     rsp, 0x88
0x1e057:    mov     qword ptr [rsp + 8], rdi
0x1e05c:    mov     qword ptr [rsp + 0x30], rsi
0x1e061:    mov     qword ptr [rsp + 0x20], rdx
0x1e066:    mov     dword ptr [rsp + 0x10], r8d

```

```

0x1e06b:    mov     qword ptr [rsp + 0x18], r9
0x1e070:    mov     rax, qword ptr fs:[0x28]
0x1e079:    mov     qword ptr [rsp + 0x78], rax
0x1e07e:    xor     eax, eax
0x1e080:    call    0x49a0
0x1e085:    cmp     ebx, 4
0x1e088:    mov     r11, qword ptr [rsp + 0x20]
0x1e08d:    mov     r10, qword ptr [rsp + 0x30]
0x1e092:    mov     qword ptr [rsp + 0x28], rax
0x1e097:    jne     0x1e832
0x1e09d:    mov     qword ptr [rsp + 0x20], 0
0x1e0a6:    mov     r8d, 1
0x1e0ac:    xor     r9d, r9d
0x1e0af:    nop
0x1e0b0:    mov     rbx, r9
0x1e0b3:    xor     r12d, r12d
0x1e0b6:    nop     word ptr cs:[rax + rax]
0x1e0c0:    cmp     r13, r12
0x1e0c3:    setne   al
0x1e0c6:    cmp     r13, -1
0x1e0ca:    jne     0x1e0d4
0x1e0cc:    cmp     byte ptr [r11 + r12], 0
0x1e0d1:    setne   al
0x1e0d4:    test    al, al
0x1e0d6:    je      0x1e7ae
0x1e0dc:    lea     rsi, [r11 + r12]
0x1e0e0:    movzx   edx, byte ptr [rsi]
0x1e0e3:    test    r9, r9
0x1e0e6:    je      0x1e4b0
0x1e0ec:    test    r8b, r8b
0x1e0ef:    je      0x1e4b0
0x1e0f5:    lea     r14, [r12 + 1]
0x1e0fa:    mov     r15, r14
0x1e0fd:    cmp     r14, r13
0x1e100:    ja      0x1e120
0x1e102:    mov     rax, qword ptr [rsp + 0x20]
0x1e107:    cmp     byte ptr [rax], dl
0x1e109:    jne     0x1e120
0x1e10b:    cmp     rbx, r10
0x1e10e:    jae     0x1e11c
0x1e110:    mov     rax, qword ptr [rsp + 8]
0x1e115:    mov     byte ptr [rax + rbx], 0x5c
0x1e119:    movzx   edx, byte ptr [rsi]
0x1e11c:    add     rbx, 1
0x1e120:    mov     ebp, edx
0x1e122:    cmp     dl, 0x3f
0x1e125:    jg      0x1e458
0x1e12b:    test    dl, dl

```

```

0x1e12d:      js      0x1e240
0x1e133:      cmp     dl, 0x3f
0x1e136:      ja      0x1e240
0x1e13c:      lea     rcx, [rip + 0x464d]
0x1e143:      movzx   eax, dl
0x1e146:      movsxd  rax, dword ptr [rcx + rax*4]
0x1e14a:      add     rax, rcx
0x1e14d:      jmp     rax
rax
->->->                                indirect JMP
0x1d9a2:      mov     esi, 2
0x1d9a7:      mov     ecx, 0x3e8
0x1d9ac:      jmp     0x1d986
0x1d986
0x1d9ae:      movsxd  rsi, esi
0x1d9b1:      mov     edi, 4
0x1d9b6:      xor     r8d, r8d
0x1d9b9:      nop     dword ptr [rax]
0x1d9c0:      mov     rax, rbx
0x1d9c3:      mul     rcx
0x1d9c6:      mov     rbx, rax
0x1d9c9:      jo      0x1dc0e
0x1d9cf:      sub     edi, 1
0x1d9d2:      jne     0x1d9c0
0x1d9d4:      or      ebp, r8d
0x1d9d7:      add     rsi, r13
0x1d9da:      mov     eax, ebp
0x1d9dc:      or      eax, 2
0x1d9df:      mov     qword ptr [r15], rsi
0x1d9e2:      cmp     byte ptr [rsi], 0
0x1d9e5:      cmovne  ebp, eax
0x1d9e8:      jmp     0x1d893
0x1d893
0x1d9ed:      movsxd  rsi, esi
0x1d9f0:      mov     rax, rbx
0x1d9f3:      mul     rcx
0x1d9f6:      jo      0x1da13
0x1d9f8:      mul     rcx
0x1d9fb:      jo      0x1da13
0x1d9fd:      mov     rbx, rax
0x1da00:      jmp     0x1d9d7
0x1d9d7
0x1da02:      movsxd  rdx, ecx
0x1da05:      movsxd  rsi, esi
0x1da08:      mov     rax, rbx
0x1da0b:      mul     rdx
0x1da0e:      mov     rbx, rax
0x1da11:      jno     0x1d9d7

```

```

0x1da13:      mov     ebp, 1
0x1da18:      or      rbx, 0xfffffffffffffff
0x1da1c:      jmp     0x1d9d7
0x1d9d7:
0x1da1e:      nop
0x1da20:      movsxd   rsi, esi
0x1da23:      mov     rax, rbx
0x1da26:      mul     rcx
0x1da29:      jo      0x1dbfe
0x1da2f:      mul     rcx
0x1da32:      jo      0x1dbfe
0x1da38:      mul     rcx
0x1da3b:      jo      0x1dbfe
0x1da41:      mov     rbx, rax
0x1da44:      xor     eax, eax
0x1da46:      or      ebp, eax
0x1da48:      jmp     0x1d9d7
0x1d9d7:
0x1da4a:      movsxd   rsi, esi
0x1da4d:      mov     edi, 5
0x1da52:      xor     r8d, r8d
0x1da55:      nop     dword ptr [rax]
0x1da58:      mov     rax, rbx
0x1da5b:      mul     rcx
0x1da5e:      mov     rbx, rax
0x1da61:      jo      0x1dc1d
0x1da67:      sub     edi, 1
0x1da6a:      jne     0x1da58
0x1da6c:      jmp     0x1d9d4
0x1d9d4:
0x1da71:      movsxd   rsi, esi
0x1da74:      mov     edi, 6
0x1da79:      xor     r8d, r8d
0x1da7c:      nop     dword ptr [rax]
0x1da80:      mov     rax, rbx
0x1da83:      mul     rcx
0x1da86:      mov     rbx, rax
0x1da89:      jo      0x1dc2c
0x1da8f:      sub     edi, 1
0x1da92:      jne     0x1da80
0x1da94:      jmp     0x1d9d4
0x1d9d4:
0x1da99:      movsxd   rsi, esi
0x1da9c:      mov     rax, rbx
0x1da9f:      shr     rax, 0x36
0x1daa3:      jne     0x1da13
0x1daa9:      shl     rbx, 0xa
0x1daad:      jmp     0x1d9d7

```

```

0x1d9d7
0x1dab2:    movsxd   rsi, esi
0x1dab5:    mov      edi, 8
0x1daba:    xor      r8d, r8d
0x1dabd:    nop      dword ptr [rax]
0x1dac0:    mov      rax, rbx
0x1dac3:    mul      rcx
0x1dac6:    mov      rbx, rax
0x1dac9:    jo       0x1dc3b
0x1dacf:    sub      edi, 1
0x1dad2:    jne      0x1dac0
0x1dad4:    jmp      0x1d9d4
0x1d9d4
0x1dad9:    movsxd   rsi, esi
0x1dadc:    test     rbx, rbx
0x1dadf:    js       0x1da13
0x1dae5:    add      rbx, rbx
0x1dae8:    jmp      0x1d9d7
0x1d9d7
0x1daed:    movsxd   rsi, esi
0x1daf0:    mov      rax, rbx
0x1daf3:    shr      rax, 0x37
0x1daf7:    jne      0x1da13
0x1dafd:    shl      rbx, 9
0x1db01:    jmp      0x1d9d7
0x1d9d7
0x1db06:    movsxd   rsi, esi
0x1db09:    mov      edi, 7
0x1db0e:    xor      r8d, r8d
0x1db11:    nop      dword ptr [rax]
0x1db18:    mov      rax, rbx
0x1db1b:    mul      rcx
0x1db1e:    mov      rbx, rax
0x1db21:    jo       0x1dc4a
0x1db27:    sub      edi, 1
0x1db2a:    jne      0x1db18
0x1db2c:    jmp      0x1d9d4
0x1d9d4
0x1db31:    movsxd   rsi, esi
0x1db34:    jmp      0x1d9d7
0x1d9d7
0x1db39:    mov      esi, 1
0x1db3e:    mov      ecx, 0x400
0x1db43:    jmp      0x1d9b1
0x1d9b1
0x1db48:    mov      esi, 1
0x1db4d:    mov      ecx, 0x400
0x1db52:    jmp      0x1d9f0

```

```

0x1d9f0
0x1db57:      mov     esi, 1
0x1db5c:      mov     edx, 0x400
0x1db61:      jmp     0x1da08
0x1da08
0x1db66:      mov     esi, 1
0x1db6b:      mov     ecx, 0x400
0x1db70:      jmp     0x1da23
0x1da23
0x1db75:      mov     esi, 1
0x1db7a:      mov     ecx, 0x400
0x1db7f:      jmp     0x1dab5
0x1dab5
0x1db84:      mov     esi, 1
0x1db89:      mov     ecx, 0x400
0x1db8e:      jmp     0x1da4d
0x1da4d
0x1db93:      mov     esi, 1
0x1db98:      mov     ecx, 0x400
0x1db9d:      jmp     0x1da74
0x1da74
0x1dba2:      mov     esi, 1
0x1dba7:      jmp     0x1da9c
0x1da9c
0x1dbac:      mov     esi, 1
0x1dbb1:      jmp     0x1dadc
0x1dadc
0x1dbb6:      mov     esi, 1
0x1dbbb:      jmp     0x1d9d7
0x1d9d7
0x1dbc0:      mov     esi, 1
0x1dbc5:      jmp     0x1daf0
0x1daf0
0x1dbca:      mov     esi, 1
0x1dbcf:      mov     ecx, 0x400
0x1dbd4:      jmp     0x1db09
0x1db09
0x1dbd9:      nop     dword ptr [rax]
0x1dbe0:      xor     esi, esi
0x1dbe2:      cmp     byte ptr [r13 + 2], 0x42
0x1dbe7:      mov     ecx, 0x400
0x1dbec:      sete    sil
0x1dbf0:      lea     esi, [rsi + rsi + 1]
0x1dbf4:      jmp     0x1d986
0x1d986
0x1dbf9:      call    0x49d0
0x1dbfe:      mov     eax, 1
0x1dc03:      or      rbx, 0xffffffffffffffff

```

```

0x1dc07:      or      ebp, eax
0x1dc09:      jmp     0x1d9d7
0x1d9d7
0x1dc0e:      mov     r8d, 1
0x1dc14:      or      rbx, 0xffffffffffffffff
0x1dc18:      jmp     0x1d9cf
0x1d9cf
0x1dc1d:      mov     r8d, 1
0x1dc23:      or      rbx, 0xffffffffffffffff
0x1dc27:      jmp     0x1da67
0x1da67
0x1dc2c:      mov     r8d, 1
0x1dc32:      or      rbx, 0xffffffffffffffff
0x1dc36:      jmp     0x1da8f
0x1da8f
0x1dc3b:      mov     r8d, 1
0x1dc41:      or      rbx, 0xffffffffffffffff
0x1dc45:      jmp     0x1dacf
0x1dacf
0x1dc4a:      mov     r8d, 1
0x1dc50:      or      rbx, 0xffffffffffffffff
0x1dc54:      jmp     0x1db27
0x1db27
0x1dc59:      nop     dword ptr [rax]
0x1dc60:      endbr64
0x1dc64:      push    rax
0x1dc65:      pop     rax
0x1dc66:      sub     rsp, 8
0x1dc6a:      mov     edi, dword ptr [rip + 0xcb90]
0x1dc70:      call    0x4e20
0x1dc75:      nop     word ptr cs:[rax + rax]
0x1dc80:      endbr64
0x1dc84:      push    r15
0x1dc86:      push    r14
0x1dc88:      mov     r14, rdi
0x1dc8b:      push    r13
0x1dc8d:      mov     r13, rcx
0x1dc90:      push    r12
0x1dc92:      push    rbp
0x1dc93:      mov     rbp, rdx
0x1dc96:      push    rbx
0x1dc97:      mov     rbx, rsi
0x1dc9a:      sub     rsp, 0x28
0x1dc9e:      mov     qword ptr [rsp + 8], rsi
0x1dca3:      mov     qword ptr [rsp + 0x18], rdx
0x1dca8:      call    0x49b0
0x1dcad:      mov     r15, qword ptr [rbx]
0x1dcb0:      test    r15, r15

```

```

0x1dcb3:      je      0x1dd86
0x1dcb9:      mov     byte ptr [rsp + 0x17], 0
0x1dcbe:      mov     r12, rax
0x1dcc1:      xor     ebx, ebx
0x1dcc3:      mov     qword ptr [rsp], -1
0x1dccb:      jmp     0x1dd18
0x1dd18:
0x1dccd:      nop     dword ptr [rax]
0x1dcd0:      mov     rax, qword ptr [rsp + 0x18]
0x1dcd5:      test    rax, rax
0x1dcd8:      je      0x1dd48
0x1dcda:      mov     rdi, qword ptr [rsp]
0x1dcde:      mov     rdx, r13
0x1dce1:      mov     rsi, rbp
0x1dce4:      imul    rdi, r13
0x1dce8:      add     rdi, rax
0x1dceb:      call    0x4af0
0x1dcf0:      mov     ecx, 1
0x1dcf5:      test    eax, eax
0x1dcf7:      movzx   eax, byte ptr [rsp + 0x17]
0x1dcfc:      cmovne  eax, ecx
0x1dcff:      mov     byte ptr [rsp + 0x17], al
0x1dd03:      mov     rax, qword ptr [rsp + 8]
0x1dd08:      add     rbx, 1
0x1dd0c:      add     rbp, r13
0x1dd0f:      mov     r15, qword ptr [rax + rbx*8]
0x1dd13:      test    r15, r15
0x1dd16:      je      0x1dd50
0x1dd18:      mov     rdx, r12
0x1dd1b:      mov     rsi, r14
0x1dd1e:      mov     rdi, r15
0x1dd21:      call    0x4890
0x1dd26:      test    eax, eax
0x1dd28:      jne     0x1dd03
0x1dd2a:      mov     rdi, r15
0x1dd2d:      call    0x49b0
0x1dd32:      cmp     rax, r12
0x1dd35:      je      0x1dd80
0x1dd37:      cmp     qword ptr [rsp], -1
0x1dd3c:      jne     0x1dcd0
0x1dd3e:      mov     qword ptr [rsp], rbx
0x1dd42:      jmp     0x1dd03
0x1dd03:
0x1dd44:      nop     dword ptr [rax]
0x1dd48:      mov     byte ptr [rsp + 0x17], 1
0x1dd4d:      jmp     0x1dd03
0x1dd03:
0x1dd4f:      nop

```



```

0x1dd50:      cmp     byte ptr [rsp + 0x17], 0
0x1dd55:      mov     rax, -2
0x1dd5c:      cmovne  rax, qword ptr [rsp]
0x1dd61:      mov     qword ptr [rsp], rax
0x1dd65:      mov     rax, qword ptr [rsp]
0x1dd69:      add     rsp, 0x28
0x1dd6d:      pop     rbx
0x1dd6e:      pop     rbp
0x1dd6f:      pop     r12
0x1dd71:      pop     r13
0x1dd73:      pop     r14
0x1dd75:      pop     r15
0x1dd77:      ret
0x1dd78:      nop     dword ptr [rax + rax]
0x1dd80:      mov     qword ptr [rsp], rbx
0x1dd84:      jmp     0x1dd65
0x1dd86:      mov     qword ptr [rsp], -1
0x1dd8e:      jmp     0x1dd65
0x1dd90:      endbr64
0x1dd94:      push    r13
0x1dd96:      cmp     rdx, -1
0x1dd9a:      mov     r13, rsi
0x1dd9d:      lea     rax, [rip + 0x4977]
0x1dda4:      push    r12
0x1dda6:      mov     rsi, rdi
0x1dda9:      lea     r12, [rip + 0x4950]
0x1ddb0:      mov     edi, 1
0x1ddb5:      push    rbx
0x1ddb6:      cmovne  r12, rax
0x1ddba:      call    0x1e000
0x1ddbf:      mov     rdx, r13
0x1ddc2:      mov     esi, 5
0x1ddc7:      xor     edi, edi
0x1ddc9:      mov     rbx, rax
0x1ddcc:      call    0x1ee20
0x1ddd1:      mov     r8, rbx
0x1ddd4:      mov     rdx, r12
0x1ddd7:      pop     rbx
0x1ddd8:      mov     rcx, rax
0x1ddd9:      pop     r12
0x1dddd:      xor     esi, esi
0x1ddd9:      xor     edi, edi
0x1dde1:      xor     eax, eax
0x1dde3:      pop     r13
0x1dde5:      jmp     0x4d50
0x4d50

```

```

0x1ddea:      nop      word ptr [rax + rax]
0x1ddf0:      endbr64
0x1ddf4:      push     r15
0x1ddf6:      push     r14
0x1ddf8:      lea      r14, [rip + 0x494e]
0x1ddff:      push     r13
0x1de01:      xor      r13d, r13d
0x1de04:      push     r12
0x1de06:      mov      r12, rdx
0x1de09:      mov      edx, 0x14
0x1de0e:      push     rbp
0x1de0f:      mov      rbp, rsi
0x1de12:      mov      esi, 1
0x1de17:      push     rbx
0x1de18:      mov      rbx, rdi
0x1de1b:      sub      rsp, 0x18
0x1de1f:      mov      rcx, qword ptr [rip + 0xca5a]
0x1de26:      mov      qword ptr [rsp + 8], rdi
0x1de2b:      lea      rdi, [rip + 0x4906]
0x1de32:      call     0x4e40
0x1de37:      mov      r15, qword ptr [rbx]
0x1de3a:      xor      ebx, ebx
0x1de3c:      test     r15, r15
0x1de3f:      jne      0x1de75
0x1de41:      jmp      0x1dec0
0x1dec0:
0x1de43:      nop      dword ptr [rax + rax]
0x1de48:      mov      rcx, r15
0x1de4b:      mov      rdx, r14
0x1de4e:      mov      esi, 1
0x1de53:      mov      rdi, r8
0x1de56:      xor      eax, eax
0x1de58:      mov      r13, rbp
0x1de5b:      call     0x4e60
0x1de60:      mov      rax, qword ptr [rsp + 8]
0x1de65:      add      rbx, 1
0x1de69:      add      rbp, r12
0x1de6c:      mov      r15, qword ptr [rax + rbx*8]
0x1de70:      test     r15, r15
0x1de73:      je       0x1dec0
0x1de75:      mov      r8, qword ptr [rip + 0xca04]
0x1de7c:      test     rbx, rbx
0x1de7f:      je       0x1de48
0x1de81:      mov      rdx, r12
0x1de84:      mov      rsi, rbp
0x1de87:      mov      rdi, r13
0x1de8a:      mov      qword ptr [rsp], r8
0x1de8e:      call     0x4af0

```

```

0x1de93:    mov     r8, qword ptr [rsp]
0x1de97:    test    eax, eax
0x1de99:    jne     0x1de48
0x1de9b:    mov     rcx, r15
0x1de9e:    lea     rdx, [rip + 0x48b2]
0x1dea5:    mov     rdi, r8
0x1dea8:    xor     eax, eax
0x1deaa:    mov     esi, 1
0x1deaf:    call    0x4e60
0x1deb4:    jmp     0x1de60
0x1de60:
0x1deb6:    nop     word ptr cs:[rax + rax]
0x1dec0:    mov     rsi, qword ptr [rip + 0xc9b9]
0x1dec7:    add     rsp, 0x18
0x1decb:    mov     edi, 0xa
0x1ded0:    pop     rbx
0x1ded1:    pop     rbp
0x1ded2:    pop     r12
0x1ded4:    pop     r13
0x1ded6:    pop     r14
0x1ded8:    pop     r15
0x1deda:    jmp     0x4b40
0x4b40:
0x1dedf:    nop
0x1dee0:    endbr64
0x1dee4:    push    r15
0x1dee6:    mov     r15, rsi
0x1dee9:    push    r14
0x1deeb:    mov     r14, r8
0x1deee:    push    r13
0x1def0:    mov     r13, rdi
0x1def3:    mov     rdi, r15
0x1def6:    push    r12
0x1def8:    mov     r12, rcx
0x1defb:    mov     rcx, r8
0x1defe:    push    rbp
0x1deff:    mov     rbp, rdx
0x1df02:    mov     rdx, r12
0x1df05:    push    rbx
0x1df06:    mov     rsi, rbp
0x1df09:    mov     rbx, r9
0x1df0c:    sub     rsp, 0x18
0x1df10:    call    0x1dc80
0x1df15:    test    rax, rax
0x1df18:    js      0x1df30
0x1df1a:    add     rsp, 0x18
0x1df1e:    pop     rbx
0x1df1f:    pop     rbp

```

```

0x1df20:    pop     r12
0x1df22:    pop     r13
0x1df24:    pop     r14
0x1df26:    pop     r15
0x1df28:    ret
0x1df29:    nop     dword ptr [rax]
0x1df30:    cmp     rax, -1
0x1df34:    lea     r9, [rip + 0x47c5]
0x1df3b:    mov     rsi, r13
0x1df3e:    mov     edi, 1
0x1df43:    lea     rax, [rip + 0x47d1]
0x1df4a:    cmovne  r9, rax
0x1df4e:    mov     qword ptr [rsp + 8], r9
0x1df53:    call    0x1e000
0x1df58:    mov     rdx, r15
0x1df5b:    mov     esi, 5
0x1df60:    xor     edi, edi
0x1df62:    mov     r13, rax
0x1df65:    call    0x1ee20
0x1df6a:    mov     rdx, qword ptr [rsp + 8]
0x1df6f:    mov     r8, r13
0x1df72:    xor     esi, esi
0x1df74:    mov     rcx, rax
0x1df77:    xor     edi, edi
0x1df79:    xor     eax, eax
0x1df7b:    call    0x4d50
0x1df80:    mov     rdx, r14
0x1df83:    mov     rsi, r12
0x1df86:    mov     rdi, rbp
0x1df89:    call    0x1ddf0
0x1df8e:    call    rbx
0x1df90:    mov     rax, -1
0x1df97:    jmp     0x1df1a
0x1df1a:
0x1df99:    nop     dword ptr [rax]
0x1dfa0:    endbr64
0x1dfa4:    push    r14
0x1dfa6:    push    r13
0x1dfa8:    push    r12
0x1dfaa:    push    rbp
0x1dfab:    push    rbx
0x1dfac:    mov     r14, qword ptr [rsi]
0x1dfaf:    test    r14, r14
0x1dfb2:    je      0x1dfe9
0x1dfb4:    mov     r12, rdi
0x1dfb7:    mov     r13, rcx
0x1dfba:    lea     rbx, [rsi + 8]
0x1dfbe:    mov     rbp, rdx

```

```

0x1dfc1:      jmp      0x1dfd7
0x1dfd7:
0x1dfc3:      nop      dword ptr [rax + rax]
0x1dfc8:      mov      r14, qword ptr [rbx]
0x1dfcb:      add      rbp, r13
0x1dfce:      add      rbx, 8
0x1dfd2:      test     r14, r14
0x1dfd5:      je       0x1dfe9
0x1dfd7:      mov      rdx, r13
0x1dfda:      mov      rsi, rbp
0x1dfdd:      mov      rdi, r12
0x1dfe0:      call     0x4af0
0x1dfe5:      test     eax, eax
0x1dfe7:      jne      0x1dfc8
0x1dfe9:      pop      rbx
0x1dfea:      mov      rax, r14
0x1dfed:      pop      rbp
0x1dfee:      pop      r12
0x1dff0:      pop      r13
0x1dff2:      pop      r14
0x1dff4:      ret
0x1dff5:      nop      word ptr cs:[rax + rax]
0x1dfff:      nop
0x1e000:      endbr64
0x1e004:      mov      rdx, rsi
0x1e007:      mov      esi, 5
0x1e00c:      jmp      0x1ee20
0x1ee20:
0x1e011:      nop      word ptr cs:[rax + rax]
0x1e01c:      nop      dword ptr [rax]
0x1e020:      endbr64
0x1e024:      mov      rdx, rdi
0x1e027:      mov      esi, 5
0x1e02c:      xor      edi, edi
0x1e02e:      jmp      0x1ee20
0x1ee20:
0x1e033:      nop      word ptr cs:[rax + rax]
0x1e03d:      nop      dword ptr [rax]
0x1e040:      push     r15
0x1e042:      push     r14
0x1e044:      push     r13
0x1e046:      mov      r13, rcx
0x1e049:      push     r12
0x1e04b:      push     rbp
0x1e04c:      push     rbx
0x1e04d:      mov      ebx, r8d
0x1e050:      sub      rsp, 0x88
0x1e057:      mov      qword ptr [rsp + 8], rdi

```

```

0x1e05c:    mov     qword ptr [rsp + 0x30], rsi
0x1e061:    mov     qword ptr [rsp + 0x20], rdx
0x1e066:    mov     dword ptr [rsp + 0x10], r8d
0x1e06b:    mov     qword ptr [rsp + 0x18], r9
0x1e070:    mov     rax, qword ptr fs:[0x28]
0x1e079:    mov     qword ptr [rsp + 0x78], rax
0x1e07e:    xor     eax, eax
0x1e080:    call    0x49a0
0x1e085:    cmp     ebx, 4
0x1e088:    mov     r11, qword ptr [rsp + 0x20]
0x1e08d:    mov     r10, qword ptr [rsp + 0x30]
0x1e092:    mov     qword ptr [rsp + 0x28], rax
0x1e097:    jne     0x1e832
0x1e09d:    mov     qword ptr [rsp + 0x20], 0
0x1e0a6:    mov     r8d, 1
0x1e0ac:    xor     r9d, r9d
0x1e0af:    nop
0x1e0b0:    mov     rbx, r9
0x1e0b3:    xor     r12d, r12d
0x1e0b6:    nop     word ptr cs:[rax + rax]
0x1e0c0:    cmp     r13, r12
0x1e0c3:    setne   al
0x1e0c6:    cmp     r13, -1
0x1e0ca:    jne     0x1e0d4
0x1e0cc:    cmp     byte ptr [r11 + r12], 0
0x1e0d1:    setne   al
0x1e0d4:    test    al, al
0x1e0d6:    je      0x1e7ae
0x1e0dc:    lea     rsi, [r11 + r12]
0x1e0e0:    movzx   edx, byte ptr [rsi]
0x1e0e3:    test    r9, r9
0x1e0e6:    je      0x1e4b0
0x1e0ec:    test    r8b, r8b
0x1e0ef:    je      0x1e4b0
0x1e0f5:    lea     r14, [r12 + 1]
0x1e0fa:    mov     r15, r14
0x1e0fd:    cmp     r14, r13
0x1e100:    ja      0x1e120
0x1e102:    mov     rax, qword ptr [rsp + 0x20]
0x1e107:    cmp     byte ptr [rax], dl
0x1e109:    jne     0x1e120
0x1e10b:    cmp     rbx, r10
0x1e10e:    jae     0x1e11c
0x1e110:    mov     rax, qword ptr [rsp + 8]
0x1e115:    mov     byte ptr [rax + rbx], 0x5c
0x1e119:    movzx   edx, byte ptr [rsi]
0x1e11c:    add     rbx, 1
0x1e120:    mov     ebp, edx

```

```

0x1e122:    cmp     dl, 0x3f
0x1e125:    jg      0x1e458
0x1e12b:    test    dl, dl
0x1e12d:    js      0x1e240
0x1e133:    cmp     dl, 0x3f
0x1e136:    ja      0x1e240
0x1e13c:    lea     rcx, [rip + 0x464d]
0x1e143:    movzx   eax, dl
0x1e146:    movsxd  rax, dword ptr [rcx + rax*4]
0x1e14a:    add     rax, rcx
0x1e14d:    jmp     rax
rax
->->->                                indirect JMP
0x10660:    mov     rax, qword ptr [rsp + 0x298]
0x10668:    sub     rax, qword ptr fs:[0x28]
0x10671:    jne     0x1075a
0x10677:    add     rsp, 0x2a0
0x1067e:    lea     rdi, [rip + 0x10b67]
0x10685:    pop     rbx
0x10686:    pop     rbp
0x10687:    pop     r12
0x10689:    jmp     0x8b00
0x8b00
0x1068e:    nop
0x10690:    mov     rax, qword ptr [r12]
0x10694:    mov     qword ptr [rcx], rax
0x10697:    mov     eax, edx
0x10699:    mov     rsi, qword ptr [r12 + rax - 8]
0x1069e:    mov     qword ptr [rcx + rax - 8], rsi
0x106a3:    lea     rsi, [rcx + 8]
0x106a7:    and     rsi, 0xfffffffffffffff8
0x106ab:    sub     rcx, rsi
0x106ae:    add     edx, ecx
0x106b0:    sub     r12, rcx
0x106b3:    and     edx, 0xffffffff8
0x106b6:    cmp     edx, 8
0x106b9:    jb      0x104d0
0x106bf:    and     edx, 0xffffffff8
0x106c2:    xor     eax, eax
0x106c4:    mov     ecx, eax
0x106c6:    add     eax, 8
0x106c9:    mov     rdi, qword ptr [r12 + rcx]
0x106cd:    mov     qword ptr [rsi + rcx], rdi
0x106d1:    cmp     eax, edx
0x106d3:    jb      0x106c4
0x106d5:    jmp     0x104d0
0x104d0
0x106da:    nop     word ptr [rax + rax]

```

```

0x106e0:    mov     esi, eax
0x106e2:    lea     r8, [rcx + 8]
0x106e6:    mov     qword ptr [rcx], rdi
0x106e9:    and     r8, 0xfffffffffffffff8
0x106ed:    mov     qword ptr [rcx + rsi - 8], rdi
0x106f2:    mov     rsi, rcx
0x106f5:    sub     rsi, r8
0x106f8:    add     eax, esi
0x106fa:    and     eax, 0xffffffff8
0x106fd:    cmp     eax, 8
0x10700:    jb      0x1061e
0x10706:    and     eax, 0xffffffff8
0x10709:    xor     esi, esi
0x1070b:    mov     r9d, esi
0x1070e:    add     esi, 8
0x10711:    mov     qword ptr [r8 + r9], rdi
0x10715:    cmp     esi, eax
0x10717:    jb      0x1070b
0x10719:    jmp     0x1061e
0x1061e:
0x1071e:    mov     eax, dword ptr [r12]
0x10722:    mov     edx, edx
0x10724:    mov     dword ptr [rcx], eax
0x10726:    mov     eax, dword ptr [r12 + rdx - 4]
0x1072b:    mov     dword ptr [rcx + rdx - 4], eax
0x1072f:    jmp     0x104d0
0x104d0:
0x10734:    mov     eax, eax
0x10736:    mov     dword ptr [rcx], 0x20202020
0x1073c:    mov     dword ptr [rcx + rax - 4], 0x20202020
0x10744:    jmp     0x1061e
0x1061e:
0x10749:    mov     eax, eax
0x1074b:    mov     esi, 0x2020
0x10750:    mov     word ptr [rcx + rax - 2], si
0x10755:    jmp     0x1061e
0x1061e:
0x1075a:    call    0x49d0
0x1075f:    nop
0x10760:    push    r15
0x10762:    push    r14
0x10764:    mov     r14d, 4
0x1076a:    push    r13
0x1076c:    mov     r13, rdi
0x1076f:    push    r12
0x10771:    push    rbp
0x10772:    push    rbx
0x10773:    mov     ebx, esi

```



```

0x10775:      mov     esi, 0x20
0x1077a:      sub     rsp, 0x28
0x1077e:      movsxd  rdx, dword ptr [rdi + 0x64]
0x10782:      mov     rdi, qword ptr [rdi + 0x20]
0x10786:      call    0x4a50
0x1078b:      mov     ebp, dword ptr [r13 + 0x64]
0x1078f:      cmp     ebp, 0x27
0x10792:      jle     0x107b0
0x10794:      movsxd  rax, dword ptr [r13 + 0x80]
0x1079b:      cmp     ebp, 0x49
0x1079e:      jg      0x107a5
0x107a0:      cmp     eax, 5
0x107a3:      je      0x107b0
0x107a5:      lea     rdx, [rip + 0x11234]
0x107ac:      movsxd  r14, dword ptr [rdx + rax*4]
0x107b0:      movsxd  rax, ebx
0x107b3:      lea     r12, [rax + rax*4]
0x107b7:      mov     rax, qword ptr [r13 + 0x38]
0x107bb:      shl     r12, 4
0x107bf:      add     rax, r12
0x107c2:      test    ebx, ebx
0x107c4:      jle     0x107e8
0x107c6:      cmp     byte ptr [rax + 0x48], 4
0x107ca:      ja      0x4f20
0x107d0:      movzx   edx, byte ptr [rax + 0x48]
0x107d4:      lea     rcx, [rip + 0x11101]
0x107db:      movsxd  rdx, dword ptr [rcx + rdx*4]
0x107df:      add     rdx, rcx
0x107e2:      jmp     rdx
rdx
->->->                                indirect JMP
0x84f0: mov     esi, 3
0x84f5: mov     edi, r15d
0x84f8: call    0x7570
0x84fd: test    eax, eax
0x84ff: je      0x8670
0x8505: mov     eax, dword ptr [rsp + 0x1c]
0x8509: lea     edx, [rax - 1]
0x850c: mov     dword ptr [rsp + 0x1c], edx
0x8510: test    eax, eax
0x8512: jne     0x84f0
0x8514: call    0xd870
0x8519: xor     eax, eax
0x851b: call    0xde0
0x8520: xor     eax, eax
0x8522: call    0xd740
0x8527: xor     eax, eax
0x8529: call    0xa820

```

```

0x852e: mov     rdi, r12
0x8531: call    0xb180
0x8536: test    rax, rax
0x8539: jne     0x848e
0x853f: xor     eax, eax
0x8541: call    0xa5a0
0x8546: xor     eax, eax
0x8548: call    0xdea0
0x854d: xor     eax, eax
0x854f: call    0xd740
0x8554: jmp     0x847d
0x847d:
0x8559: nop     dword ptr [rax]
0x8560: mov     rdi, r13
0x8563: call    0xdb10
0x8568: test    rbp, rbp
0x856b: je      0x857b
0x856d: mov     rsi, qword ptr [r13]
0x8571: mov     rdi, rbp
0x8574: xor     eax, eax
0x8576: call    0x7e00
0x857b: mov     rax, qword ptr [r13]
0x857f: mov     rdx, qword ptr [rsp + 0x28]
0x8584: sub     rdx, qword ptr fs:[0x28]
0x858d: jne     0x86b3
0x8593: add     rsp, 0x38
0x8597: pop     rbx
0x8598: pop     rbp
0x8599: pop     r12
0x859b: pop     r13
0x859d: pop     r14
0x859f: pop     r15
0x85a1: ret
0x85a2: nop     word ptr [rax + rax]
0x85a8: test    rbp, rbp
0x85ab: je      0x8519
0x85b1: lea     eax, [r14 + 1]
0x85b5: mov     rsi, qword ptr [rsp]
0x85b9: mov     rdi, rbp
0x85bc: mov     edx, eax
0x85be: mov     r15d, eax
0x85c1: call    0x19200
0x85c6: test    rax, rax
0x85c9: je      0x86a7
0x85cf: mov     r8, qword ptr [rsp + 0x20]
0x85d4: test    r8, r8
0x85d7: je      0x86a7
0x85dd: xor     edi, edi

```

```

0x85df: mov     qword ptr [rsp + 8], r8
0x85e4: mov     r14d, r15d
0x85e7: call    0xd1a0
0x85ec: mov     rdi, qword ptr [rsp + 8]
0x85f1: call    0xd610
0x85f6: xor     eax, eax
0x85f8: call    0xd870
0x85fd: xor     eax, eax
0x85ff: call    0xd740
0x8604: jmp     0x8519
0x8519:
0x8609: nop     dword ptr [rax]
0x8610: test    rbp, rbp
0x8613: je      0x8519
0x8619: xor     edi, edi
0x861b: test    r14d, r14d
0x861e: je      0x8680
0x8620: call    0xd1a0
0x8625: sub     r14d, 1
0x8629: je      0x864b
0x862b: mov     rsi, qword ptr [rsp]
0x862f: mov     rdi, rbp
0x8632: mov     edx, r14d
0x8635: call    0x19200
0x863a: mov     rdi, rax
0x863d: test    rax, rax
0x8640: cmovne  rdi, qword ptr [rsp + 0x20]
0x8646: call    0xd610
0x864b: xor     eax, eax
0x864d: call    0xd870
0x8652: xor     eax, eax
0x8654: call    0xd740
0x8659: jmp     0x8519
0x8519:
0x865e: nop
0x8660: xor     edi, edi
0x8662: xor     eax, eax
0x8664: call    0x7bd0
0x8669: jmp     0x8519
0x8519:
0x866e: nop
0x8670: call    0xa5a0
0x8675: jmp     0x8505
0x8505:
0x867a: nop     word ptr [rax + rax]
0x8680: call    0xd1a0
0x8685: xor     eax, eax
0x8687: call    0xd870

```

```

0x868c: xor     eax, eax
0x868e: call    0xd740
0x8693: jmp     0x8519
0x8519:
0x8698: nop     dword ptr [rax + rax]
0x86a0: xor     eax, eax
0x86a2: jmp     0x857f
0x857f:
0x86a7: xor     eax, eax
0x86a9: call    0xa5a0
0x86ae: jmp     0x8519
0x8519:
0x86b3: call    0x49d0
0x86b8: nop     dword ptr [rax + rax]
0x86c0: endbr64
0x86c4: push    r15
0x86c6: push    r14
0x86c8: push    r13
0x86ca: push    r12
0x86cc: push    rbp
0x86cd: push    rbx
0x86ce: sub     rsp, 0x18
0x86d2: test    edx, edx
0x86d4: je      0x8890
0x86da: mov     ebx, edx
0x86dc: cmp     edx, 3
0x86df: je      0x88f8
0x86e5: mov     rbp, rcx
0x86e8: test    rcx, rcx
0x86eb: je      0x8864
0x86f1: mov     dword ptr [rcx], 0
0x86f7: xor     eax, eax
0x86f9: mov     r13, rsi
0x86fc: lea     r15, [rip + 0x2091d]
0x8703: movabs  r14, 0xea0ea0ea0ea0ea1
0x870d: call    0xd870
0x8712: xor     eax, eax
0x8714: call    0xd740
0x8719: xor     eax, eax
0x871b: call    0xa820
0x8720: xor     edi, edi
0x8722: call    0xb180
0x8727: mov     r12, rax
0x872a: test    rax, rax
0x872d: je      0x8928
0x8733: mov     rax, qword ptr [rax]
0x8736: movzx   edi, byte ptr [rax]
0x8739: mov     rax, qword ptr [r12 + 0x10]

```

```

0x873e: test    rax, rax
0x8741: je      0x874d
0x8743: cmp     byte ptr [rax + 0x1b], 0
0x8747: jne     0x8808
0x874d: cmp     edi, -7
0x8750: jge     0x87b0
0x8752: cmp     edi, -0x3d
0x8755: je      0x8840
0x875b: cmp     edi, -0x32
0x875e: jne     0x88bb
0x8764: xor     eax, eax
0x8766: call    0xca40
0x876b: test    eax, eax
0x876d: jne     0x8880
0x8773: mov     dword ptr [rbp], 1
0x877a: xor     eax, eax
0x877c: call    0xdea0
0x8781: xor     eax, eax
0x8783: call    0xd870
0x8788: xor     eax, eax
0x878a: call    0xd740
0x878f: xor     eax, eax
0x8791: call    0xa820
0x8796: mov     rdi, r13
0x8799: call    0xdb10
0x879e: mov     r12, qword ptr [r13]
0x87a2: jmp     0x8867
0x8867:
0x87a7: nop     word ptr [rax + rax]
0x87b0: cmp     edi, 7
0x87b3: je      0x8840
0x87b9: mov     dword ptr [rsp + 8], edi
0x87bd: call    0x4f00
0x87c2: mov     r8, rax
0x87c5: movsxd  rax, dword ptr [rsp + 8]
0x87ca: mov     rdx, qword ptr [r8]
0x87cd: mov     rdi, rax
0x87d0: test    byte ptr [rdx + rax*2 + 1], 0x40
0x87d5: jne     0x8948
0x87db: xor     eax, eax
0x87dd: xor     r12d, r12d
0x87e0: call    0xdea0
0x87e5: xor     eax, eax
0x87e7: call    0xd870
0x87ec: xor     eax, eax
0x87ee: call    0xd740
0x87f3: xor     eax, eax
0x87f5: call    0xa820

```

```

0x87fa: mov     rdi, r13
0x87fd: call    0xdb10
0x8802: jmp     0x8867
0x8867
0x8804: nop     dword ptr [rax]
0x8808: mov     rcx, qword ptr [rax]
0x880b: sub     rcx, r15
0x880e: mov     rax, rcx
0x8811: sar     rcx, 0x3f
0x8815: imul    r14
0x8818: sar     rdx, 1
0x881b: sub     rdx, rcx
0x881e: mov     edi, edx
0x8820: not     edi
0x8822: cmp     edx, 7
0x8825: jne     0x874d
0x882b: xor     edi, edi
0x882d: xor     eax, eax
0x882f: call    0x7bd0
0x8834: jmp     0x8720
0x8720
0x8839: nop     dword ptr [rax]
0x8840: xor     eax, eax
0x8842: call    0xdea0
0x8847: xor     eax, eax
0x8849: call    0xd870
0x884e: xor     eax, eax
0x8850: call    0xd740
0x8855: xor     eax, eax
0x8857: call    0xa820
0x885c: mov     rdi, r13
0x885f: call    0xdb10
0x8864: xor     r12d, r12d
0x8867: add     rsp, 0x18
0x886b: mov     rax, r12
0x886e: pop     rbx
0x886f: pop     rbp
0x8870: pop     r12
0x8872: pop     r13
0x8874: pop     r14
0x8876: pop     r15
0x8878: ret
0x8879: nop     dword ptr [rax]
0x8880: xor     eax, eax
0x8882: call    0xa5a0
0x8887: jmp     0x877a
0x877a
0x888c: nop     dword ptr [rax]

```

```

0x8890: xor     eax, eax
0x8892: mov     qword ptr [rsp + 8], rdi
0x8897: call    0xc8c0
0x889c: mov     qword ptr [rip + 0x2205d], rax
0x88a3: xor     eax, eax
0x88a5: call    0xcff0
0x88aa: mov     rdi, qword ptr [rsp + 8]
0x88af: test    rdi, rdi
0x88b2: je      0x8864
0x88b4: call    0xd540
0x88b9: jmp     0x8864
0x8864:
0x88bb: cmp     edi, -0x44
0x88be: jne     0x88df
0x88c0: cmp     ebx, 1
0x88c3: jne     0x88df
0x88c5: xor     eax, eax
0x88c7: call    0xca40
0x88cc: cmp     eax, 1
0x88cf: mov     eax, 2
0x88d4: sbb     eax, -1
0x88d7: mov     dword ptr [rbp], eax
0x88da: jmp     0x877a
0x877a:
0x88df: cmp     edi, -0x45
0x88e2: jne     0x87b9
0x88e8: cmp     ebx, 2
0x88eb: je      0x88c5
0x88ed: jmp     0x87b9
0x87b9:
0x88f2: nop     word ptr [rax + rax]
0x88f8: mov     rdi, qword ptr [rip + 0x22001]
0x88ff: xor     r12d, r12d
0x8902: call    0xc970
0x8907: xor     eax, eax
0x8909: call    0xd870
0x890e: xor     eax, eax
0x8910: call    0xd740
0x8915: xor     eax, eax
0x8917: call    0xa820
0x891c: jmp     0x8867
0x8867:
0x8921: nop     dword ptr [rax]
0x8928: xor     eax, eax
0x892a: call    0xdea0
0x892f: xor     eax, eax
0x8931: call    0xd740
0x8936: xor     eax, eax

```

```

0x8938: call    0xa820
0x893d: jmp     0x8867
0x8867:
0x8942: nop      word ptr [rax + rax]
0x8948: call    0xcd30
0x894d: mov     dword ptr [rbp], 3
0x8954: jmp     0x877a
0x877a:
0x8959: nop      dword ptr [rax]
0x8960: endbr64
0x8964: mov     rdi, qword ptr [rip + 0x21f2d]
0x896b: push    rbx
0x896c: test    rdi, rdi
0x896f: je      0x89ac
0x8971: mov     eax, dword ptr [rip + 0x21f1d]
0x8977: test    eax, eax
0x8979: jle     0x899c
0x897b: xor     ebx, ebx
0x897d: nop      dword ptr [rax]
0x8980: mov     rdi, qword ptr [rdi + rbx*8]
0x8984: add     rbx, 1
0x8988: call    0x18d10
0x898d: cmp     dword ptr [rip + 0x21f01], ebx
0x8993: mov     rdi, qword ptr [rip + 0x21efe]
0x899a: jg      0x8980
0x899c: call    0x18d10
0x89a1: mov     qword ptr [rip + 0x21eec], 0
0x89ac: mov     dword ptr [rip + 0x21ede], 0
0x89b6: pop     rbx
0x89b7: mov     dword ptr [rip + 0x21ecf], 0
0x89c1: ret
0x89c2: nop      word ptr cs:[rax + rax]
0x89cd: nop      dword ptr [rax]
0x89d0: endbr64
0x89d4: push    rbp
0x89d5: mov     rbp, rdi
0x89d8: push    rbx
0x89d9: sub     rsp, 8
0x89dd: mov     esi, dword ptr [rip + 0x21eb1]
0x89e3: mov     rdi, qword ptr [rip + 0x21eae]
0x89ea: mov     dword ptr [rip + 0x21ea0], esi
0x89f0: add     esi, 1
0x89f3: mov     dword ptr [rip + 0x21e9b], esi
0x89f9: movsxd  rsi, esi
0x89fc: shl     rsi, 3
0x8a00: call    0x1d690
0x8a05: movsxd  rdx, dword ptr [rip + 0x21e84]
0x8a0c: mov     rdi, rbp

```



```

0x8a0f: mov     qword ptr [rip + 0x21e82], rax
0x8a16: lea     rbx, [rax + rdx*8]
0x8a1a: call    0x1d7a0
0x8a1f: mov     qword ptr [rbx], rax
0x8a22: add     rsp, 8
0x8a26: pop     rbx
0x8a27: pop     rbp
0x8a28: ret
0x8a29: nop     dword ptr [rax]
0x8a30: endbr64
0x8a34: mov     rdx, rsi
0x8a37: mov     esi, dword ptr [rip + 0x21e57]
0x8a3d: mov     eax, dword ptr [rip + 0x21e4d]
0x8a43: lea     ecx, [rsi - 1]
0x8a46: cmp     ecx, eax
0x8a48: jle     0x8a70
0x8a4a: add     eax, 1
0x8a4d: mov     r8d, 1
0x8a53: mov     esi, 0x21
0x8a58: mov     rcx, qword ptr [rip + 0x21e39]
0x8a5f: mov     dword ptr [rip + 0x21e2b], eax
0x8a65: cdqe
0x8a67: mov     rcx, qword ptr [rcx + rax*8]
0x8a6b: jmp     0x13e30
0x13e30
0x8a70: xor     eax, eax
0x8a72: jmp     0xa5a0
0xa5a0
0x8a77: nop     word ptr [rax + rax]
0x8a80: endbr64
0x8a84: mov     eax, dword ptr [rip + 0x21e06]
0x8a8a: mov     rdx, rsi
0x8a8d: test    eax, eax
0x8a8f: je      0x8ac0
0x8a91: sub     eax, 1
0x8a94: mov     r8d, 1
0x8a9a: mov     esi, 0x21
0x8a9f: mov     rcx, qword ptr [rip + 0x21df2]
0x8aa6: mov     dword ptr [rip + 0x21de4], eax
0x8aac: cdqe
0x8aae: mov     rcx, qword ptr [rcx + rax*8]
0x8ab2: jmp     0x13e30
0x13e30
0x8ab7: nop     word ptr [rax + rax]
0x8ac0: xor     eax, eax
0x8ac2: jmp     0xa5a0
0xa5a0
0x8ac7: nop     word ptr [rax + rax]

```

```

0x8ad0: endbr64
0x8ad4: sub     rsp, 8
0x8ad8: xor      edi, edi
0x8ada: call     0xbbc0
0x8adf: xor      eax, eax
0x8ae1: call     0xde40
0x8ae6: xor      eax, eax
0x8ae8: add      rsp, 8
0x8aec: jmp      0x19580
0x19580
0x8af1: nop      word ptr cs:[rax + rax]
0x8afc: nop      dword ptr [rax]
0x8b00: endbr64
0x8b04: push     rbx
0x8b05: xor      eax, eax
0x8b07: mov      rbx, rdi
0x8b0a: call     0x9c90
0x8b0f: cmp      eax, 1
0x8b12: je       0x8b42
0x8b14: mov      rdi, qword ptr [rip + 0x21d65]
0x8b1b: mov      rcx, qword ptr [rip + 0x21dc6]
0x8b22: mov      r8, rbx
0x8b25: xor      eax, eax
0x8b27: lea      rdx, [rip + 0x179e8]
0x8b2e: mov      esi, 1
0x8b33: call     0x4e60
0x8b38: mov      edi, 1
0x8b3d: call     0x4e20
0x8b42: xor      eax, eax
0x8b44: call     0x8ad0
0x8b49: jmp      0x8b14
0x8b14
0x8b4b: nop      dword ptr [rax + rax]
0x8b50: endbr64
0x8b54: push     r15
0x8b56: push     r14
0x8b58: push     r13
0x8b5a: push     r12
0x8b5c: mov      r12, rsi
0x8b5f: push     rbp
0x8b60: push     rbx
0x8b61: sub      rsp, 0x68
0x8b65: mov      r14, qword ptr [rdi + 8]
0x8b69: mov      qword ptr [rsp + 0x18], rdi
0x8b6e: mov      qword ptr [rsp + 8], rdx
0x8b73: mov      rdi, r14
0x8b76: mov      qword ptr [rsp + 0x10], rcx
0x8b7b: mov      rax, qword ptr fs:[0x28]

```

```

0x8b84: mov     qword ptr [rsp + 0x58], rax
0x8b89: xor     eax, eax
0x8b8b: mov     qword ptr [rsp + 0x48], 0
0x8b94: mov     qword ptr [rsp + 0x50], 0
0x8b9d: call    0x49b0
0x8ba2: lea     r13, [rax + 1]
0x8ba6: mov     rdi, r13
0x8ba9: call    0x1d670
0x8bae: mov     qword ptr [r12], rax
0x8bb2: mov     rbx, rax
0x8bb5: movzx   eax, byte ptr [r14]
0x8bb9: test    al, al
0x8bbb: je      0x925c
0x8bc1: mov     dword ptr [rsp + 0x24], 0
0x8bc9: jmp     0x8be5
0x8be5:
0x8bcb: nop     dword ptr [rax + rax]
0x8bd0: mov     byte ptr [rbx], al
0x8bd2: mov     r14, rbp
0x8bd5: add     rbx, 1
0x8bd9: movzx   eax, byte ptr [r14]
0x8bdd: test    al, al
0x8bdf: je      0x8d38
0x8be5: lea     rbp, [r14 + 1]
0x8be9: cmp     al, 0x25
0x8beb: jne     0x8bd0
0x8bed: call    0x4f00
0x8bf2: movsx   rcx, byte ptr [r14 + 1]
0x8bf7: mov     r15, qword ptr [rsp + 0x10]
0x8bfc: mov     rax, qword ptr [rax]
0x8bff: mov     rdx, rcx
0x8c02: test    byte ptr [rax + rcx*2 + 1], 2
0x8c07: lea     eax, [rcx - 0x3f]
0x8c0a: cmovne  r15, qword ptr [rsp + 8]
0x8c10: cmp     al, 0x3b
0x8c12: ja      0x8c38
0x8c14: lea     rcx, [rip + 0x174f1]
0x8c1b: movzx   eax, al
0x8c1e: movsxd  rax, dword ptr [rcx + rax*4]
0x8c22: add     rax, rcx
0x8c25: jmp     rax
rax
->->->                                indirect JMP
0x13e84: movsxd  rax, dword ptr [rbp + 0x28]
0x13e88: mov     rdx, qword ptr [rbp + 0x18]
0x13e8c: mov     dword ptr [rbp + 0x84], r15d
0x13e93: mov     rdi, qword ptr [rbp + 0x38]
0x13e97: mov     dword ptr [rip + 0x1778a], r15d

```

```

0x13e9e:    lea     rax, [rax + rax*4]
0x13ea2:    shl     rax, 4
0x13ea6:    cmp     byte ptr [rdx], 0x2f
0x13ea9:    mov     rbx, qword ptr [rdi + rax]
0x13ead:    mov     eax, dword ptr [rbp + 0x68]
0x13eb0:    jne     0x13ebc
0x13eb2:    cmp     byte ptr [rdx + 1], 0
0x13eb6:    je      0x1498b
0x13ebc:    sub     eax, 1
0x13ebf:    add     rdi, 0x50
0x13ec3:    mov     edx, 0x50
0x13ec8:    movsxd  rsi, eax
0x13ecb:    lea     rcx, [rip - 0x5852]
0x13ed2:    call    0x48e0
0x13ed7:    mov     ecx, dword ptr [rbp + 0x68]
0x13eda:    test    ecx, ecx
0x13edc:    jle     0x14cab
0x13ee2:    mov     rdx, qword ptr [rbp + 0x38]
0x13ee6:    xor     eax, eax
0x13ee8:    jmp     0x13efb
0x13efb:
0x13eea:    nop     word ptr [rax + rax]
0x13ef0:    add     eax, 1
0x13ef3:    add     rdx, 0x50
0x13ef7:    cmp     eax, ecx
0x13ef9:    je      0x13f00
0x13efb:    cmp     qword ptr [rdx], rbx
0x13efe:    jne     0x13ef0
0x13f00:    mov     dword ptr [rbp + 0x28], eax
0x13f03:    mov     rdi, rbp
0x13f06:    xor     eax, eax
0x13f08:    call    0xf110
0x13f0d:    mov     dword ptr [rbp + 0x2c], eax
0x13f10:    jmp     0x13f40
0x13f40:
0x13f12:    nop     word ptr [rax + rax]
0x13f18:    sub     r15d, 9
0x13f1c:    cmp     dword ptr [rbp + 0x64], 0x4f
0x13f20:    mov     dword ptr [rbp + 0x80], r15d
0x13f27:    jg      0x13f40
0x13f29:    cmp     r15d, 5
0x13f2d:    jne     0x13f40
0x13f2f:    mov     dword ptr [rbp + 0x80], 0
0x13f39:    nop     dword ptr [rax]
0x13f40:    cmp     byte ptr [rbp + 0x6d], 0
0x13f44:    je      0x13f50
0x13f46:    mov     rdi, rbp
0x13f49:    xor     eax, eax

```

```

0x13f4b:    call    0x10d70
0x13f50:    mov     rax, qword ptr [rsp + 0x38]
0x13f55:    sub     rax, qword ptr fs:[0x28]
0x13f5e:    jne     0x14ce9
0x13f64:    add     rsp, 0x48
0x13f68:    xor     eax, eax
0x13f6a:    pop     rbx
0x13f6b:    pop     rbp
0x13f6c:    pop     r12
0x13f6e:    pop     r13
0x13f70:    pop     r14
0x13f72:    pop     r15
0x13f74:    ret
0x13f75:    nop     dword ptr [rax]
0x13f78:    movsxd  rax, dword ptr [rdi + 0x28]
0x13f7c:    mov     esi, 0x2e
0x13f81:    lea     rax, [rax + rax*4]
0x13f85:    shl     rax, 4
0x13f89:    add     rax, qword ptr [rdi + 0x38]
0x13f8d:    mov     r12, qword ptr [rax]
0x13f90:    mov     rdi, r12
0x13f93:    call    0x4a20
0x13f98:    mov     rbx, rax
0x13f9b:    test    rax, rax
0x13f9e:    je      0x13f50
0x13fa0:    cmp     r12, rax
0x13fa3:    je      0x13f50
0x13fa5:    mov     rdi, rax
0x13fa8:    call    0x49b0
0x13fad:    lea     rdi, [rax + 2]
0x13fb1:    call    0x4c20
0x13fb6:    mov     r8, rbx
0x13fb9:    mov     esi, 1
0x13fbe:    xor     ebx, ebx
0x13fc0:    mov     rdi, rax
0x13fc3:    mov     r13, rax
0x13fc6:    xor     eax, eax
0x13fc8:    mov     rdx, -1
0x13fcf:    lea     rcx, [rip + 0xd3be]
0x13fd6:    call    0x4f10
0x13fdb:    mov     eax, dword ptr [rbp + 0x68]
0x13fde:    test    eax, eax
0x13fe0:    jle     0x14132
0x13fe6:    nop     word ptr cs:[rax + rax]
0x13ff0:    mov     rax, qword ptr [rbp + 0x38]
0x13ff4:    lea     r12, [rbx + rbx*4]
0x13ff8:    shl     r12, 4
0x13ffc:    add     rax, r12

```

```

0x13fff:    cmp     byte ptr [rax + 0x48], 0
0x14003:    je      0x14048
0x14005:    cmp     dword ptr [rip + 0x17618], 1
0x1400c:    mov     rsi, qword ptr [rax]
0x1400f:    mov     rdi, r13
0x14012:    sbb     edx, edx
0x14014:    and     edx, 4
0x14017:    add     edx, 1
0x1401a:    call    0x4a40
0x1401f:    test    eax, eax
0x14021:    jne     0x14048
0x14023:    add     r12, qword ptr [rbp + 0x38]
0x14027:    movzx   eax, byte ptr [r12 + 0x49]
0x1402d:    cmp     r15d, 0x32
0x14031:    je      0x149b3
0x14037:    test    al, al
0x14039:    je      0x14048
0x1403b:    mov     byte ptr [r12 + 0x49], 0
0x14041:    sub     dword ptr [rbp + 0x70], 1
0x14045:    nop     dword ptr [rax]
0x14048:    add     rbx, 1
0x1404c:    cmp     dword ptr [rbp + 0x68], ebx
0x1404f:    jg      0x13ff0
0x14051:    jmp     0x14132
0x14132:
0x14056:    nop     word ptr cs:[rax + rax]
0x14060:    xor     eax, eax
0x14062:    mov     rdi, rcx
0x14065:    call    0x13ca0
0x1406a:    cmp     qword ptr [rax], 0
0x1406e:    mov     r14, rax
0x14071:    mov     r13, rax
0x14074:    je      0x1412a
0x1407a:    nop     word ptr [rax + rax]
0x14080:    mov     edx, dword ptr [rbp + 0x68]
0x14083:    xor     ebx, ebx
0x14085:    test    edx, edx
0x14087:    jle     0x140f9
0x14089:    nop     dword ptr [rax]
0x14090:    mov     rax, qword ptr [rbp + 0x38]
0x14094:    lea     r12, [rbx + rbx*4]
0x14098:    shl     r12, 4
0x1409c:    add     rax, r12
0x1409f:    cmp     byte ptr [rax + 0x48], 0
0x140a3:    je      0x140f0
0x140a5:    cmp     dword ptr [rip + 0x17578], 1
0x140ac:    mov     rsi, qword ptr [rax]
0x140af:    sbb     edx, edx

```

```

0x140b1:      mov     rdi, qword ptr [r13]
0x140b5:      and     edx, 4
0x140b8:      add     edx, 1
0x140bb:      call    0x4a40
0x140c0:      test    eax, eax
0x140c2:      jne     0x140f0
0x140c4:      add     r12, qword ptr [rbp + 0x38]
0x140c8:      movzx   eax, byte ptr [r12 + 0x49]
0x140ce:      cmp     r15d, 0x24
0x140d2:      je      0x14840
0x140d8:      test    al, al
0x140da:      je      0x140f0
0x140dc:      mov     byte ptr [r12 + 0x49], 0
0x140e2:      sub     dword ptr [rbp + 0x70], 1
0x140e6:      nop     word ptr cs:[rax + rax]
0x140f0:      add     rbx, 1
0x140f4:      cmp     dword ptr [rbp + 0x68], ebx
0x140f7:      jg      0x14090
0x140f9:      add     r13, 8
0x140fd:      cmp     qword ptr [r13], 0
0x14102:      jne     0x14080
0x14108:      mov     rdi, qword ptr [r14]
0x1410b:      test    rdi, rdi
0x1410e:      je      0x1412a
0x14110:      mov     rbx, r14
0x14113:      nop     dword ptr [rax + rax]
0x14118:      call    0x18d10
0x1411d:      mov     rdi, qword ptr [rbx + 8]
0x14121:      add     rbx, 8
0x14125:      test    rdi, rdi
0x14128:      jne     0x14118
0x1412a:      mov     rdi, r14
0x1412d:      call    0x18d10
0x14132:      cmp     byte ptr [rbp + 0x6d], 0
0x14136:      je      0x13f50
0x1413c:      mov     rdi, rbp
0x1413f:      xor     eax, eax
0x14141:      call    0x10d70
0x14146:      cmp     byte ptr [rbp + 0x6d], 0
0x1414a:      je      0x13f50
0x14150:      mov     rdi, rbp
0x14153:      xor     eax, eax
0x14155:      call    0xec50
0x1415a:      jmp     0x13f50
0x13f50:
0x1415f:      mov     rax, qword ptr [rdi + 0x18]
0x14163:      cmp     byte ptr [rax], 0x2f
0x14166:      je      0x149a4

```

```

0x1416c:    mov     esi, dword ptr [rbp + 0x28]
0x1416f:    test    esi, esi
0x14171:    je      0x14187
0x14173:    mov     dword ptr [rbp + 0x28], 0
0x1417a:    mov     rdi, rbp
0x1417d:    xor     eax, eax
0x1417f:    call    0xf110
0x14184:    mov     dword ptr [rbp + 0x2c], eax
0x14187:    mov     rax, qword ptr [rsp + 0x38]
0x1418c:    sub     rax, qword ptr fs:[0x28]
0x14195:    jne     0x14ce9
0x1419b:    add     rsp, 0x48
0x1419f:    mov     rsi, r12
0x141a2:    mov     rdi, rbp
0x141a5:    xor     eax, eax
0x141a7:    pop     rbx
0x141a8:    pop     rbp
0x141a9:    pop     r12
0x141ab:    pop     r13
0x141ad:    pop     r14
0x141af:    pop     r15
0x141b1:    jmp     0x162b0
0x162b0:
0x141b6:    mov     edx, dword ptr [rdi + 0x68]
0x141b9:    test    edx, edx
0x141bb:    jle     0x141ec
0x141bd:    sub     edx, 1
0x141c0:    mov     rcx, qword ptr [rdi + 0x38]
0x141c4:    lea     rdx, [rdx + rdx*4]
0x141c8:    shl     rdx, 4
0x141cc:    lea     rax, [rcx + 0x49]
0x141d0:    lea     rdx, [rcx + rdx + 0x99]
0x141d8:    nop     dword ptr [rax + rax]
0x141e0:    mov     byte ptr [rax], 0
0x141e3:    add     rax, 0x50
0x141e7:    cmp     rdx, rax
0x141ea:    jne     0x141e0
0x141ec:    cmp     byte ptr [rbp + 0x6d], 0
0x141f0:    mov     dword ptr [rbp + 0x70], 0
0x141f7:    je      0x13f50
0x141fd:    jmp     0x1413c
0x1413c:
0x14202:    movzx   edx, byte ptr [rdi + 0x6d]
0x14206:    mov     ebx, r8d
0x14209:    xor     r13d, r13d
0x1420c:    xor     r12d, r12d
0x1420f:    test    r8d, r8d
0x14212:    jne     0x14249

```



```

0x14214:      jmp      0x1488f
0x1488f
0x14219:      nop      dword ptr [rax]
0x14220:      test     r12d, r12d
0x14223:      jne      0x14240
0x14225:      mov      r12d, 1
0x1422b:      test     dl, dl
0x1422d:      je       0x14240
0x1422f:      mov      rdi, rbp
0x14232:      call     0x10760
0x14237:      movzx     edx, byte ptr [rbp + 0x6d]
0x1423b:      nop      dword ptr [rax + rax]
0x14240:      sub      ebx, 1
0x14243:      je       0x14870
0x14249:      mov      esi, dword ptr [rbp + 0x28]
0x1424c:      test     esi, esi
0x1424e:      je       0x14870
0x14254:      lea      eax, [rsi - 1]
0x14257:      mov      dword ptr [rbp + 0x28], eax
0x1425a:      cmp      esi, dword ptr [rbp + 0x2c]
0x1425d:      jne      0x14220
0x1425f:      sub      esi, dword ptr [rbp + 0x88]
0x14265:      mov      eax, 0
0x1426a:      mov      r13d, 1
0x14270:      cmovs     esi, eax
0x14273:      mov      dword ptr [rbp + 0x2c], esi
0x14276:      jmp      0x14240
0x14240
0x14278:      mov      rsi, r12
0x1427b:      xor      eax, eax
0x1427d:      call     0x15bb0
0x14282:      jmp      0x13f50
0x13f50
0x14287:      mov      rsi, r12
0x1428a:      xor      eax, eax
0x1428c:      call     0x15910
0x14291:      jmp      0x13f50
0x13f50
0x14296:      mov      edx, dword ptr [rcx]
0x14298:      mov      rsi, r12
0x1429b:      xor      eax, eax
0x1429d:      call     0x167a0
0x142a2:      jmp      0x13f50
0x13f50
0x142a7:      mov      edx, dword ptr [rcx]
0x142a9:      mov      rsi, r12
0x142ac:      xor      eax, eax
0x142ae:      call     0x12fd0

```

```

0x142b3:      jmp      0x13f50
0x13f50
0x142b8:      mov     rdi, qword ptr [rdi + 0x48]
0x142bc:      call    0x19120
0x142c1:      mov     qword ptr [rbp + 0x48], 0
0x142c9:      jmp     0x13f50
0x13f50
0x142ce:      mov     eax, dword ptr [rdi + 0x30]
0x142d1:      mov     edx, dword ptr [rdi + 0x64]
0x142d4:      mov     esi, 4
0x142d9:      add     r8d, dword ptr [rdi + 0x74]
0x142dd:      add     eax, 1
0x142e0:      cmp     edx, 0x27
0x142e3:      jle     0x14300
0x142e5:      movsxd  rcx, dword ptr [rdi + 0x80]
0x142ec:      cmp     edx, 0x49
0x142ef:      jg      0x142f6
0x142f1:      cmp     ecx, 5
0x142f4:      je      0x14300
0x142f6:      lea     rsi, [rip + 0xd6e3]
0x142fd:      mov     esi, dword ptr [rsi + rcx*4]
0x14300:      sub     edx, esi
0x14302:      sub     eax, edx
0x14304:      mov     edx, 0
0x14309:      cmovs   eax, edx
0x1430c:      cmp     eax, r8d
0x1430f:      cmovg   eax, r8d
0x14313:      mov     dword ptr [rbp + 0x74], eax
0x14316:      mov     rdi, rbp
0x14319:      xor     eax, eax
0x1431b:      call    0x10ec0
0x14320:      xor     eax, eax
0x14322:      mov     rdi, rbp
0x14325:      call    0x10160
0x1432a:      cmp     byte ptr [rbp + 0x6d], 0
0x1432e:      je      0x1433a
0x14330:      mov     rdi, rbp
0x14333:      xor     eax, eax
0x14335:      call    0xec50
0x1433a:      mov     rdi, rbp
0x1433d:      xor     eax, eax
0x1433f:      call    0x10240
0x14344:      jmp     0x13f40
0x13f40
0x14349:      sub     dword ptr [rdi + 0x74], r8d
0x1434d:      mov     eax, dword ptr [rdi + 0x74]
0x14350:      xor     edx, edx
0x14352:      test    eax, eax

```

```

0x14354:      cmovs    eax, edx
0x14357:      jmp     0x14313
0x14313
0x14359:      xor     esi, esi
0x1435b:      mov     rdi, rcx
0x1435e:      mov     edx, 0xa
0x14363:      call    0x4ba0
0x14368:      mov     rsi, r12
0x1436b:      mov     rdi, rbp
0x1436e:      movsxd  rdx, eax
0x14371:      xor     eax, eax
0x14373:      call    0x137c0
0x14378:      jmp     0x13f50
0x13f50
0x1437d:      mov     rsi, r12
0x14380:      xor     eax, eax
0x14382:      call    0x12ca0
0x14387:      jmp     0x13f50
0x13f50
0x1438c:      mov     eax, dword ptr [rdi + 0x60]
0x1438f:      mov     edx, dword ptr [r12 + 0x60]
0x14394:      mov     dword ptr [rdi + 0x60], edx
0x14397:      mov     edx, dword ptr [r12 + 0x64]
0x1439c:      mov     dword ptr [r12 + 0x60], eax
0x143a1:      mov     eax, dword ptr [rdi + 0x64]
0x143a4:      mov     dword ptr [rdi + 0x64], edx
0x143a7:      mov     edx, dword ptr [r12 + 0x58]
0x143ac:      mov     dword ptr [r12 + 0x64], eax
0x143b1:      mov     eax, dword ptr [rdi + 0x58]
0x143b4:      mov     dword ptr [rdi + 0x58], edx
0x143b7:      mov     edx, dword ptr [r12 + 0x5c]
0x143bc:      mov     dword ptr [r12 + 0x58], eax
0x143c1:      mov     eax, dword ptr [rdi + 0x5c]
0x143c4:      mov     dword ptr [rdi + 0x5c], edx
0x143c7:      mov     dword ptr [r12 + 0x5c], eax
0x143cc:      movsxd  rcx, dword ptr [rdi + 0x60]
0x143d0:      movsxd  rdx, dword ptr [rdi + 0x5c]
0x143d4:      movsxd  rsi, dword ptr [rdi + 0x58]
0x143d8:      movsxd  r8, dword ptr [rdi + 0x64]
0x143dc:      call    0xf590
0x143e1:      movsxd  rcx, dword ptr [r12 + 0x60]
0x143e6:      movsxd  rdx, dword ptr [r12 + 0x5c]
0x143eb:      mov     rdi, r12
0x143ee:      movsxd  rsi, dword ptr [r12 + 0x58]
0x143f3:      movsxd  r8, dword ptr [r12 + 0x64]
0x143f8:      call    0xf590
0x143fd:      jmp     0x13f50
0x13f50

```

```

0x14402:      mov     rsi, r12
0x14405:      xor     eax, eax
0x14407:      call    0x14e20
0x1440c:      jmp     0x13f50
0x13f50
0x14411:      mov     eax, dword ptr [rdi + 0x84]
0x14417:      add     eax, 1
0x1441a:      movsxd   r15, eax
0x1441d:      cdq
0x1441e:      imul     r15, r15, 0x38e38e39
0x14425:      sar     r15, 0x21
0x14429:      sub     r15d, edx
0x1442c:      lea     edx, [r15 + r15*8]
0x14430:      sub     eax, edx
0x14432:      mov     r15d, eax
0x14435:      jmp     0x13e84
0x13e84
0x1443a:      mov     eax, dword ptr [rdi + 0x80]
0x14440:      add     eax, 1
0x14443:      movsxd   r15, eax
0x14446:      cdq
0x14447:      imul     r15, r15, 0x2aaaaaab
0x1444e:      shr     r15, 0x20
0x14452:      sub     r15d, edx
0x14455:      lea     edx, [r15 + r15*2]
0x14459:      add     edx, edx
0x1445b:      sub     eax, edx
0x1445d:      mov     r15d, eax
0x14460:      jmp     0x13f1c
0x13f1c
0x14465:      mov     edx, dword ptr [rdi + 0x68]
0x14468:      mov     dword ptr [rdi + 0x70], 0
0x1446f:      test    edx, edx
0x14471:      jle     0x14132
0x14477:      sub     edx, 1
0x1447a:      mov     rcx, qword ptr [rdi + 0x38]
0x1447e:      lea     rdx, [rdx + rdx*4]
0x14482:      shl     rdx, 4
0x14486:      lea     rax, [rcx + 0x49]
0x1448a:      lea     rcx, [rcx + rdx + 0x99]
0x14492:      nop     word ptr [rax + rax]
0x14498:      cmp     byte ptr [rax - 1], 0
0x1449c:      je      0x144ac
0x1449e:      cmp     byte ptr [rax], 0
0x144a1:      sete    dl
0x144a4:      mov     byte ptr [rax], dl
0x144a6:      movzx   edx, dl
0x144a9:      add     dword ptr [rbp + 0x70], edx

```

```

0x144ac:      add     rax, 0x50
0x144b0:      cmp     rcx, rax
0x144b3:      jne     0x14498
0x144b5:      jmp     0x14132
0x14132
0x144ba:      mov     rsi, rcx
0x144bd:      xor     eax, eax
0x144bf:      call    0x161d0
0x144c4:      jmp     0x13f50
0x13f50
0x144c9:      mov     rdx, rcx
0x144cc:      mov     rsi, r12
0x144cf:      xor     eax, eax
0x144d1:      call    0x165f0
0x144d6:      jmp     0x13f50
0x13f50
0x144db:      mov     rcx, qword ptr [rcx + 8]
0x144df:      mov     rdi, rcx
0x144e2:      mov     qword ptr [rsp + 8], rcx
0x144e7:      call    0x49b0
0x144ec:      mov     rcx, qword ptr [rsp + 8]
0x144f1:      mov     qword ptr [rsp + 0x10], rax
0x144f6:      mov     r14, rax
0x144f9:      mov     eax, dword ptr [r13]
0x144fd:      cmp     eax, 2
0x14500:      je      0x14bb4
0x14506:      cmp     eax, 3
0x14509:      je      0x14aea
0x1450f:      cmp     eax, 1
0x14512:      je      0x148db
0x14518:      mov     rax, qword ptr [rbp + 0x40]
0x1451c:      mov     qword ptr [r13 + 0x10], rax
0x14520:      jmp     0x13f50
0x13f50
0x14525:      mov     rdi, qword ptr [rcx + 8]
0x14529:      mov     qword ptr [rsp + 8], rdi
0x1452e:      call    0x49b0
0x14533:      mov     rdi, qword ptr [rsp + 8]
0x14538:      mov     qword ptr [rsp + 0x10], rax
0x1453d:      mov     r14, rax
0x14540:      mov     eax, dword ptr [r13]
0x14544:      cmp     eax, 2
0x14547:      je      0x14a96
0x1454d:      cmp     eax, 3
0x14550:      je      0x149de
0x14556:      cmp     eax, 1
0x14559:      je      0x148db
0x1455f:      mov     rax, qword ptr [rbp + 0x40]

```

```

0x14563:    mov     qword ptr [rsp + 0x10], rax
0x14568:    mov     rax, qword ptr [rsp + 0x10]
0x1456d:    mov     qword ptr [r13 + 0x10], rax
0x14571:    jmp     0x13f50
0x13f50
0x14576:    mov     edi, 0x10
0x1457b:    call    0x19100
0x14580:    lea     rsi, [rsp + 0x20]
0x14585:    mov     qword ptr [rsp + 0x28], 0
0x1458e:    mov     qword ptr [rbp + 0x48], rax
0x14592:    mov     rdi, rax
0x14595:    mov     eax, dword ptr [rbp + 0x28]
0x14598:    mov     dword ptr [rsp + 0x20], eax
0x1459c:    call    0x19150
0x145a1:    mov     qword ptr [rbp + 0x40], 0
0x145a9:    mov     byte ptr [rbp + 0x6e], 0
0x145ad:    jmp     0x13f50
0x13f50
0x145b2:    xor     esi, esi
0x145b4:    mov     edx, 0xa
0x145b9:    mov     rdi, rcx
0x145bc:    call    0x4ba0
0x145c1:    test    eax, eax
0x145c3:    jle     0x13f50
0x145c9:    mov     ecx, dword ptr [rbp + 0x60]
0x145cc:    lea     edx, [rcx - 1]
0x145cf:    cmp     edx, eax
0x145d1:    jle     0x13f50
0x145d7:    mov     dword ptr [r12 + 0x88], eax
0x145df:    mov     dword ptr [rbp + 0x88], eax
0x145e5:    jmp     0x13f50
0x13f50
0x145ea:    mov     eax, dword ptr [rdi + 0x68]
0x145ed:    sub     eax, 1
0x145f0:    cmp     dword ptr [rdi + 0x28], eax
0x145f3:    je      0x13f50
0x145f9:    mov     dword ptr [rdi + 0x28], eax
0x145fc:    mov     rdi, rbp
0x145ff:    xor     eax, eax
0x14601:    call    0xf110
0x14606:    mov     dword ptr [rbp + 0x2c], eax
0x14609:    jmp     0x14132
0x14132
0x1460e:    mov     ecx, dword ptr [rdi + 0x28]
0x14611:    test    ecx, ecx
0x14613:    je      0x13f50
0x14619:    mov     dword ptr [rdi + 0x28], 0
0x14620:    jmp     0x145fc

```

```

0x145fc
0x14622:    mov     r9d, dword ptr [rdi + 0x68]
0x14626:    mov     r10d, dword ptr [rdi + 0x28]
0x1462a:    lea     eax, [r9 - 1]
0x1462e:    cmp     r10d, eax
0x14631:    je      0x13f50
0x14637:    mov     r8d, dword ptr [rdi + 0x60]
0x1463b:    mov     ecx, eax
0x1463d:    sub     ecx, dword ptr [rdi + 0x2c]
0x14640:    lea     edx, [r8 - 2]
0x14644:    cmp     ecx, edx
0x14646:    jl      0x14656
0x14648:    mov     ecx, eax
0x1464a:    sub     ecx, r10d
0x1464d:    cmp     edx, ecx
0x1464f:    jg      0x14656
0x14651:    lea     eax, [r10 + r8 - 2]
0x14656:    mov     dword ptr [rbp + 0x28], eax
0x14659:    mov     rdi, rbp
0x1465c:    xor     eax, eax
0x1465e:    sub     r8d, 1
0x14662:    call    0xf110
0x14667:    mov     dword ptr [rbp + 0x2c], eax
0x1466a:    movzx   eax, byte ptr [rbp + 0x6d]
0x1466e:    cmp     r9d, r8d
0x14671:    jge     0x148ce
0x14677:    test    al, al
0x14679:    je      0x13f50
0x1467f:    mov     esi, r10d
0x14682:    mov     rdi, rbp
0x14685:    call    0x10760
0x1468a:    cmp     byte ptr [rbp + 0x6d], 0
0x1468e:    mov     esi, dword ptr [rbp + 0x28]
0x14691:    je      0x13f50
0x14697:    mov     rdi, rbp
0x1469a:    call    0x10760
0x1469f:    jmp     0x14146
0x14146
0x146a4:    mov     r8d, dword ptr [rdi + 0x28]
0x146a8:    test    r8d, r8d
0x146ab:    je      0x13f50
0x146b1:    mov     r9d, dword ptr [rdi + 0x60]
0x146b5:    lea     eax, [r9 - 2]
0x146b9:    cmp     r8d, eax
0x146bc:    jge     0x149ca
0x146c2:    mov     dword ptr [rdi + 0x28], 0
0x146c9:    xor     eax, eax
0x146cb:    call    0xf110

```

```

0x146d0:    mov     dword ptr [rbp + 0x2c], eax
0x146d3:    sub     r9d, 1
0x146d7:    movzx   eax, byte ptr [rbp + 0x6d]
0x146db:    cmp     r9d, dword ptr [rbp + 0x68]
0x146df:    jle     0x148ce
0x146e5:    mov     esi, r8d
0x146e8:    test    al, al
0x146ea:    je      0x13f50
0x146f0:    jmp     0x14682
0x14682:
0x146f2:    movzx   edi, byte ptr [rdi + 0x6d]
0x146f6:    mov     ebx, r8d
0x146f9:    xor     r13d, r13d
0x146fc:    xor     r12d, r12d
0x146ff:    test    r8d, r8d
0x14702:    jne     0x14739
0x14704:    jmp     0x148c0
0x148c0:
0x14709:    nop     dword ptr [rax]
0x14710:    test    r12d, r12d
0x14713:    jne     0x14730
0x14715:    mov     r12d, 1
0x1471b:    test    dil, dil
0x1471e:    je      0x14730
0x14720:    mov     rdi, rbp
0x14723:    mov     esi, r9d
0x14726:    call    0x10760
0x1472b:    movzx   edi, byte ptr [rbp + 0x6d]
0x1472f:    nop
0x14730:    sub     ebx, 1
0x14733:    je      0x148a0
0x14739:    mov     eax, dword ptr [rbp + 0x68]
0x1473c:    mov     r9d, dword ptr [rbp + 0x28]
0x14740:    sub     eax, 1
0x14743:    cmp     r9d, eax
0x14746:    jge     0x148a0
0x1474c:    mov     esi, dword ptr [rbp + 0x2c]
0x1474f:    mov     ecx, dword ptr [rbp + 0x60]
0x14752:    lea     edx, [r9 + 1]
0x14756:    mov     dword ptr [rbp + 0x28], edx
0x14759:    sub     ecx, 2
0x1475c:    sub     edx, esi
0x1475e:    cmp     edx, ecx
0x14760:    jl      0x14710
0x14762:    sub     eax, ecx
0x14764:    add     esi, dword ptr [rbp + 0x88]
0x1476a:    mov     r13d, 1
0x14770:    add     eax, 1

```



```

0x14773:      cmp     esi, eax
0x14775:      cmovg   esi, eax
0x14778:      mov     dword ptr [rbp + 0x2c], esi
0x1477b:      jmp     0x14730
0x14730
0x1477d:      mov     rax, qword ptr [rdi + 0x18]
0x14781:      mov     esi, dword ptr [rdi + 0x28]
0x14784:      cmp     byte ptr [rax], 0x2f
0x14787:      je      0x1497c
0x1478d:      test    esi, esi
0x1478f:      je      0x147d0
0x14791:      movsxd   rax, esi
0x14794:      lea     rax, [rax + rax*4]
0x14798:      shl     rax, 4
0x1479c:      add     rax, qword ptr [rbp + 0x38]
0x147a0:      movzx   edx, byte ptr [rax + 0x49]
0x147a4:      test    dl, dl
0x147a6:      sete    byte ptr [rax + 0x49]
0x147aa:      cmp     dl, 1
0x147ad:      sbb     eax, eax
0x147af:      and     eax, 2
0x147b2:      sub     eax, 1
0x147b5:      add     dword ptr [rbp + 0x70], eax
0x147b8:      cmp     byte ptr [rbp + 0x6d], 0
0x147bc:      je      0x147d0
0x147be:      mov     rdi, rbp
0x147c1:      mov     dword ptr [rsp + 8], r8d
0x147c6:      call    0x10760
0x147cb:      mov     r8d, dword ptr [rsp + 8]
0x147d0:      xor     ecx, ecx
0x147d2:      mov     rdx, r12
0x147d5:      mov     esi, 0x1c
0x147da:      mov     rdi, rbp
0x147dd:      call    0x13e30
0x147e2:      jmp     0x13f50
0x13f50
0x147e7:      mov     rsi, r12
0x147ea:      xor     eax, eax
0x147ec:      call    0x15400
0x147f1:      jmp     0x13f50
0x13f50
0x147f6:      mov     edx, dword ptr [rdi + 0x68]
0x147f9:      test    edx, edx
0x147fb:      jle     0x14132
0x14801:      sub     edx, 1
0x14804:      mov     rcx, qword ptr [rdi + 0x38]
0x14808:      lea     rdx, [rdx + rdx*4]
0x1480c:      shl     rdx, 4

```

```

0x14810:    lea     rax, [rcx + 0x49]
0x14814:    lea     rdx, [rcx + rdx + 0x99]
0x1481c:    nop     dword ptr [rax]
0x14820:    cmp     byte ptr [rax], 0
0x14823:    jne     0x14832
0x14825:    cmp     byte ptr [rax - 1], 0
0x14829:    je      0x14832
0x1482b:    mov     byte ptr [rax], 1
0x1482e:    add     dword ptr [rbp + 0x70], 1
0x14832:    add     rax, 0x50
0x14836:    cmp     rax, rdx
0x14839:    jne     0x14820
0x1483b:    jmp     0x14132
0x14132:
0x14840:    test    al, al
0x14842:    jne     0x140f0
0x14848:    mov     byte ptr [r12 + 0x49], 1
0x1484e:    add     dword ptr [rbp + 0x70], 1
0x14852:    jmp     0x140f0
0x140f0:
0x14857:    lea     rdi, [rip + 0xcb3a]
0x1485e:    call    0x8b00
0x14863:    jmp     0x13f50
0x13f50:
0x14868:    nop     dword ptr [rax + rax]
0x14870:    test    r13d, r13d
0x14873:    je      0x14931
0x14879:    test    dl, dl
0x1487b:    je      0x13f50
0x14881:    mov     rdi, rbp
0x14884:    xor     eax, eax
0x14886:    call    0x10d70
0x1488b:    movzx   edx, byte ptr [rbp + 0x6d]
0x1488f:    test    dl, dl
0x14891:    jne     0x14150
0x14897:    jmp     0x13f50
0x13f50:
0x1489c:    nop     dword ptr [rax]
0x148a0:    test    r13d, r13d
0x148a3:    je      0x14956
0x148a9:    test    dil, dil
0x148ac:    je      0x13f50
0x148b2:    mov     rdi, rbp
0x148b5:    xor     eax, eax
0x148b7:    call    0x10d70
0x148bc:    movzx   edi, byte ptr [rbp + 0x6d]
0x148c0:    test    dil, dil
0x148c3:    jne     0x14150

```

```

0x148c9:      jmp      0x13f50
0x13f50
0x148ce:      test     al, al
0x148d0:      je       0x13f50
0x148d6:      jmp      0x1413c
0x1413c
0x148db:      mov      rdi, qword ptr [rbp + 0x48]
0x148df:      lea      r14, [rsp + 0x20]
0x148e4:      mov      rsi, r14
0x148e7:      call     0x191a0
0x148ec:      mov      edx, dword ptr [rsp + 0x20]
0x148f0:      mov      rax, qword ptr [rbp + 0x40]
0x148f4:      cmp      rax, qword ptr [rsp + 0x10]
0x148f9:      jne      0x1451c
0x148ff:      mov      r8d, dword ptr [rbp + 0x28]
0x14903:      cmp      r8d, edx
0x14906:      jl       0x14cb2
0x1490c:      jg       0x14ccf
0x14912:      mov      rdi, qword ptr [rbp + 0x48]
0x14916:      mov      edx, 1
0x1491b:      mov      rsi, r14
0x1491e:      call     0x19200
0x14923:      mov      rax, qword ptr [rsp + 0x28]
0x14928:      mov      qword ptr [rbp + 0x40], rax
0x1492c:      jmp      0x1451c
0x1451c
0x14931:      test     r12d, r12d
0x14934:      je       0x1488f
0x1493a:      mov      esi, dword ptr [rbp + 0x28]
0x1493d:      test     dl, dl
0x1493f:      je       0x13f50
0x14945:      mov      rdi, rbp
0x14948:      call     0x10760
0x1494d:      movzx    edx, byte ptr [rbp + 0x6d]
0x14951:      jmp      0x1488f
0x1488f
0x14956:      test     r12d, r12d
0x14959:      je       0x148c0
0x1495f:      mov      esi, dword ptr [rbp + 0x28]
0x14962:      test     dil, dil
0x14965:      je       0x13f50
0x1496b:      mov      rdi, rbp
0x1496e:      call     0x10760
0x14973:      movzx    edi, byte ptr [rbp + 0x6d]
0x14977:      jmp      0x148c0
0x148c0
0x1497c:      cmp      byte ptr [rax + 1], 0
0x14980:      je       0x14791

```

```

0x14986:      jmp      0x1478d
0x1478d
0x1498b:      movsxd  rsi, eax
0x1498e:      lea      rcx, [rip - 0x6315]
0x14995:      mov      edx, 0x50
0x1499a:      call     0x48e0
0x1499f:      jmp      0x13ed7
0x13ed7
0x149a4:      cmp      byte ptr [rax + 1], 0
0x149a8:      je       0x13f50
0x149ae:      jmp      0x1416c
0x1416c
0x149b3:      test     al, al
0x149b5:      jne      0x14048
0x149bb:      mov      byte ptr [r12 + 0x49], 1
0x149c1:      add      dword ptr [rbp + 0x70], 1
0x149c5:      jmp      0x14048
0x14048
0x149ca:      mov      edx, r8d
0x149cd:      sub      edx, eax
0x149cf:      xor      eax, eax
0x149d1:      mov      dword ptr [rdi + 0x28], edx
0x149d4:      call     0xf110
0x149d9:      jmp      0x146d0
0x146d0
0x149de:      mov      eax, dword ptr [rbp + 0x28]
0x149e1:      mov      rdi, qword ptr [rbp + 0x48]
0x149e5:      lea      rsi, [rsp + 0x20]
0x149ea:      mov      qword ptr [rsp + 0x28], r14
0x149ef:      mov      dword ptr [rsp + 0x20], eax
0x149f3:      call     0x19150
0x149f8:      mov      rax, qword ptr [r13 + 8]
0x149fc:      mov      ebx, dword ptr [rbp + 0x28]
0x149ff:      mov      qword ptr [rsp + 8], rax
0x14a04:      test     ebx, ebx
0x14a06:      js       0x14a80
0x14a08:      movsxd  rax, ebx
0x14a0b:      lea      r8, [rax + rax*4]
0x14a0f:      shl      r8, 4
0x14a13:      add      r8, qword ptr [rbp + 0x38]
0x14a17:      mov      r15, r8
0x14a1a:      jmp      0x14a2c
0x14a2c
0x14a1c:      nop      dword ptr [rax]
0x14a20:      sub      ebx, 1
0x14a23:      sub      r15, 0x50
0x14a27:      cmp      ebx, -1
0x14a2a:      je       0x14a80

```

```

0x14a2c:    mov     rsi, qword ptr [r15]
0x14a2f:    mov     rdi, qword ptr [rsp + 8]
0x14a34:    mov     rdx, r14
0x14a37:    call    0x4c30
0x14a3c:    test    eax, eax
0x14a3e:    jne     0x14a20
0x14a40:    cmp     byte ptr [rbp + 0x6e], 0
0x14a44:    mov     qword ptr [rbp + 0x40], r14
0x14a48:    je      0x14c8d
0x14a4e:    mov     dword ptr [rbp + 0x28], ebx
0x14a51:    xor     eax, eax
0x14a53:    mov     rdi, rbp
0x14a56:    call    0xf110
0x14a5b:    cmp     byte ptr [rbp + 0x6d], 0
0x14a5f:    mov     dword ptr [rbp + 0x2c], eax
0x14a62:    je      0x14a77
0x14a64:    xor     eax, eax
0x14a66:    mov     rdi, rbp
0x14a69:    call    0x10d70
0x14a6e:    mov     rax, qword ptr [rbp + 0x40]
0x14a72:    mov     qword ptr [rsp + 0x10], rax
0x14a77:    mov     byte ptr [rbp + 0x6e], 0
0x14a7b:    jmp     0x14568
0x14568
0x14a80:    mov     rax, qword ptr [rbp + 0x40]
0x14a84:    mov     dword ptr [r13], 0xffffffff
0x14a8c:    mov     qword ptr [rsp + 0x10], rax
0x14a91:    jmp     0x14568
0x14568
0x14a96:    cmp     byte ptr [rbp + 0x6e], 0
0x14a9a:    je      0x14c0e
0x14aa0:    mov     eax, dword ptr [rbp + 0x68]
0x14aa3:    lea     ebx, [rax - 1]
0x14aa6:    test    ebx, ebx
0x14aa8:    js      0x14a80
0x14aaa:    movsxd  rax, ebx
0x14aad:    lea     r15, [rax + rax*4]
0x14ab1:    shl     r15, 4
0x14ab5:    add     r15, qword ptr [rbp + 0x38]
0x14ab9:    jmp     0x14ad1
0x14ad1
0x14abb:    nop     dword ptr [rax + rax]
0x14ac0:    sub     ebx, 1
0x14ac3:    sub     r15, 0x50
0x14ac7:    mov     rdi, qword ptr [rsp + 8]
0x14acc:    cmp     ebx, -1
0x14acf:    je      0x14a80
0x14ad1:    mov     rsi, qword ptr [r15]

```

```

0x14ad4:    mov     rdx, r14
0x14ad7:    mov     qword ptr [rsp + 8], rdi
0x14adc:    call    0x4c30
0x14ae1:    test    eax, eax
0x14ae3:    jne     0x14ac0
0x14ae5:    jmp     0x14a40
0x14a40:
0x14aea:    mov     eax, dword ptr [rbp + 0x28]
0x14aed:    mov     rdi, qword ptr [rbp + 0x48]
0x14af1:    lea     rsi, [rsp + 0x20]
0x14af6:    mov     qword ptr [rsp + 0x28], r14
0x14afb:    mov     dword ptr [rsp + 0x20], eax
0x14aff:    call    0x19150
0x14b04:    mov     rax, qword ptr [r13 + 8]
0x14b08:    mov     ebx, dword ptr [rbp + 0x28]
0x14b0b:    mov     qword ptr [rsp + 8], rax
0x14b10:    mov     eax, dword ptr [rbp + 0x68]
0x14b13:    mov     dword ptr [rsp + 0x1c], eax
0x14b17:    cmp     ebx, eax
0x14b19:    jge     0x14ba3
0x14b1f:    movsxd  rax, ebx
0x14b22:    lea     rcx, [rax + rax*4]
0x14b26:    shl     rcx, 4
0x14b2a:    add     rcx, qword ptr [rbp + 0x38]
0x14b2e:    mov     r15, rcx
0x14b31:    jmp     0x14b45
0x14b45:
0x14b33:    nop     dword ptr [rax + rax]
0x14b38:    add     ebx, 1
0x14b3b:    add     r15, 0x50
0x14b3f:    cmp     ebx, dword ptr [rsp + 0x1c]
0x14b43:    je      0x14ba3
0x14b45:    mov     rsi, qword ptr [r15]
0x14b48:    mov     rdi, qword ptr [rsp + 8]
0x14b4d:    mov     rdx, r14
0x14b50:    call    0x4c30
0x14b55:    test    eax, eax
0x14b57:    jne     0x14b38
0x14b59:    cmp     ebx, -1
0x14b5c:    je      0x14ba3
0x14b5e:    cmp     byte ptr [rbp + 0x6e], 0
0x14b62:    mov     qword ptr [rbp + 0x40], r14
0x14b66:    je      0x14c70
0x14b6c:    mov     dword ptr [rbp + 0x28], ebx
0x14b6f:    xor     eax, eax
0x14b71:    mov     rdi, rbp
0x14b74:    call    0xf110
0x14b79:    cmp     byte ptr [rbp + 0x6d], 0

```

```

0x14b7d:    mov     dword ptr [rbp + 0x2c], eax
0x14b80:    je      0x14b95
0x14b82:    xor     eax, eax
0x14b84:    mov     rdi, rbp
0x14b87:    call    0x10d70
0x14b8c:    mov     rax, qword ptr [rbp + 0x40]
0x14b90:    mov     qword ptr [rsp + 0x10], rax
0x14b95:    mov     byte ptr [rbp + 0x6e], 0
0x14b99:    mov     rax, qword ptr [rsp + 0x10]
0x14b9e:    jmp     0x1451c
0x1451c:
0x14ba3:    mov     dword ptr [r13], 0xffffffff
0x14bab:    mov     rax, qword ptr [rbp + 0x40]
0x14baf:    jmp     0x1451c
0x1451c:
0x14bb4:    xor     ebx, ebx
0x14bb6:    cmp     byte ptr [rbp + 0x6e], 0
0x14bba:    je      0x14c41
0x14bc0:    mov     eax, dword ptr [rbp + 0x68]
0x14bc3:    mov     dword ptr [rsp + 0x1c], eax
0x14bc7:    cmp     ebx, eax
0x14bc9:    jge     0x14ba3
0x14bcb:    movsxd  rax, ebx
0x14bce:    lea     r15, [rax + rax*4]
0x14bd2:    shl     r15, 4
0x14bd6:    add     r15, qword ptr [rbp + 0x38]
0x14bda:    jmp     0x14bf2
0x14bf2:
0x14bdc:    nop     dword ptr [rax]
0x14be0:    add     ebx, 1
0x14be3:    add     r15, 0x50
0x14be7:    cmp     ebx, dword ptr [rsp + 0x1c]
0x14beb:    mov     rcx, qword ptr [rsp + 8]
0x14bf0:    je      0x14ba3
0x14bf2:    mov     rsi, qword ptr [r15]
0x14bf5:    mov     rdx, r14
0x14bf8:    mov     rdi, rcx
0x14bfb:    mov     qword ptr [rsp + 8], rcx
0x14c00:    call    0x4c30
0x14c05:    test    eax, eax
0x14c07:    jne     0x14be0
0x14c09:    jmp     0x14b59
0x14b59:
0x14c0e:    mov     eax, dword ptr [rbp + 0x28]
0x14c11:    mov     rdi, qword ptr [rbp + 0x48]
0x14c15:    lea     rsi, [rsp + 0x20]
0x14c1a:    mov     qword ptr [rsp + 0x28], r14
0x14c1f:    mov     dword ptr [rsp + 0x20], eax

```

```

0x14c23:    call    0x19150
0x14c28:    cmp     byte ptr [rbp + 0x6e], 0
0x14c2c:    jne     0x14cee
0x14c32:    mov     eax, dword ptr [rbp + 0x28]
0x14c35:    mov     rdi, qword ptr [r13 + 8]
0x14c39:    lea     ebx, [rax - 1]
0x14c3c:    jmp     0x14aa6
0x14aa6:
0x14c41:    mov     eax, dword ptr [rbp + 0x28]
0x14c44:    mov     rdi, qword ptr [rbp + 0x48]
0x14c48:    lea     rsi, [rsp + 0x20]
0x14c4d:    mov     qword ptr [rsp + 0x28], r14
0x14c52:    mov     dword ptr [rsp + 0x20], eax
0x14c56:    call    0x19150
0x14c5b:    cmp     byte ptr [rbp + 0x6e], 0
0x14c5f:    jne     0x14c67
0x14c61:    mov     eax, dword ptr [rbp + 0x28]
0x14c64:    lea     ebx, [rax + 1]
0x14c67:    mov     rcx, qword ptr [r13 + 8]
0x14c6b:    jmp     0x14bc0
0x14bc0:
0x14c70:    sub     ebx, dword ptr [rbp + 0x28]
0x14c73:    xor     ecx, ecx
0x14c75:    mov     rdx, r12
0x14c78:    mov     esi, 0x1c
0x14c7d:    mov     r8d, ebx
0x14c80:    mov     rdi, rbp
0x14c83:    call    0x13e30
0x14c88:    jmp     0x14518
0x14518:
0x14c8d:    mov     r8d, dword ptr [rbp + 0x28]
0x14c91:    xor     ecx, ecx
0x14c93:    mov     rdx, r12
0x14c96:    mov     esi, 0x1b
0x14c9b:    mov     rdi, rbp
0x14c9e:    sub     r8d, ebx
0x14ca1:    call    0x13e30
0x14ca6:    jmp     0x1455f
0x1455f:
0x14cab:    xor     eax, eax
0x14cad:    jmp     0x13f00
0x13f00:
0x14cb2:    sub     edx, r8d
0x14cb5:    xor     ecx, ecx
0x14cb7:    mov     esi, 0x1c
0x14cbc:    mov     rdi, rbp
0x14cbf:    mov     r8d, edx
0x14cc2:    mov     rdx, r12

```



```

0x14cc5:      call    0x13e30
0x14cca:      jmp     0x14912
0x14912
0x14ccf:      sub     r8d, edx
0x14cd2:      xor     ecx, ecx
0x14cd4:      mov     rdx, r12
0x14cd7:      mov     esi, 0x1b
0x14cdc:      mov     rdi, rbp
0x14cdf:      call    0x13e30
0x14ce4:      jmp     0x14912
0x14912
0x14ce9:      call    0x49d0
0x14cee:      mov     rdi, qword ptr [r13 + 8]
0x14cf2:      jmp     0x14aa0
0x14aa0
0x14cf7:      nop     word ptr [rax + rax]
0x14d00:      endbr64
0x14d04:      push    rbp
0x14d05:      xor     eax, eax
0x14d07:      mov     rbp, rdi
0x14d0a:      mov     dword ptr [rdi + 0x28], 0
0x14d11:      call    0xf110
0x14d16:      mov     rsi, qword ptr [rbp + 0x18]
0x14d1a:      xor     edx, edx
0x14d1c:      mov     ecx, 0x1e
0x14d21:      mov     dword ptr [rbp + 0x2c], eax
0x14d24:      lea     rdi, [rip + 0xc677]
0x14d2b:      xor     eax, eax
0x14d2d:      call    0xf760
0x14d32:      mov     rax, qword ptr [rbp + 0x18]
0x14d36:      cmp     byte ptr [rax], 0x2f
0x14d39:      jne     0x14d51
0x14d3b:      cmp     byte ptr [rax + 1], 0
0x14d3f:      jne     0x14d51
0x14d41:      lea     rdi, [rip + 0xc672]
0x14d48:      call    0x8b00
0x14d4d:      mov     rax, qword ptr [rbp + 0x18]
0x14d51:      mov     edx, 0x2f
0x14d56:      mov     word ptr [rax], dx
0x14d59:      mov     rdi, qword ptr [rbp + 0x18]
0x14d5d:      mov     qword ptr [rbp + 0x78], 1
0x14d65:      call    0x49c0
0x14d6a:      mov     rdi, rbp
0x14d6d:      mov     r8d, 1
0x14d73:      xor     ecx, ecx
0x14d75:      xor     edx, edx
0x14d77:      mov     esi, 0x22
0x14d7c:      pop     rbp

```

```

0x14d7d:      jmp      0x13e30
0x13e30
0x14d82:      nop      word ptr cs:[rax + rax]
0x14d8d:      nop      dword ptr [rax]
0x14d90:      endbr64
0x14d94:      push     rbp
0x14d95:      cmp      byte ptr [rdi + 0x6d], 0
0x14d99:      mov      rbp, rdi
0x14d9c:      mov      eax, esi
0x14d9e:      mov      byte ptr [rdi + 0x6c], sil
0x14da2:      je        0x14db0
0x14da4:      mov      esi, dword ptr [rdi + 0x28]
0x14da7:      call     0x10760
0x14dac:      movzx     eax, byte ptr [rbp + 0x6c]
0x14db0:      test     al, al
0x14db2:      jne      0x14dc0
0x14db4:      pop      rbp
0x14db5:      ret
0x14db6:      nop      word ptr cs:[rax + rax]
0x14dc0:      mov      rdi, qword ptr [rbp + 0x18]
0x14dc4:      call     0x49c0
0x14dc9:      cmp      eax, -1
0x14dcc:      jne      0x14db4
0x14dce:      xor      eax, eax
0x14dd0:      mov      rdi, rbp
0x14dd3:      call     0x14d00
0x14dd8:      xor      eax, eax
0x14dda:      mov      rdi, rbp
0x14ddd:      call     0x10ec0
0x14de2:      xor      eax, eax
0x14de4:      mov      rdi, rbp
0x14de7:      call     0x10160
0x14dec:      cmp      byte ptr [rbp + 0x6d], 0
0x14df0:      jne      0x14e10
0x14df2:      xor      eax, eax
0x14df4:      mov      rdi, rbp
0x14df7:      call     0x10240
0x14dfc:      cmp      byte ptr [rbp + 0x6d], 0
0x14e00:      je        0x14db4
0x14e02:      mov      rdi, rbp
0x14e05:      xor      eax, eax
0x14e07:      pop      rbp
0x14e08:      jmp      0x10d70
0x10d70
0x14e0d:      nop      dword ptr [rax]
0x14e10:      mov      rdi, rbp
0x14e13:      xor      eax, eax
0x14e15:      call     0xec50

```

```

0x14e1a:      jmp      0x14df2
0x14df2
0x14e1c:      nop      dword ptr [rax]
0x14e20:      endbr64
0x14e24:      push     r15
0x14e26:      mov      r15, rsi
0x14e29:      push     r14
0x14e2b:      push     r13
0x14e2d:      push     r12
0x14e2f:      push     rbp
0x14e30:      mov      rbp, rdi
0x14e33:      push     rbx
0x14e34:      sub      rsp, 0x58
0x14e38:      mov      edx, dword ptr [rdi + 0x70]
0x14e3b:      mov      rax, qword ptr fs:[0x28]
0x14e44:      mov      qword ptr [rsp + 0x48], rax
0x14e49:      xor      eax, eax
0x14e4b:      mov      qword ptr [rsp + 0x38], 0
0x14e54:      mov      qword ptr [rsp + 0x40], 0
0x14e5d:      mov      dword ptr [rdi + 0x90], 1
0x14e67:      test     edx, edx
0x14e69:      jne      0x15138
0x14e6f:      movsxd   rax, dword ptr [rdi + 0x28]
0x14e73:      mov      rdx, rax
0x14e76:      lea      rax, [rax + rax*4]
0x14e7a:      shl      rax, 4
0x14e7e:      add      rax, qword ptr [rdi + 0x38]
0x14e82:      mov      r12, qword ptr [rax]
0x14e85:      test     edx, edx
0x14e87:      jne      0x14ea0
0x14e89:      mov      rax, qword ptr [rdi + 0x18]
0x14e8d:      cmp      byte ptr [rax], 0x2f
0x14e90:      jne      0x15110
0x14e96:      cmp      byte ptr [rax + 1], 0
0x14e9a:      jne      0x15110
0x14ea0:      mov      rdi, r12
0x14ea3:      call     0x49b0
0x14ea8:      lea      rdi, [rax + 0x11]
0x14eac:      call     0x1d670
0x14eb1:      mov      edx, 0xe
0x14eb6:      mov      rsi, r12
0x14eb9:      lea      rdi, [rip + 0x16720]
0x14ec0:      mov      r13, rax
0x14ec3:      call     0x4880
0x14ec8:      lea      rcx, [rip + 0xc500]
0x14ecf:      mov      esi, 1
0x14ed4:      mov      rdi, r13
0x14ed7:      mov      r8, rax

```

```

0x14eda:    mov     rdx, -1
0x14ee1:    xor     eax, eax
0x14ee3:    call    0x4f10
0x14ee8:    mov     rdi, r12
0x14eeb:    call    0x49b0
0x14ef0:    mov     rdi, qword ptr [r15 + 0x18]
0x14ef4:    mov     rbx, rax
0x14ef7:    call    0x49b0
0x14efc:    lea     rdi, [rbx + rax + 2]
0x14f01:    call    0x1d670
0x14f06:    mov     r8, qword ptr [r15 + 0x18]
0x14f0a:    mov     r9, r12
0x14f0d:    lea     rcx, [rip + 0xb7ed]
0x14f14:    mov     rdi, rax
0x14f17:    mov     r14, rax
0x14f1a:    mov     esi, 1
0x14f1f:    xor     eax, eax
0x14f21:    mov     rdx, -1
0x14f28:    call    0x4f10
0x14f2d:    mov     rdi, r13
0x14f30:    lea     rsi, [rsp + 0x40]
0x14f35:    mov     rdx, r14
0x14f38:    mov     rcx, qword ptr [rip + 0x166d9]
0x14f3f:    call    0x83e0
0x14f44:    mov     rdi, r13
0x14f47:    test    rax, rax
0x14f4a:    je      0x153c3
0x14f50:    call    0x18d10
0x14f55:    movsxd  rax, dword ptr [rbp + 0x28]
0x14f59:    lea     rax, [rax + rax*4]
0x14f5d:    shl     rax, 4
0x14f61:    add     rax, qword ptr [rbp + 0x38]
0x14f65:    mov     eax, dword ptr [rax + 0x10]
0x14f68:    and     eax, 0xf000
0x14f6d:    cmp     eax, 0x4000
0x14f72:    je      0x152f0
0x14f78:    mov     rdi, qword ptr [rip + 0x15509]
0x14f7f:    call    0xdb60
0x14f84:    xor     eax, eax
0x14f86:    call    0xa820
0x14f8b:    mov     rdi, qword ptr [rsp + 0x40]
0x14f90:    xor     eax, eax
0x14f92:    call    0x19940
0x14f97:    mov     rdi, qword ptr [rsp + 0x40]
0x14f9c:    mov     r13, rax
0x14f9f:    call    0x18d10
0x14fa4:    mov     rdi, r12
0x14fa7:    xor     eax, eax

```

```

0x14fa9:      mov     rsi, r13
0x14fac:      mov     qword ptr [rsp + 0x40], r13
0x14fb1:      call    0x11190
0x14fb6:      mov     rdi, r14
0x14fb9:      mov     ebx, eax
0x14fbb:      call    0x18d10
0x14fc0:      test    ebx, ebx
0x14fc2:      jne     0x153a0
0x14fc8:      call    0x4bf0
0x14fcd:      mov     ebx, eax
0x14fcf:      call    0x49e0
0x14fd4:      movsxd   rdx, dword ptr [rbp + 0x28]
0x14fd8:      mov     r9d, ebx
0x14fdb:      mov     rsi, r12
0x14fde:      mov     r8d, eax
0x14fe1:      mov     rdi, rbp
0x14fe4:      xor     eax, eax
0x14fe6:      lea     rdx, [rdx + rdx*4]
0x14fea:      shl     rdx, 4
0x14fee:      add     rdx, qword ptr [rbp + 0x38]
0x14ff2:      mov     ecx, dword ptr [rdx + 0x10]
0x14ff5:      mov     rdx, qword ptr [rsp + 0x40]
0x14ffa:      call    0x11330
0x14fff:      mov     rdi, qword ptr [rsp + 0x40]
0x15004:      mov     ebx, eax
0x15006:      call    0x18d10
0x1500b:      lea     eax, [rbx - 1]
0x1500e:      cmp     eax, 1
0x15011:      jbe     0x15039
0x15013:      cdqe
0x15015:      lea     rdx, [rip + 0x15404]
0x1501c:      xor     ecx, ecx
0x1501e:      mov     rsi, r12
0x15021:      mov     rdx, qword ptr [rdx + rax*8]
0x15025:      mov     r8d, 0x1e
0x1502b:      lea     rdi, [rip + 0xc277]
0x15032:      xor     eax, eax
0x15034:      call    0xf7e0
0x15039:      xor     eax, eax
0x1503b:      call    0xe4e0
0x15040:      mov     rdi, rbp
0x15043:      xor     eax, eax
0x15045:      call    0x10240
0x1504a:      xor     eax, eax
0x1504c:      mov     rdi, r15
0x1504f:      call    0x10240
0x15054:      xor     eax, eax
0x15056:      call    0xa820

```

```

0x1505b:    mov     rsi, qword ptr [r15 + 0x18]
0x1505f:    xor     eax, eax
0x15061:    mov     edx, 1
0x15066:    mov     rdi, r15
0x15069:    call    0xf880
0x1506e:    test    eax, eax
0x15070:    je      0x153e0
0x15076:    xor     eax, eax
0x15078:    mov     rdi, r15
0x1507b:    call    0x10ec0
0x15080:    xor     eax, eax
0x15082:    mov     rdi, r15
0x15085:    call    0x10160
0x1508a:    cmp     byte ptr [r15 + 0x6d], 0
0x1508f:    je      0x1509b
0x15091:    mov     rdi, r15
0x15094:    xor     eax, eax
0x15096:    call    0xec50
0x1509b:    xor     eax, eax
0x1509d:    mov     rdi, r15
0x150a0:    call    0x10240
0x150a5:    cmp     byte ptr [r15 + 0x6d], 0
0x150aa:    je      0x150b6
0x150ac:    mov     rdi, r15
0x150af:    xor     eax, eax
0x150b1:    call    0x10d70
0x150b6:    mov     rsi, qword ptr [rbp + 0x18]
0x150ba:    xor     eax, eax
0x150bc:    mov     edx, 1
0x150c1:    mov     rdi, rbp
0x150c4:    call    0xf880
0x150c9:    test    eax, eax
0x150cb:    je      0x153d0
0x150d1:    xor     eax, eax
0x150d3:    mov     rdi, rbp
0x150d6:    call    0x10ec0
0x150db:    xor     eax, eax
0x150dd:    mov     rdi, rbp
0x150e0:    call    0x10160
0x150e5:    cmp     byte ptr [rbp + 0x6d], 0
0x150e9:    je      0x150f5
0x150eb:    mov     rdi, rbp
0x150ee:    xor     eax, eax
0x150f0:    call    0xec50
0x150f5:    xor     eax, eax
0x150f7:    mov     rdi, rbp
0x150fa:    call    0x10240
0x150ff:    cmp     byte ptr [rbp + 0x6d], 0

```

```

0x15103:    je      0x15110
0x15105:    mov     rdi, rbp
0x15108:    xor     eax, eax
0x1510a:    call    0x10d70
0x1510f:    nop
0x15110:    mov     rax, qword ptr [rsp + 0x48]
0x15115:    sub     rax, qword ptr fs:[0x28]
0x1511e:    jne     0x153ed
0x15124:    add     rsp, 0x58
0x15128:    pop     rbx
0x15129:    pop     rbp
0x1512a:    pop     r12
0x1512c:    pop     r13
0x1512e:    pop     r14
0x15130:    pop     r15
0x15132:    ret
0x15133:    nop     dword ptr [rax + rax]
0x15138:    mov     rdx, qword ptr [rsi + 0x18]
0x1513c:    mov     rcx, qword ptr [rip + 0x164d5]
0x15143:    lea     rsi, [rsp + 0x38]
0x15148:    lea     rdi, [rip + 0xc28d]
0x1514f:    call    0x83e0
0x15154:    test    rax, rax
0x15157:    je      0x15110
0x15159:    mov     rdi, qword ptr [rbp + 0x18]
0x1515d:    mov     rsi, qword ptr [rsp + 0x38]
0x15162:    xor     eax, eax
0x15164:    call    0x11190
0x15169:    test    eax, eax
0x1516b:    jne     0x15308
0x15171:    mov     r12, qword ptr [rsp + 0x38]
0x15176:    mov     rdi, r12
0x15179:    call    0x49b0
0x1517e:    mov     rdi, r12
0x15181:    mov     rbx, rax
0x15184:    lea     rax, [rax + 1]
0x15188:    mov     qword ptr [rsp + 0x18], rax
0x1518d:    lea     rax, [rbx + 2]
0x15191:    mov     rsi, rax
0x15194:    mov     qword ptr [rsp + 0x10], rax
0x15199:    call    0x1d690
0x1519e:    mov     qword ptr [rsp + 0x38], rax
0x151a3:    mov     byte ptr [rax + rbx], 0x2f
0x151a7:    mov     rax, qword ptr [rsp + 0x38]
0x151ac:    mov     byte ptr [rax + rbx + 1], 0
0x151b1:    mov     eax, dword ptr [rbp + 0x70]
0x151b4:    mov     dword ptr [rbp + 0x50], 0xffffffff
0x151bb:    mov     dword ptr [rbp + 0x54], eax

```

```

0x151be:    mov     rdi, rbp
0x151c1:    call   0x100e0
0x151c6:    cmp     eax, -1
0x151c9:    je      0x152ce
0x151cf:    cdq     rax
0x151d1:    lea     rax, [rax + rax*4]
0x151d5:    shl     rax, 4
0x151d9:    mov     qword ptr [rsp + 8], rax
0x151de:    add     rax, qword ptr [rbp + 0x38]
0x151e2:    mov     r12, qword ptr [rax]
0x151e5:    mov     r14d, dword ptr [rax + 0x10]
0x151e9:    call   0x49e0
0x151ee:    mov     ebx, eax
0x151f0:    call   0x4bf0
0x151f5:    mov     rdi, r12
0x151f8:    mov     r13d, eax
0x151fb:    call   0x49b0
0x15200:    mov     rcx, qword ptr [rsp + 0x10]
0x15205:    mov     rdi, qword ptr [rsp + 0x38]
0x1520a:    lea     rsi, [rcx + rax]
0x1520e:    call   0x1d690
0x15213:    mov     rcx, qword ptr [rsp + 0x18]
0x15218:    mov     rsi, r12
0x1521b:    mov     qword ptr [rsp + 0x38], rax
0x15220:    lea     rdi, [rax + rcx]
0x15224:    call   0x48b0
0x15229:    mov     eax, dword ptr [rip + 0x16881]
0x1522f:    test    eax, eax
0x15231:    jne     0x15338
0x15237:    mov     rdi, qword ptr [rip + 0x15252]
0x1523e:    call   0xdb60
0x15243:    xor     eax, eax
0x15245:    call   0xa820
0x1524a:    mov     rdx, qword ptr [rsp + 0x38]
0x1524f:    xor     eax, eax
0x15251:    mov     r9d, r13d
0x15254:    mov     r8d, ebx
0x15257:    mov     ecx, r14d
0x1525a:    mov     rsi, r12
0x1525d:    mov     rdi, rbp
0x15260:    call   0x11330
0x15265:    cmp     eax, 1
0x15268:    je      0x15320
0x1526e:    cmp     eax, 2
0x15271:    je      0x152ce
0x15273:    cmp     eax, 3
0x15276:    je      0x151be
0x1527c:    sub     eax, 1

```



```

0x1527f:      lea      rbx, [rip + 0x1519a]
0x15286:      xor      ecx, ecx
0x15288:      mov      rsi, r12
0x1528b:      cdqe
0x1528d:      mov      r8d, 0xe
0x15293:      lea      rdi, [rip + 0xc00f]
0x1529a:      mov      rdx, qword ptr [rbx + rax*8]
0x1529e:      xor      eax, eax
0x152a0:      call     0xf7e0
0x152a5:      test     al, al
0x152a7:      je       0x152ce
0x152a9:      xor      eax, eax
0x152ab:      mov      rdi, rbp
0x152ae:      call     0x10240
0x152b3:      mov      rdi, r15
0x152b6:      xor      eax, eax
0x152b8:      call     0x10240
0x152bd:      mov      rdi, rbp
0x152c0:      call     0x100e0
0x152c5:      cmp      eax, -1
0x152c8:      jne      0x151cf
0x152ce:      mov      rdi, qword ptr [rsp + 0x38]
0x152d3:      test     rdi, rdi
0x152d6:      je       0x152dd
0x152d8:      call     0x18d10
0x152dd:      xor      eax, eax
0x152df:      call     0xe4e0
0x152e4:      xor      eax, eax
0x152e6:      call     0xa820
0x152eb:      jmp      0x1505b
0x1505b
0x152f0:      mov      rdi, qword ptr [rip + 0x151a1]
0x152f7:      call     0xdb60
0x152fc:      jmp      0x14f84
0x14f84
0x15301:      nop      dword ptr [rax]
0x15308:      mov      edx, 0xc
0x1530d:      xor      esi, esi
0x1530f:      lea      rdi, [rip + 0x14fea]
0x15316:      call     0x8070
0x1531b:      jmp      0x15110
0x15110
0x15320:      mov      rax, qword ptr [rbp + 0x38]
0x15324:      mov      rbx, qword ptr [rsp + 8]
0x15329:      mov      byte ptr [rax + rbx + 0x49], 0
0x1532e:      jmp      0x152a9
0x152a9
0x15333:      nop      dword ptr [rax + rax]

```

```

0x15338:    mov     dword ptr [rip + 0x1676e], 0
0x15342:    xor     eax, eax
0x15344:    call    0xc8c0
0x15349:    mov     edx, 4
0x1534e:    lea     rsi, [rip + 0xb1dd]
0x15355:    lea     rdi, [rip + 0xbe9d]
0x1535c:    mov     qword ptr [rsp + 0x28], rax
0x15361:    call    0x8070
0x15366:    mov     rdi, qword ptr [rsp + 0x28]
0x1536b:    mov     byte ptr [rsp + 0x27], al
0x1536f:    call    0xc970
0x15374:    xor     eax, eax
0x15376:    call    0xd870
0x1537b:    xor     eax, eax
0x1537d:    call    0xd740
0x15382:    xor     eax, eax
0x15384:    call    0xa820
0x15389:    movzx   eax, byte ptr [rsp + 0x27]
0x1538e:    and     eax, 0xffffffff
0x15391:    cmp     al, 0x4e
0x15393:    je      0x15237
0x15399:    jmp     0x152ce
0x152ce:
0x1539e:    nop
0x153a0:    mov     rdx, qword ptr [rsp + 0x40]
0x153a5:    lea     rdi, [rip + 0xc8a4]
0x153ac:    xor     ecx, ecx
0x153ae:    xor     eax, eax
0x153b0:    mov     r8d, 0x1e
0x153b6:    mov     rsi, r12
0x153b9:    call    0xf7e0
0x153be:    mov     rdi, qword ptr [rsp + 0x40]
0x153c3:    call    0x18d10
0x153c8:    jmp     0x15110
0x15110:
0x153cd:    nop     dword ptr [rax]
0x153d0:    mov     rdi, rbp
0x153d3:    call    0x14d00
0x153d8:    jmp     0x150d1
0x150d1:
0x153dd:    nop     dword ptr [rax]
0x153e0:    mov     rdi, r15
0x153e3:    call    0x14d00
0x153e8:    jmp     0x15076
0x15076:
0x153ed:    call    0x49d0
0x153f2:    nop     word ptr cs:[rax + rax]
0x153fd:    nop     dword ptr [rax]

```

```

0x15400:      endbr64
0x15404:      push    r15
0x15406:      push    r14
0x15408:      push    r13
0x1540a:      push    r12
0x1540c:      push    rbp
0x1540d:      mov     rbp, rdi
0x15410:      push    rbx
0x15411:      sub     rsp, 0x28
0x15415:      mov     qword ptr [rsp + 0x10], rsi
0x1541a:      mov     esi, dword ptr [rdi + 0x70]
0x1541d:      test    esi, esi
0x1541f:      jne     0x15450
0x15421:      mov     ecx, dword ptr [rdi + 0x28]
0x15424:      test    ecx, ecx
0x15426:      jne     0x15450
0x15428:      mov     rax, qword ptr [rdi + 0x18]
0x1542c:      cmp     byte ptr [rax], 0x2f
0x1542f:      je      0x15440
0x15431:      add     rsp, 0x28
0x15435:      pop     rbx
0x15436:      pop     rbp
0x15437:      pop     r12
0x15439:      pop     r13
0x1543b:      pop     r14
0x1543d:      pop     r15
0x1543f:      ret
0x15440:      cmp     byte ptr [rax + 1], 0
0x15444:      jne     0x15431
0x15446:      nop     word ptr cs:[rax + rax]
0x15450:      xor     edx, edx
0x15452:      lea     rsi, [rip + 0xb0d9]
0x15459:      lea     rdi, [rip + 0xbf97]
0x15460:      call    0x8070
0x15465:      cmp     al, 0x79
0x15467:      jne     0x15431
0x15469:      mov     edx, dword ptr [rbp + 0x68]
0x1546c:      test    edx, edx
0x1546e:      jle     0x158fc
0x15474:      mov     rax, qword ptr [rbp + 0x38]
0x15478:      xor     ecx, ecx
0x1547a:      add     rax, 0x49
0x1547e:      jmp     0x1548b
0x1548b:
0x15480:      add     ecx, 1
0x15483:      add     rax, 0x50
0x15487:      cmp     ecx, edx
0x15489:      je      0x15490

```

```

0x1548b:    cmp     byte ptr [rax], 0
0x1548e:    je      0x15480
0x15490:    mov     dword ptr [rsp + 0x1c], ecx
0x15494:    mov     eax, dword ptr [rbp + 0x70]
0x15497:    mov     dword ptr [rbp + 0x50], 0xffffffff
0x1549e:    xor     r13d, r13d
0x154a1:    lea     r15, [rip + 0xbf69]
0x154a8:    mov     dword ptr [rsp + 0x18], 1
0x154b0:    mov     dword ptr [rbp + 0x54], eax
0x154b3:    nop     dword ptr [rax + rax]
0x154b8:    mov     rdi, rbp
0x154bb:    call    0x100e0
0x154c0:    cmp     eax, -1
0x154c3:    je      0x155c6
0x154c9:    cdqe
0x154cb:    lea     rbx, [rax + rax*4]
0x154cf:    mov     rax, qword ptr [rbp + 0x38]
0x154d3:    shl     rbx, 4
0x154d7:    mov     r12, qword ptr [rax + rbx]
0x154db:    xor     eax, eax
0x154dd:    call    0x18ab0
0x154e2:    mov     rdi, r12
0x154e5:    call    0x49b0
0x154ea:    lea     rdi, [rax + 0x20]
0x154ee:    call    0x1d670
0x154f3:    mov     r8, r12
0x154f6:    mov     rcx, r15
0x154f9:    mov     esi, 1
0x154fe:    mov     r14, rax
0x15501:    mov     rdi, rax
0x15504:    mov     rdx, -1
0x1550b:    xor     eax, eax
0x1550d:    call    0x4f10
0x15512:    mov     edx, 1
0x15517:    mov     esi, 2
0x1551c:    mov     rdi, r14
0x1551f:    call    0xe480
0x15524:    xor     eax, eax
0x15526:    call    0xa820
0x1552b:    mov     rdi, r14
0x1552e:    call    0x18d10
0x15533:    mov     edx, dword ptr [rip + 0x16577]
0x15539:    test    edx, edx
0x1553b:    jne     0x15760
0x15541:    mov     eax, dword ptr [rsp + 0x18]
0x15545:    test    eax, eax
0x15547:    jne     0x15738
0x1554d:    mov     rdi, qword ptr [rip + 0x14f54]

```

```

0x15554:      call    0xdb60
0x15559:      xor     eax, eax
0x1555b:      call    0xa820
0x15560:      cmp     r13d, 0x61
0x15564:      je      0x15728
0x1556a:      cmp     r13d, 0x73
0x1556e:      je      0x154b8
0x15574:      cmp     r13d, 0x79
0x15578:      jne     0x155c6
0x1557a:      mov     rax, qword ptr [rbp + 0x38]
0x1557e:      cmp     byte ptr [rax + rbx + 0x48], 0
0x15583:      jne     0x156e8
0x15589:      mov     rdi, qword ptr [rip + 0x14f10]
0x15590:      call    0xdb60
0x15595:      xor     eax, eax
0x15597:      call    0xa820
0x1559c:      mov     rdi, r12
0x1559f:      call    0x4990
0x155a4:      test    eax, eax
0x155a6:      jne     0x157c8
0x155ac:      mov     rax, qword ptr [rbp + 0x38]
0x155b0:      mov     rdi, rbp
0x155b3:      mov     byte ptr [rax + rbx + 0x49], 0
0x155b8:      call    0x100e0
0x155bd:      cmp     eax, -1
0x155c0:      jne     0x154c9
0x155c6:      mov     eax, dword ptr [rsp + 0x1c]
0x155ca:      cmp     dword ptr [rbp + 0x68], eax
0x155cd:      je      0x155e3
0x155cf:      mov     eax, dword ptr [rsp + 0x1c]
0x155d3:      mov     rdi, rbp
0x155d6:      mov     dword ptr [rbp + 0x28], eax
0x155d9:      xor     eax, eax
0x155db:      call    0xf110
0x155e0:      mov     dword ptr [rbp + 0x2c], eax
0x155e3:      xor     eax, eax
0x155e5:      mov     rdi, rbp
0x155e8:      call    0x10240
0x155ed:      mov     rdi, qword ptr [rsp + 0x10]
0x155f2:      xor     eax, eax
0x155f4:      call    0x10240
0x155f9:      xor     eax, eax
0x155fb:      call    0xe4e0
0x15600:      xor     eax, eax
0x15602:      call    0xa820
0x15607:      mov     rsi, qword ptr [rbp + 0x18]
0x1560b:      xor     eax, eax
0x1560d:      mov     edx, 1

```

```

0x15612:    mov     rdi, rbp
0x15615:    call   0xf880
0x1561a:    test    eax, eax
0x1561c:    je      0x158e0
0x15622:    xor     eax, eax
0x15624:    mov     rdi, rbp
0x15627:    call   0x10ec0
0x1562c:    xor     eax, eax
0x1562e:    mov     rdi, rbp
0x15631:    call   0x10160
0x15636:    cmp     byte ptr [rbp + 0x6d], 0
0x1563a:    jne     0x15890
0x15640:    xor     eax, eax
0x15642:    mov     rdi, rbp
0x15645:    call   0x10240
0x1564a:    cmp     byte ptr [rbp + 0x6d], 0
0x1564e:    jne     0x15880
0x15654:    mov     rax, qword ptr [rsp + 0x10]
0x15659:    mov     rdi, qword ptr [rbp + 0x18]
0x1565d:    mov     r12, qword ptr [rax + 0x18]
0x15661:    mov     rsi, r12
0x15664:    call   0x4b30
0x15669:    test    eax, eax
0x1566b:    jne     0x15431
0x15671:    mov     rdi, qword ptr [rsp + 0x10]
0x15676:    mov     edx, 1
0x1567b:    mov     rsi, r12
0x1567e:    call   0xf880
0x15683:    test    eax, eax
0x15685:    je      0x158ed
0x1568b:    mov     rbx, qword ptr [rsp + 0x10]
0x15690:    xor     eax, eax
0x15692:    mov     rdi, rbx
0x15695:    call   0x10ec0
0x1569a:    xor     eax, eax
0x1569c:    mov     rdi, rbx
0x1569f:    call   0x10160
0x156a4:    cmp     byte ptr [rbx + 0x6d], 0
0x156a8:    je      0x156b4
0x156aa:    mov     rdi, rbx
0x156ad:    xor     eax, eax
0x156af:    call   0xec50
0x156b4:    mov     rbx, qword ptr [rsp + 0x10]
0x156b9:    xor     eax, eax
0x156bb:    mov     rdi, rbx
0x156be:    call   0x10240
0x156c3:    cmp     byte ptr [rbx + 0x6d], 0
0x156c7:    je      0x15431

```

```

0x156cd:      add     rsp, 0x28
0x156d1:      mov     rdi, rbx
0x156d4:      xor     eax, eax
0x156d6:      pop     rbx
0x156d7:      pop     rbp
0x156d8:      pop     r12
0x156da:      pop     r13
0x156dc:      pop     r14
0x156de:      pop     r15
0x156e0:      jmp     0x10d70
0x10d70
0x156e5:      nop     dword ptr [rax]
0x156e8:      mov     rdi, r12
0x156eb:      call    0x4870
0x156f0:      test    eax, eax
0x156f2:      je      0x155ac
0x156f8:      xor     eax, eax
0x156fa:      mov     ecx, 0xe
0x156ff:      mov     rsi, r12
0x15702:      lea     rdx, [rip + 0xae29]
0x15709:      lea     rdi, [rip + 0xc5e8]
0x15710:      call    0xf760
0x15715:      cmp     al, 0x79
0x15717:      je      0x154b8
0x1571d:      jmp     0x155c6
0x155c6
0x15722:      nop     word ptr [rax + rax]
0x15728:      mov     dword ptr [rsp + 0x18], 0
0x15730:      jmp     0x1557a
0x1557a
0x15735:      nop     dword ptr [rax]
0x15738:      mov     ecx, 2
0x1573d:      lea     rdx, [rip + 0xbcd9]
0x15744:      mov     rsi, r12
0x15747:      xor     eax, eax
0x15749:      lea     rdi, [rip + 0xc528]
0x15750:      call    0xf760
0x15755:      movsx   r13d, al
0x15759:      jmp     0x1554d
0x1554d
0x1575e:      nop
0x15760:      mov     dword ptr [rip + 0x16346], 0
0x1576a:      xor     eax, eax
0x1576c:      call    0xc8c0
0x15771:      mov     edx, 4
0x15776:      lea     rsi, [rip + 0xad5]
0x1577d:      lea     rdi, [rip + 0xba75]
0x15784:      mov     qword ptr [rsp + 8], rax

```

```

0x15789:      call    0x8070
0x1578e:      mov     rdi, qword ptr [rsp + 8]
0x15793:      mov     r14d, eax
0x15796:      call    0xc970
0x1579b:      xor     eax, eax
0x1579d:      and     r14d, 0xffffffff
0x157a1:      call    0xd870
0x157a6:      xor     eax, eax
0x157a8:      call    0xd740
0x157ad:      xor     eax, eax
0x157af:      call    0xa820
0x157b4:      cmp     r14b, 0x4e
0x157b8:      je      0x15541
0x157be:      jmp     0x155c6
0x155c6:
0x157c3:      nop     dword ptr [rax + rax]
0x157c8:      xor     eax, eax
0x157ca:      mov     ecx, 0x12
0x157cf:      mov     rsi, r12
0x157d2:      lea     rdx, [rip + 0xad59]
0x157d9:      lea     rdi, [rip + 0xc4c0]
0x157e0:      call    0xf760
0x157e5:      cmp     al, 0x79
0x157e7:      jne     0x156f8
0x157ed:      mov     rdi, r12
0x157f0:      call    0x49b0
0x157f5:      lea     rdi, [rax + 0x21]
0x157f9:      call    0x1d670
0x157fe:      mov     rdx, -1
0x15805:      mov     r8, r12
0x15808:      lea     rcx, [rip + 0xbb38]
0x1580f:      mov     r14, rax
0x15812:      mov     rdi, rax
0x15815:      mov     esi, 1
0x1581a:      xor     eax, eax
0x1581c:      call    0x4f10
0x15821:      mov     esi, 1
0x15826:      mov     rdi, r14
0x15829:      call    0x193f0
0x1582e:      mov     rdi, r14
0x15831:      mov     dword ptr [rsp + 8], eax
0x15835:      call    0x18d10
0x1583a:      mov     rdi, qword ptr [rbp + 0x18]
0x1583e:      call    0xc4b0
0x15843:      mov     eax, dword ptr [rsp + 8]
0x15847:      mov     edx, eax
0x15849:      and     edx, 0x7f
0x1584c:      add     edx, 1

```



```

0x1584f:      cmp     dl, 1
0x15852:      jg      0x158b0
0x15854:      test    al, 0x7f
0x15856:      jne     0x156f8
0x1585c:      test    ah, 0xff
0x1585f:      je      0x155ac
0x15865:      lea     rdi, [rip + 0xb37e]
0x1586c:      call    0x195e0
0x15871:      jmp     0x156f8
0x156f8
0x15876:      nop     word ptr cs:[rax + rax]
0x15880:      mov     rdi, rbp
0x15883:      xor     eax, eax
0x15885:      call    0x10d70
0x1588a:      jmp     0x15654
0x15654
0x1588f:      nop
0x15890:      xor     eax, eax
0x15892:      mov     rdi, rbp
0x15895:      call    0xec50
0x1589a:      xor     eax, eax
0x1589c:      mov     rdi, rbp
0x1589f:      call    0x10240
0x158a4:      cmp     byte ptr [rbp + 0x6d], 0
0x158a8:      je      0x15654
0x158ae:      jmp     0x15880
0x15880
0x158b0:      xor     eax, eax
0x158b2:      mov     ecx, 0xe
0x158b7:      mov     rsi, r12
0x158ba:      lea     rdx, [rip + 0xac71]
0x158c1:      lea     rdi, [rip + 0xc408]
0x158c8:      call    0xf760
0x158cd:      cmp     al, 0x79
0x158cf:      je      0x154b8
0x158d5:      jmp     0x155c6
0x155c6
0x158da:      nop     word ptr [rax + rax]
0x158e0:      mov     rdi, rbp
0x158e3:      call    0x14d00
0x158e8:      jmp     0x15622
0x15622
0x158ed:      mov     rdi, qword ptr [rsp + 0x10]
0x158f2:      call    0x14d00
0x158f7:      jmp     0x1568b
0x1568b
0x158fc:      mov     dword ptr [rsp + 0x1c], 0
0x15904:      jmp     0x15494

```

```

0x15494
0x15909:      nop      dword ptr [rax]
0x15910:      endbr64
0x15914:      push     r13
0x15916:      xor      edx, edx
0x15918:      push     r12
0x1591a:      push     rbp
0x1591b:      mov      rbp, rsi
0x1591e:      push     rbx
0x1591f:      mov      rbx, rdi
0x15922:      lea      rdi, [rip + 0xbaf9]
0x15929:      sub      rsp, 0x18
0x1592d:      mov      rcx, qword ptr [rip + 0x15cd4]
0x15934:      mov      rax, qword ptr fs:[0x28]
0x1593d:      mov      qword ptr [rsp + 8], rax
0x15942:      xor      eax, eax
0x15944:      mov      rsi, rsp
0x15947:      mov      qword ptr [rsp], 0
0x1594f:      call     0x83e0
0x15954:      test     rax, rax
0x15957:      je       0x15a0b
0x1595d:      mov      rdi, qword ptr [rsp]
0x15961:      cmp      byte ptr [rdi], 0
0x15964:      je       0x15a06
0x1596a:      xor      eax, eax
0x1596c:      call     0x19940
0x15971:      mov      rdi, qword ptr [rsp]
0x15975:      mov      r13, rax
0x15978:      call     0x18d10
0x1597d:      mov      rdi, r13
0x15980:      mov      qword ptr [rsp], r13
0x15984:      call     0x49b0
0x15989:      mov      rdi, r13
0x1598c:      lea      rsi, [rax + 2]
0x15990:      mov      r12, rax
0x15993:      call     0x4ce0
0x15998:      mov      esi, 0x41ff
0x1599d:      mov      qword ptr [rsp], rax
0x159a1:      mov      byte ptr [rax + r12], 0x2f
0x159a6:      mov      rax, qword ptr [rsp]
0x159aa:      mov      byte ptr [rax + r12 + 1], 0
0x159b0:      mov      rdi, qword ptr [rsp]
0x159b4:      xor      eax, eax
0x159b6:      call     0x124d0
0x159bb:      cmp      eax, -1
0x159be:      je       0x15a90
0x159c4:      mov      rsi, qword ptr [rbx + 0x18]
0x159c8:      xor      eax, eax

```

```

0x159ca:    mov     edx, 1
0x159cf:    mov     rdi, rbx
0x159d2:    call    0xf880
0x159d7:    test    eax, eax
0x159d9:    jne     0x15a30
0x159db:    mov     rdi, rbx
0x159de:    call    0x14d00
0x159e3:    mov     rsi, qword ptr [rbp + 0x18]
0x159e7:    mov     rdi, qword ptr [rbx + 0x18]
0x159eb:    call    0x4b30
0x159f0:    test    eax, eax
0x159f2:    je      0x15af0
0x159f8:    mov     rdi, rbp
0x159fb:    xor     eax, eax
0x159fd:    call    0x10240
0x15a02:    mov     rdi, qword ptr [rsp]
0x15a06:    call    0x18d10
0x15a0b:    mov     rax, qword ptr [rsp + 8]
0x15a10:    sub     rax, qword ptr fs:[0x28]
0x15a19:    jne     0x15ba4
0x15a1f:    add     rsp, 0x18
0x15a23:    pop     rbx
0x15a24:    pop     rbp
0x15a25:    pop     r12
0x15a27:    pop     r13
0x15a29:    ret
0x15a2a:    nop     word ptr [rax + rax]
0x15a30:    mov     rsi, qword ptr [rsp]
0x15a34:    mov     rdi, rbx
0x15a37:    xor     eax, eax
0x15a39:    call    0x122d0
0x15a3e:    mov     edx, dword ptr [rbx + 0x60]
0x15a41:    mov     esi, dword ptr [rbx + 0x68]
0x15a44:    mov     edi, 0
0x15a49:    mov     dword ptr [rbx + 0x28], eax
0x15a4c:    sub     edx, 2
0x15a4f:    mov     ecx, edx
0x15a51:    shr     ecx, 0x1f
0x15a54:    add     ecx, edx
0x15a56:    sar     ecx, 1
0x15a58:    sub     eax, ecx
0x15a5a:    lea     ecx, [rdx + rax]
0x15a5d:    cmovs   eax, edi
0x15a60:    cmp     esi, ecx
0x15a62:    jg      0x15a70
0x15a64:    mov     eax, esi
0x15a66:    sub     eax, edx
0x15a68:    mov     edx, 0

```

```

0x15a6d:      cmovs    eax, edx
0x15a70:      cmp     byte ptr [rbx + 0x6d], 0
0x15a74:      mov     dword ptr [rbx + 0x2c], eax
0x15a77:      jne     0x15ac0
0x15a79:      mov     rdi, rbx
0x15a7c:      xor     eax, eax
0x15a7e:      call    0x10240
0x15a83:      jmp     0x159e3
0x159e3
0x15a88:      nop     dword ptr [rax + rax]
0x15a90:      mov     rsi, qword ptr [rsp]
0x15a94:      mov     ecx, 0x1c
0x15a99:      xor     edx, edx
0x15a9b:      xor     eax, eax
0x15a9d:      lea     rdi, [rip + 0xb993]
0x15aa4:      call    0xf760
0x15aa9:      mov     rdi, qword ptr [rsp]
0x15aad:      call    0x18d10
0x15ab2:      jmp     0x15a0b
0x15a0b
0x15ab7:      nop     word ptr [rax + rax]
0x15ac0:      xor     eax, eax
0x15ac2:      mov     rdi, rbx
0x15ac5:      call    0x10d70
0x15aca:      cmp     byte ptr [rbx + 0x6d], 0
0x15ace:      je      0x15a79
0x15ad0:      mov     rdi, rbx
0x15ad3:      xor     eax, eax
0x15ad5:      call    0xec50
0x15ada:      mov     rdi, rbx
0x15add:      xor     eax, eax
0x15adf:      call    0x10240
0x15ae4:      jmp     0x159e3
0x159e3
0x15ae9:      nop     dword ptr [rax]
0x15af0:      movsxd  rax, dword ptr [rbp + 0x28]
0x15af4:      lea     rax, [rax + rax*4]
0x15af8:      shl     rax, 4
0x15afc:      add     rax, qword ptr [rbp + 0x38]
0x15b00:      mov     rdi, qword ptr [rax]
0x15b03:      call    0x1d7a0
0x15b08:      mov     rsi, qword ptr [rbp + 0x18]
0x15b0c:      mov     edx, 1
0x15b11:      mov     rdi, rbp
0x15b14:      mov     r12, rax
0x15b17:      xor     eax, eax
0x15b19:      call    0xf880
0x15b1e:      test    eax, eax

```

```

0x15b20:    jne     0x15b40
0x15b22:    mov     rdi, rbp
0x15b25:    call    0x14d00
0x15b2a:    mov     rdi, r12
0x15b2d:    call    0x18d10
0x15b32:    jmp     0x159f8
0x159f8
0x15b37:    nop     word ptr [rax + rax]
0x15b40:    mov     rsi, r12
0x15b43:    mov     rdi, rbp
0x15b46:    xor     eax, eax
0x15b48:    call    0x122d0
0x15b4d:    mov     edx, dword ptr [rbp + 0x60]
0x15b50:    mov     esi, dword ptr [rbp + 0x68]
0x15b53:    mov     edi, 0
0x15b58:    mov     dword ptr [rbp + 0x28], eax
0x15b5b:    sub     edx, 2
0x15b5e:    mov     ecx, edx
0x15b60:    shr     ecx, 0x1f
0x15b63:    add     ecx, edx
0x15b65:    sar     ecx, 1
0x15b67:    sub     eax, ecx
0x15b69:    lea     ecx, [rdx + rax]
0x15b6c:    cmovs   eax, edi
0x15b6f:    cmp     esi, ecx
0x15b71:    jg      0x15b7f
0x15b73:    mov     eax, esi
0x15b75:    sub     eax, edx
0x15b77:    mov     edx, 0
0x15b7c:    cmovs   eax, edx
0x15b7f:    cmp     byte ptr [rbp + 0x6d], 0
0x15b83:    mov     dword ptr [rbp + 0x2c], eax
0x15b86:    je      0x15b2a
0x15b88:    xor     eax, eax
0x15b8a:    mov     rdi, rbp
0x15b8d:    call    0x10d70
0x15b92:    cmp     byte ptr [rbp + 0x6d], 0
0x15b96:    je      0x15b2a
0x15b98:    mov     rdi, rbp
0x15b9b:    xor     eax, eax
0x15b9d:    call    0xec50
0x15ba2:    jmp     0x15b2a
0x15b2a
0x15ba4:    call    0x49d0
0x15ba9:    nop     dword ptr [rax]
0x15bb0:    endbr64
0x15bb4:    push    r15
0x15bb6:    mov     r15, rdi

```

```

0x15bb9:    push    r14
0x15bbb:    push    r13
0x15bbd:    push    r12
0x15bbf:    mov     r12, rsi
0x15bc2:    push    rbp
0x15bc3:    push    rbx
0x15bc4:    sub     rsp, 0x48
0x15bc8:    mov     edx, dword ptr [rdi + 0x70]
0x15bcb:    mov     rax, qword ptr fs:[0x28]
0x15bd4:    mov     qword ptr [rsp + 0x38], rax
0x15bd9:    xor     eax, eax
0x15bdb:    mov     qword ptr [rsp + 0x28], 0
0x15be4:    mov     qword ptr [rsp + 0x30], 0
0x15bed:    mov     dword ptr [rdi + 0x90], 1
0x15bf7:    test    edx, edx
0x15bf9:    jne     0x15ec8
0x15bff:    movsxd  rax, dword ptr [rdi + 0x28]
0x15c03:    mov     rdx, rax
0x15c06:    lea     rax, [rax + rax*4]
0x15c0a:    shl     rax, 4
0x15c0e:    add     rax, qword ptr [rdi + 0x38]
0x15c12:    mov     rbp, qword ptr [rax]
0x15c15:    test    edx, edx
0x15c17:    jne     0x15c30
0x15c19:    mov     rax, qword ptr [rdi + 0x18]
0x15c1d:    cmp     byte ptr [rax], 0x2f
0x15c20:    jne     0x15ea0
0x15c26:    cmp     byte ptr [rax + 1], 0
0x15c2a:    jne     0x15ea0
0x15c30:    mov     rdi, rbp
0x15c33:    call    0x49b0
0x15c38:    lea     rdi, [rax + 0x11]
0x15c3c:    call    0x1d670
0x15c41:    mov     edx, 0xe
0x15c46:    mov     rsi, rbp
0x15c49:    lea     rdi, [rip + 0x15990]
0x15c50:    mov     r13, rax
0x15c53:    call    0x4880
0x15c58:    lea     rcx, [rip + 0xb7ef]
0x15c5f:    mov     esi, 1
0x15c64:    mov     rdi, r13
0x15c67:    mov     r8, rax
0x15c6a:    mov     rdx, -1
0x15c71:    xor     eax, eax
0x15c73:    call    0x4f10
0x15c78:    mov     rdi, rbp
0x15c7b:    call    0x49b0
0x15c80:    mov     rdi, qword ptr [r12 + 0x18]

```

```

0x15c85:    mov     rbx, rax
0x15c88:    call   0x49b0
0x15c8d:    lea     rdi, [rbx + rax + 2]
0x15c92:    call   0x1d670
0x15c97:    mov     r8, qword ptr [r12 + 0x18]
0x15c9c:    mov     r9, rbp
0x15c9f:    lea     rcx, [rip + 0xaa5b]
0x15ca6:    mov     rdi, rax
0x15ca9:    mov     r14, rax
0x15cac:    mov     esi, 1
0x15cb1:    xor     eax, eax
0x15cb3:    mov     rdx, -1
0x15cba:    call   0x4f10
0x15cbf:    mov     rdi, r13
0x15cc2:    lea     rsi, [rsp + 0x30]
0x15cc7:    mov     rdx, r14
0x15cca:    mov     rcx, qword ptr [rip + 0x1593f]
0x15cd1:    call   0x83e0
0x15cd6:    mov     rdi, r13
0x15cd9:    test    rax, rax
0x15cdc:    je      0x16173
0x15ce2:    call   0x18d10
0x15ce7:    movsxd  rax, dword ptr [r15 + 0x28]
0x15ceb:    lea     rax, [rax + rax*4]
0x15cef:    shl     rax, 4
0x15cf3:    add     rax, qword ptr [r15 + 0x38]
0x15cf7:    mov     eax, dword ptr [rax + 0x10]
0x15cfa:    and     eax, 0xf000
0x15cff:    cmp     eax, 0x4000
0x15d04:    je      0x160a0
0x15d0a:    mov     rdi, qword ptr [rip + 0x1479f]
0x15d11:    call   0xdb60
0x15d16:    xor     eax, eax
0x15d18:    call   0xa820
0x15d1d:    mov     rdi, qword ptr [rsp + 0x30]
0x15d22:    xor     eax, eax
0x15d24:    call   0x19940
0x15d29:    mov     rdi, qword ptr [rsp + 0x30]
0x15d2e:    mov     r13, rax
0x15d31:    call   0x18d10
0x15d36:    mov     rdi, rbp
0x15d39:    xor     eax, eax
0x15d3b:    mov     rsi, r13
0x15d3e:    mov     qword ptr [rsp + 0x30], r13
0x15d43:    call   0x11190
0x15d48:    mov     rdi, r14
0x15d4b:    mov     ebx, eax
0x15d4d:    call   0x18d10

```

```

0x15d52:      test     ebx, ebx
0x15d54:      jne      0x16150
0x15d5a:      movsxd   rax, dword ptr [r15 + 0x28]
0x15d5e:      mov      rdx, qword ptr [rsp + 0x30]
0x15d63:      mov      rsi, rbp
0x15d66:      mov      rdi, r15
0x15d69:      lea      rax, [rax + rax*4]
0x15d6d:      shl      rax, 4
0x15d71:      add      rax, qword ptr [r15 + 0x38]
0x15d75:      mov      ecx, dword ptr [rax + 0x10]
0x15d78:      xor      eax, eax
0x15d7a:      call     0x11d10
0x15d7f:      cmp      eax, 1
0x15d82:      je       0x15db6
0x15d84:      cmp      eax, 2
0x15d87:      je       0x161a0
0x15d8d:      sub      eax, 1
0x15d90:      lea      rdx, [rip + 0x14629]
0x15d97:      xor      ecx, ecx
0x15d99:      mov      rsi, rbp
0x15d9c:      cdqe
0x15d9e:      mov      r8d, 0xe
0x15da4:      lea      rdi, [rip + 0xb6b0]
0x15dab:      mov      rdx, qword ptr [rdx + rax*8]
0x15daf:      xor      eax, eax
0x15db1:      call     0xf7e0
0x15db6:      mov      rdi, qword ptr [rsp + 0x30]
0x15dbb:      call     0x18d10
0x15dc0:      xor      eax, eax
0x15dc2:      call     0xe4e0
0x15dc7:      mov      rdi, r15
0x15dca:      xor      eax, eax
0x15dcc:      call     0x10240
0x15dd1:      xor      eax, eax
0x15dd3:      mov      rdi, r12
0x15dd6:      call     0x10240
0x15ddb:      xor      eax, eax
0x15ddd:      call     0xa820
0x15de2:      mov      rsi, qword ptr [r12 + 0x18]
0x15de7:      xor      eax, eax
0x15de9:      mov      edx, 1
0x15dee:      mov      rdi, r12
0x15df1:      call     0xf880
0x15df6:      test     eax, eax
0x15df8:      je       0x16190
0x15dfe:      xor      eax, eax
0x15e00:      mov      rdi, r12
0x15e03:      call     0x10ec0

```



```

0x15e08:    xor     eax, eax
0x15e0a:    mov     rdi, r12
0x15e0d:    call    0x10160
0x15e12:    cmp     byte ptr [r12 + 0x6d], 0
0x15e18:    je      0x15e24
0x15e1a:    mov     rdi, r12
0x15e1d:    xor     eax, eax
0x15e1f:    call    0xec50
0x15e24:    xor     eax, eax
0x15e26:    mov     rdi, r12
0x15e29:    call    0x10240
0x15e2e:    cmp     byte ptr [r12 + 0x6d], 0
0x15e34:    je      0x15e40
0x15e36:    mov     rdi, r12
0x15e39:    xor     eax, eax
0x15e3b:    call    0x10d70
0x15e40:    mov     rsi, qword ptr [r15 + 0x18]
0x15e44:    xor     eax, eax
0x15e46:    mov     edx, 1
0x15e4b:    mov     rdi, r15
0x15e4e:    call    0xf880
0x15e53:    test    eax, eax
0x15e55:    je      0x16180
0x15e5b:    xor     eax, eax
0x15e5d:    mov     rdi, r15
0x15e60:    call    0x10ec0
0x15e65:    xor     eax, eax
0x15e67:    mov     rdi, r15
0x15e6a:    call    0x10160
0x15e6f:    cmp     byte ptr [r15 + 0x6d], 0
0x15e74:    je      0x15e80
0x15e76:    mov     rdi, r15
0x15e79:    xor     eax, eax
0x15e7b:    call    0xec50
0x15e80:    xor     eax, eax
0x15e82:    mov     rdi, r15
0x15e85:    call    0x10240
0x15e8a:    cmp     byte ptr [r15 + 0x6d], 0
0x15e8f:    je      0x15ea0
0x15e91:    mov     rdi, r15
0x15e94:    xor     eax, eax
0x15e96:    call    0x10d70
0x15e9b:    nop     dword ptr [rax + rax]
0x15ea0:    mov     rax, qword ptr [rsp + 0x38]
0x15ea5:    sub     rax, qword ptr fs:[0x28]
0x15eae:    jne     0x161bd
0x15eb4:    add     rsp, 0x48
0x15eb8:    pop     rbx

```

```

0x15eb9:    pop     rbp
0x15eba:    pop     r12
0x15ebc:    pop     r13
0x15ebe:    pop     r14
0x15ec0:    pop     r15
0x15ec2:    ret
0x15ec3:    nop     dword ptr [rax + rax]
0x15ec8:    mov     rdx, qword ptr [rsi + 0x18]
0x15ecc:    mov     rcx, qword ptr [rip + 0x1573d]
0x15ed3:    lea     rsi, [rsp + 0x28]
0x15ed8:    lea     rdi, [rip + 0xb591]
0x15edf:    call    0x83e0
0x15ee4:    test    rax, rax
0x15ee7:    je      0x15ea0
0x15ee9:    mov     rdi, qword ptr [r15 + 0x18]
0x15eed:    mov     rsi, qword ptr [rsp + 0x28]
0x15ef2:    xor     eax, eax
0x15ef4:    call    0x11190
0x15ef9:    mov     r13d, eax
0x15efc:    test    eax, eax
0x15efe:    jne     0x160b8
0x15f04:    mov     rbp, qword ptr [rsp + 0x28]
0x15f09:    mov     rdi, rbp
0x15f0c:    call    0x49b0
0x15f11:    mov     rdi, rbp
0x15f14:    mov     rbx, rax
0x15f17:    lea     rax, [rax + 1]
0x15f1b:    lea     r14, [rbx + 2]
0x15f1f:    mov     qword ptr [rsp + 8], rax
0x15f24:    mov     rsi, r14
0x15f27:    call    0x1d690
0x15f2c:    mov     qword ptr [rsp + 0x28], rax
0x15f31:    mov     byte ptr [rax + rbx], 0x2f
0x15f35:    mov     rax, qword ptr [rsp + 0x28]
0x15f3a:    mov     byte ptr [rax + rbx + 1], 0
0x15f3f:    mov     edx, dword ptr [r15 + 0x68]
0x15f43:    test    edx, edx
0x15f45:    jle     0x15f6a
0x15f47:    mov     rax, qword ptr [r15 + 0x38]
0x15f4b:    add     rax, 0x49
0x15f4f:    jmp     0x15f65
0x15f65:
0x15f51:    nop     dword ptr [rax]
0x15f58:    add     r13d, 1
0x15f5c:    add     rax, 0x50
0x15f60:    cmp     r13d, edx
0x15f63:    je      0x15f6a
0x15f65:    cmp     byte ptr [rax], 0

```

```

0x15f68:    je      0x15f58
0x15f6a:    mov     eax, dword ptr [r15 + 0x70]
0x15f6e:    mov     dword ptr [r15 + 0x50], 0xffffffff
0x15f76:    mov     dword ptr [r15 + 0x54], eax
0x15f7a:    nop     word ptr [rax + rax]
0x15f80:    mov     rdi, r15
0x15f83:    call    0x100e0
0x15f88:    cmp     eax, -1
0x15f8b:    je      0x16060
0x15f91:    cdqe
0x15f93:    lea     rbx, [rax + rax*4]
0x15f97:    mov     rax, qword ptr [r15 + 0x38]
0x15f9b:    shl     rbx, 4
0x15f9f:    mov     rbp, qword ptr [rax + rbx]
0x15fa3:    xor     eax, eax
0x15fa5:    call    0x18ab0
0x15faa:    mov     rdi, rbp
0x15fad:    call    0x49b0
0x15fb2:    mov     rdi, qword ptr [rsp + 0x28]
0x15fb7:    lea     rsi, [r14 + rax]
0x15fbb:    call    0x1d690
0x15fc0:    mov     rcx, qword ptr [rsp + 8]
0x15fc5:    mov     rsi, rbp
0x15fc8:    mov     qword ptr [rsp + 0x28], rax
0x15fcd:    lea     rdi, [rax + rcx]
0x15fd1:    call    0x48b0
0x15fd6:    mov     eax, dword ptr [rip + 0x15ad4]
0x15fdc:    test    eax, eax
0x15fde:    jne     0x160e0
0x15fe4:    mov     rdi, qword ptr [rip + 0x144cd]
0x15feb:    call    0xdb60
0x15ff0:    xor     eax, eax
0x15ff2:    call    0xa820
0x15ff7:    mov     rax, qword ptr [r15 + 0x38]
0x15ffb:    mov     rsi, rbp
0x15ffe:    mov     rdi, r15
0x16001:    mov     rdx, qword ptr [rsp + 0x28]
0x16006:    mov     ecx, dword ptr [rax + rbx + 0x10]
0x1600a:    xor     eax, eax
0x1600c:    call    0x11d10
0x16011:    cmp     eax, 1
0x16014:    je      0x160d0
0x1601a:    cmp     eax, 2
0x1601d:    je      0x16060
0x1601f:    cmp     eax, 3
0x16022:    je      0x15f80
0x16028:    sub     eax, 1
0x1602b:    lea     rdx, [rip + 0x1438e]

```

```

0x16032:    xor     ecx, ecx
0x16034:    mov     rsi, rbp
0x16037:    cdqe
0x16039:    mov     r8d, 0xe
0x1603f:    lea     rdi, [rip + 0xb415]
0x16046:    mov     rdx, qword ptr [rdx + rax*8]
0x1604a:    xor     eax, eax
0x1604c:    call    0xf7e0
0x16051:    test    al, al
0x16053:    jne     0x15f80
0x16059:    nop     dword ptr [rax]
0x16060:    mov     rdi, qword ptr [rsp + 0x28]
0x16065:    test    rdi, rdi
0x16068:    je      0x1606f
0x1606a:    call    0x18d10
0x1606f:    cmp     dword ptr [r15 + 0x68], r13d
0x16073:    je      0x16087
0x16075:    mov     dword ptr [r15 + 0x28], r13d
0x16079:    mov     rdi, r15
0x1607c:    xor     eax, eax
0x1607e:    call    0xf110
0x16083:    mov     dword ptr [r15 + 0x2c], eax
0x16087:    xor     eax, eax
0x16089:    call    0xe4e0
0x1608e:    xor     eax, eax
0x16090:    call    0xa820
0x16095:    jmp     0x15de2
0x15de2
0x1609a:    nop     word ptr [rax + rax]
0x160a0:    mov     rdi, qword ptr [rip + 0x14419]
0x160a7:    call    0xdb60
0x160ac:    jmp     0x15d16
0x15d16
0x160b1:    nop     dword ptr [rax]
0x160b8:    mov     edx, 0xc
0x160bd:    xor     esi, esi
0x160bf:    lea     rdi, [rip + 0x1423a]
0x160c6:    call    0x8070
0x160cb:    jmp     0x15ea0
0x15ea0
0x160d0:    mov     rax, qword ptr [r15 + 0x38]
0x160d4:    mov     byte ptr [rax + rbx + 0x49], 0
0x160d9:    jmp     0x15f80
0x15f80
0x160de:    nop
0x160e0:    mov     dword ptr [rip + 0x159c6], 0
0x160ea:    xor     eax, eax
0x160ec:    call    0xc8c0

```

```

0x160f1:      mov     edx, 4
0x160f6:      lea     rsi, [rip + 0xa435]
0x160fd:      lea     rdi, [rip + 0xb0f5]
0x16104:      mov     qword ptr [rsp + 0x18], rax
0x16109:      call    0x8070
0x1610e:      mov     rdi, qword ptr [rsp + 0x18]
0x16113:      mov     byte ptr [rsp + 0x17], al
0x16117:      call    0xc970
0x1611c:      xor     eax, eax
0x1611e:      call    0xd870
0x16123:      xor     eax, eax
0x16125:      call    0xd740
0x1612a:      xor     eax, eax
0x1612c:      call    0xa820
0x16131:      movzx   eax, byte ptr [rsp + 0x17]
0x16136:      and     eax, 0xfffffffff
0x16139:      cmp     al, 0x4e
0x1613b:      je      0x15fe4
0x16141:      jmp     0x16060
0x16060:
0x16146:      nop     word ptr cs:[rax + rax]
0x16150:      mov     rdx, qword ptr [rsp + 0x30]
0x16155:      lea     rdi, [rip + 0xbaf4]
0x1615c:      xor     ecx, ecx
0x1615e:      xor     eax, eax
0x16160:      mov     r8d, 0x1e
0x16166:      mov     rsi, rbp
0x16169:      call    0xf7e0
0x1616e:      mov     rdi, qword ptr [rsp + 0x30]
0x16173:      call    0x18d10
0x16178:      jmp     0x15ea0
0x15ea0:
0x1617d:      nop     dword ptr [rax]
0x16180:      mov     rdi, r15
0x16183:      call    0x14d00
0x16188:      jmp     0x15e5b
0x15e5b:
0x1618d:      nop     dword ptr [rax]
0x16190:      mov     rdi, r12
0x16193:      call    0x14d00
0x16198:      jmp     0x15dfe
0x15dfe:
0x1619d:      nop     dword ptr [rax]
0x161a0:      mov     rdi, qword ptr [rsp + 0x30]
0x161a5:      call    0x18d10
0x161aa:      xor     eax, eax
0x161ac:      call    0xe4e0
0x161b1:      xor     eax, eax

```

```

0x161b3:      call    0xa820
0x161b8:      jmp     0x15ea0
0x15ea0:
0x161bd:      call    0x49d0
0x161c2:      nop     word ptr cs:[rax + rax]
0x161cd:      nop     dword ptr [rax]
0x161d0:      endbr64
0x161d4:      push    r13
0x161d6:      xor     r13d, r13d
0x161d9:      push    r12
0x161db:      lea     r12, [rip + 0xc203]
0x161e2:      push    rbp
0x161e3:      mov     rbp, rdi
0x161e6:      push    rbx
0x161e7:      mov     rbx, rsi
0x161ea:      sub     rsp, 8
0x161ee:      mov     rdx, qword ptr [rdi + 0x38]
0x161f2:      test    rdx, rdx
0x161f5:      je      0x1621a
0x161f7:      movsxd  rax, dword ptr [rdi + 0x28]
0x161fb:      lea     rax, [rax + rax*4]
0x161ff:      shl     rax, 4
0x16203:      mov     rdi, qword ptr [rdx + rax]
0x16207:      test    rdi, rdi
0x1620a:      je      0x1621a
0x1620c:      call    0x1d7a0
0x16211:      mov     r13d, 1
0x16217:      mov     r12, rax
0x1621a:      xor     edx, edx
0x1621c:      mov     rsi, qword ptr [rbp + 0x18]
0x16220:      cmp     rbx, -1
0x16224:      mov     rdi, rbp
0x16227:      sete    dl
0x1622a:      xor     eax, eax
0x1622c:      call    0xf880
0x16231:      test    eax, eax
0x16233:      je      0x16278
0x16235:      cmp     rbx, -1
0x16239:      je      0x16288
0x1623b:      mov     dword ptr [rbp + 0x28], 0
0x16242:      mov     rdi, rbp
0x16245:      xor     eax, eax
0x16247:      call    0xf110
0x1624c:      mov     dword ptr [rbp + 0x2c], eax
0x1624f:      test    r13d, r13d
0x16252:      jne     0x16260
0x16254:      add     rsp, 8
0x16258:      pop     rbx

```

```

0x16259:      pop     rbp
0x1625a:      pop     r12
0x1625c:      pop     r13
0x1625e:      ret
0x1625f:      nop
0x16260:      add     rsp, 8
0x16264:      mov     rdi, r12
0x16267:      pop     rbx
0x16268:      pop     rbp
0x16269:      pop     r12
0x1626b:      pop     r13
0x1626d:      jmp     0x18d10
0x18d10
0x16272:      nop     word ptr [rax + rax]
0x16278:      mov     rdi, rbp
0x1627b:      xor     eax, eax
0x1627d:      call    0x14d00
0x16282:      jmp     0x1624f
0x1624f
0x16284:      nop     dword ptr [rax]
0x16288:      mov     rsi, r12
0x1628b:      mov     rdi, rbp
0x1628e:      xor     eax, eax
0x16290:      call    0x122d0
0x16295:      mov     rdi, rbp
0x16298:      mov     dword ptr [rbp + 0x28], eax
0x1629b:      xor     eax, eax
0x1629d:      call    0xf110
0x162a2:      mov     dword ptr [rbp + 0x2c], eax
0x162a5:      jmp     0x1624f
0x1624f
0x162a7:      nop     word ptr [rax + rax]
0x162b0:      endbr64
0x162b4:      push    r15
0x162b6:      push    r14
0x162b8:      push    r13
0x162ba:      push    r12
0x162bc:      mov     r12, rdi
0x162bf:      push    rbp
0x162c0:      movsxd  rax, dword ptr [rdi + 0x28]
0x162c4:      mov     rbp, rsi
0x162c7:      lea     rax, [rax + rax*4]
0x162cb:      shl     rax, 4
0x162cf:      add     rax, qword ptr [rdi + 0x38]
0x162d3:      movzx   edx, byte ptr [rax + 0x48]
0x162d7:      mov     r14, qword ptr [rax]
0x162da:      test    dl, dl
0x162dc:      je      0x16300

```

```

0x162de:    cmp     dl, 1
0x162e1:    jne     0x162ed
0x162e3:    cmp     byte ptr [rax + 0x4a], 0
0x162e7:    jne     0x16420
0x162ed:    xor     r13d, r13d
0x162f0:    pop     rbp
0x162f1:    mov     eax, r13d
0x162f4:    pop     r12
0x162f6:    pop     r13
0x162f8:    pop     r14
0x162fa:    pop     r15
0x162fc:    ret
0x162fd:    nop     dword ptr [rax]
0x16300:    cmp     byte ptr [r14], 0x2e
0x16304:    jne     0x163c0
0x1630a:    cmp     byte ptr [r14 + 1], 0x2e
0x1630f:    jne     0x163c0
0x16315:    cmp     byte ptr [r14 + 2], 0
0x1631a:    jne     0x163c0
0x16320:    mov     rax, qword ptr [rdi + 0x18]
0x16324:    cmp     byte ptr [rax], 0x2f
0x16327:    jne     0x16330
0x16329:    cmp     byte ptr [rax + 1], 0
0x1632d:    je      0x162ed
0x1632f:    nop
0x16330:    mov     rax, qword ptr [r12 + 0x78]
0x16335:    lea     rdi, [rax + 1]
0x16339:    call    0x1d670
0x1633e:    mov     rsi, qword ptr [r12 + 0x18]
0x16343:    mov     rdi, rax
0x16346:    mov     r15, rax
0x16349:    call    0x48b0
0x1634e:    mov     edx, 1
0x16353:    mov     rsi, r14
0x16356:    mov     rdi, r12
0x16359:    xor     eax, eax
0x1635b:    call    0xf880
0x16360:    mov     r13d, eax
0x16363:    test    eax, eax
0x16365:    jne     0x165c0
0x1636b:    xor     eax, eax
0x1636d:    mov     rdi, r12
0x16370:    call    0x14d00
0x16375:    xor     eax, eax
0x16377:    mov     rdi, r12
0x1637a:    call    0x10ec0
0x1637f:    xor     eax, eax
0x16381:    mov     rdi, r12

```



```

0x16384:      call    0x10160
0x16389:      cmp     byte ptr [r12 + 0x6d], 0
0x1638f:      jne     0x165b0
0x16395:      xor     eax, eax
0x16397:      mov     rdi, r12
0x1639a:      call    0x10240
0x1639f:      cmp     byte ptr [r12 + 0x6d], 0
0x163a5:      je      0x162ed
0x163ab:      mov     rdi, r12
0x163ae:      xor     eax, eax
0x163b0:      call    0x10d70
0x163b5:      jmp     0x162f0
0x162f0
0x163ba:      nop     word ptr [rax + rax]
0x163c0:      mov     rax, qword ptr [r12 + 0x78]
0x163c5:      lea     rdi, [rax + 1]
0x163c9:      call    0x1d670
0x163ce:      mov     rsi, qword ptr [r12 + 0x18]
0x163d3:      mov     rdi, rax
0x163d6:      mov     r15, rax
0x163d9:      call    0x48b0
0x163de:      mov     edx, 1
0x163e3:      mov     rsi, r14
0x163e6:      mov     rdi, r12
0x163e9:      xor     eax, eax
0x163eb:      call    0xf880
0x163f0:      mov     r13d, eax
0x163f3:      test    eax, eax
0x163f5:      jne     0x164b0
0x163fb:      mov     ecx, 0x1c
0x16400:      xor     edx, edx
0x16402:      mov     rsi, r14
0x16405:      xor     eax, eax
0x16407:      lea     rdi, [rip + 0xaf94]
0x1640e:      call    0xf760
0x16413:      jmp     0x162f0
0x162f0
0x16418:      nop     dword ptr [rax + rax]
0x16420:      mov     rdi, r14
0x16423:      call    0x49b0
0x16428:      lea     rdi, [rax + 0x21]
0x1642c:      call    0x1d670
0x16431:      mov     r8, r14
0x16434:      mov     esi, 1
0x16439:      lea     rcx, [rip + 0xb04b]
0x16440:      mov     rdx, -1
0x16447:      mov     r13, rax
0x1644a:      mov     rdi, rax

```

```

0x1644d:    xor     eax, eax
0x1644f:    call   0x4f10
0x16454:    xor     esi, esi
0x16456:    mov     rdi, r13
0x16459:    call   0x193f0
0x1645e:    mov     rdi, r13
0x16461:    mov     r13d, 1
0x16467:    mov     dword ptr [rip + 0x1446b], 1
0x16471:    call   0x18d10
0x16476:    xor     eax, eax
0x16478:    call   0xa480
0x1647d:    mov     rdi, qword ptr [rip + 0x14444]
0x16484:    call   0xbb10
0x16489:    mov     rdi, r14
0x1648c:    mov     dword ptr [r12 + 0x14], 0xffffffff
0x16495:    mov     dword ptr [rbp + 0x14], 0xffffffff
0x1649c:    call   0xd610
0x164a1:    mov     rdi, qword ptr [r12 + 0x18]
0x164a6:    call   0xc4b0
0x164ab:    jmp     0x162f0
0x162f0:
0x164b0:    mov     dword ptr [r12 + 0x28], 0
0x164b9:    mov     rdi, r12
0x164bc:    xor     eax, eax
0x164be:    call   0xf110
0x164c3:    mov     dword ptr [r12 + 0x2c], eax
0x164c8:    mov     rdi, r15
0x164cb:    call   0x18d10
0x164d0:    xor     eax, eax
0x164d2:    mov     rdi, r12
0x164d5:    call   0x10ec0
0x164da:    xor     eax, eax
0x164dc:    mov     rdi, r12
0x164df:    call   0x10160
0x164e4:    cmp     byte ptr [r12 + 0x6d], 0
0x164ea:    je      0x164f6
0x164ec:    mov     rdi, r12
0x164ef:    xor     eax, eax
0x164f1:    call   0xec50
0x164f6:    xor     eax, eax
0x164f8:    mov     rdi, r12
0x164fb:    call   0x10240
0x16500:    cmp     byte ptr [r12 + 0x6d], 0
0x16506:    je      0x16512
0x16508:    mov     rdi, r12
0x1650b:    xor     eax, eax
0x1650d:    call   0x10d70
0x16512:    mov     rsi, qword ptr [rbp + 0x18]

```

```

0x16516:      mov     rdi, qword ptr [r12 + 0x18]
0x1651b:      call    0x4b30
0x16520:      mov     r13d, eax
0x16523:      test    eax, eax
0x16525:      jne      0x16588
0x16527:      mov     r8d, 1
0x1652d:      mov     rdx, r12
0x16530:      mov     esi, 0x22
0x16535:      mov     rdi, rbp
0x16538:      mov     rcx, -1
0x1653f:      call    0x13e30
0x16544:      xor     eax, eax
0x16546:      mov     rdi, rbp
0x16549:      call    0x10ec0
0x1654e:      xor     eax, eax
0x16550:      mov     rdi, rbp
0x16553:      call    0x10160
0x16558:      cmp     byte ptr [rbp + 0x6d], 0
0x1655c:      jne      0x165a0
0x1655e:      xor     eax, eax
0x16560:      mov     rdi, rbp
0x16563:      call    0x10240
0x16568:      cmp     byte ptr [rbp + 0x6d], 0
0x1656c:      je       0x162ed
0x16572:      mov     rdi, rbp
0x16575:      xor     eax, eax
0x16577:      call    0x10d70
0x1657c:      jmp     0x162f0
0x162f0
0x16581:      nop     dword ptr [rax]
0x16588:      mov     rdi, rbp
0x1658b:      xor     eax, eax
0x1658d:      call    0x10240
0x16592:      jmp     0x162ed
0x162ed
0x16597:      nop     word ptr [rax + rax]
0x165a0:      mov     rdi, rbp
0x165a3:      xor     eax, eax
0x165a5:      call    0xec50
0x165aa:      jmp     0x1655e
0x1655e
0x165ac:      nop     dword ptr [rax]
0x165b0:      mov     rdi, r12
0x165b3:      xor     eax, eax
0x165b5:      call    0xec50
0x165ba:      jmp     0x16395
0x16395
0x165bf:      nop

```

```

0x165c0:      mov     rdi, r15
0x165c3:      mov     esi, 0x2f
0x165c8:      call    0x4a20
0x165cd:      mov     rdi, r12
0x165d0:      lea     rsi, [rax + 1]
0x165d4:      xor     eax, eax
0x165d6:      call    0x122d0
0x165db:      mov     rdi, r12
0x165de:      mov     dword ptr [r12 + 0x28], eax
0x165e3:      xor     eax, eax
0x165e5:      call    0xf110
0x165ea:      jmp     0x164c3
0x164c3:
0x165ef:      nop
0x165f0:      endbr64
0x165f4:      push    r13
0x165f6:      push    r12
0x165f8:      mov     r12, rdx
0x165fb:      push    rbp
0x165fc:      mov     rbp, rsi
0x165ff:      push    rbx
0x16600:      mov     rbx, rdi
0x16603:      mov     rdi, rdx
0x16606:      sub     rsp, 8
0x1660a:      call    0x49b0
0x1660f:      cmp     byte ptr [r12], 0x2f
0x16614:      je      0x166f0
0x1661a:      add     rax, qword ptr [rbx + 0x78]
0x1661e:      mov     rdi, qword ptr [rbx + 0x18]
0x16622:      lea     rdx, [rax + 1]
0x16626:      lea     rsi, [rax + 2]
0x1662a:      mov     qword ptr [rbx + 0x78], rdx
0x1662e:      call    0x1d690
0x16633:      mov     qword ptr [rbx + 0x18], rax
0x16637:      mov     rdi, rax
0x1663a:      mov     r13, rax
0x1663d:      call    0x49b0
0x16642:      mov     edx, 0x2f
0x16647:      mov     rsi, r12
0x1664a:      mov     word ptr [r13 + rax], dx
0x16650:      mov     rdi, qword ptr [rbx + 0x18]
0x16654:      call    0x4df0
0x16659:      mov     rdi, qword ptr [rbx + 0x18]
0x1665d:      call    0x18040
0x16662:      mov     dword ptr [rbx + 0x28], 0
0x16669:      mov     rdi, rbx
0x1666c:      xor     eax, eax
0x1666e:      mov     r8d, 1

```

```

0x16674:      call    0xf110
0x16679:      xor     ecx, ecx
0x1667b:      xor     edx, edx
0x1667d:      mov     esi, 0x22
0x16682:      mov     dword ptr [rbx + 0x2c], eax
0x16685:      mov     rdi, rbx
0x16688:      call    0x13e30
0x1668d:      xor     eax, eax
0x1668f:      mov     rdi, rbx
0x16692:      call    0x10ec0
0x16697:      xor     eax, eax
0x16699:      mov     rdi, rbx
0x1669c:      call    0x10160
0x166a1:      cmp     byte ptr [rbx + 0x6d], 0
0x166a5:      je      0x166b1
0x166a7:      mov     rdi, rbx
0x166aa:      xor     eax, eax
0x166ac:      call    0xec50
0x166b1:      xor     eax, eax
0x166b3:      mov     rdi, rbx
0x166b6:      call    0x10240
0x166bb:      cmp     byte ptr [rbx + 0x6d], 0
0x166bf:      je      0x166cb
0x166c1:      mov     rdi, rbx
0x166c4:      xor     eax, eax
0x166c6:      call    0x10d70
0x166cb:      mov     rsi, qword ptr [rbp + 0x18]
0x166cf:      mov     rdi, qword ptr [rbx + 0x18]
0x166d3:      call    0x4b30
0x166d8:      test    eax, eax
0x166da:      je      0x16718
0x166dc:      add     rsp, 8
0x166e0:      pop     rbx
0x166e1:      pop     rbp
0x166e2:      pop     r12
0x166e4:      pop     r13
0x166e6:      ret
0x166e7:      nop     word ptr [rax + rax]
0x166f0:      mov     qword ptr [rbx + 0x78], rax
0x166f4:      mov     rdi, qword ptr [rbx + 0x18]
0x166f8:      lea     rsi, [rax + 1]
0x166fc:      call    0x1d690
0x16701:      mov     rsi, r12
0x16704:      mov     qword ptr [rbx + 0x18], rax
0x16708:      mov     rdi, rax
0x1670b:      call    0x48b0
0x16710:      jmp     0x16662
0x16662:

```

```

0x16715:      nop      dword ptr [rax]
0x16718:      xor      edx, edx
0x1671a:      mov      r8d, 1
0x16720:      mov      esi, 0x22
0x16725:      mov      rdi, rbp
0x16728:      mov      rcx, -1
0x1672f:      call     0x13e30
0x16734:      xor      eax, eax
0x16736:      mov      rdi, rbp
0x16739:      call     0x10ec0
0x1673e:      xor      eax, eax
0x16740:      mov      rdi, rbp
0x16743:      call     0x10160
0x16748:      cmp      byte ptr [rbp + 0x6d], 0
0x1674c:      jne      0x16780
0x1674e:      xor      eax, eax
0x16750:      mov      rdi, rbp
0x16753:      call     0x10240
0x16758:      cmp      byte ptr [rbp + 0x6d], 0
0x1675c:      je       0x166dc
0x16762:      add      rsp, 8
0x16766:      mov      rdi, rbp
0x16769:      xor      eax, eax
0x1676b:      pop      rbx
0x1676c:      pop      rbp
0x1676d:      pop      r12
0x1676f:      pop      r13
0x16771:      jmp      0x10d70
0x10d70
0x16776:      nop      word ptr cs:[rax + rax]
0x16780:      xor      eax, eax
0x16782:      mov      rdi, rbp
0x16785:      call     0xec50
0x1678a:      xor      eax, eax
0x1678c:      mov      rdi, rbp
0x1678f:      call     0x10240
0x16794:      cmp      byte ptr [rbp + 0x6d], 0
0x16798:      je       0x166dc
0x1679e:      jmp      0x16762
0x16762
0x167a0:      endbr64
0x167a4:      push     r15
0x167a6:      lea      r15, [rip + 0x13bd3]
0x167ad:      push     r14
0x167af:      lea      r14, [rip + 0xacdc]
0x167b6:      push     r13
0x167b8:      mov      r13, rsi
0x167bb:      push     r12

```

```

0x167bd:      mov     r12d, edx
0x167c0:      push    rbp
0x167c1:      mov     rbp, rdi
0x167c4:      push    rbx
0x167c5:      sub     rsp, 8
0x167c9:      mov     rdi, qword ptr [rip + 0x13d08]
0x167d0:      call    0xdb60
0x167d5:      xor     eax, eax
0x167d7:      call    0xa820
0x167dc:      mov     eax, dword ptr [rbp + 0x70]
0x167df:      mov     dword ptr [rbp + 0x50], 0xffffffff
0x167e6:      mov     dword ptr [rbp + 0x90], 1
0x167f0:      mov     dword ptr [rbp + 0x54], eax
0x167f3:      nop     dword ptr [rax + rax]
0x167f8:      mov     rdi, rbp
0x167fb:      call    0x100e0
0x16800:      movsxd  rbx, eax
0x16803:      cmp     ebx, -1
0x16806:      je      0x1686b
0x16808:      xor     eax, eax
0x1680a:      call    0x18ab0
0x1680f:      xor     eax, eax
0x16811:      mov     edx, r12d
0x16814:      mov     esi, ebx
0x16816:      mov     rdi, rbp
0x16819:      call    0x134d0
0x1681e:      cmp     eax, 1
0x16821:      je      0x168f0
0x16827:      cmp     eax, 2
0x1682a:      je      0x1686b
0x1682c:      cmp     eax, 3
0x1682f:      je      0x167f8
0x16831:      sub     eax, 1
0x16834:      xor     ecx, ecx
0x16836:      mov     rdi, r14
0x16839:      mov     r8d, 0x1e
0x1683f:      cdqe
0x16841:      mov     rdx, qword ptr [r15 + rax*8]
0x16845:      lea     rax, [rbx + rbx*4]
0x16849:      shl     rax, 4
0x1684d:      add     rax, qword ptr [rbp + 0x38]
0x16851:      mov     rsi, qword ptr [rax]
0x16854:      xor     eax, eax
0x16856:      call    0xf7e0
0x1685b:      mov     rdi, rbp
0x1685e:      call    0x100e0
0x16863:      movsxd  rbx, eax
0x16866:      cmp     ebx, -1

```

```

0x16869:    jne    0x16808
0x1686b:    xor     eax, eax
0x1686d:    call   0xe4e0
0x16872:    xor     eax, eax
0x16874:    call   0xa820
0x16879:    xor     ecx, ecx
0x1687b:    xor     edx, edx
0x1687d:    mov     r8d, 1
0x16883:    mov     esi, 0x22
0x16888:    mov     rdi, rbp
0x1688b:    call   0x13e30
0x16890:    xor     eax, eax
0x16892:    mov     rdi, rbp
0x16895:    call   0x10ec0
0x1689a:    xor     eax, eax
0x1689c:    mov     rdi, rbp
0x1689f:    call   0x10160
0x168a4:    cmp     byte ptr [rbp + 0x6d], 0
0x168a8:    je      0x168b4
0x168aa:    mov     rdi, rbp
0x168ad:    xor     eax, eax
0x168af:    call   0xec50
0x168b4:    xor     eax, eax
0x168b6:    mov     rdi, rbp
0x168b9:    call   0x10240
0x168be:    cmp     byte ptr [rbp + 0x6d], 0
0x168c2:    je      0x168ce
0x168c4:    mov     rdi, rbp
0x168c7:    xor     eax, eax
0x168c9:    call   0x10d70
0x168ce:    mov     rsi, qword ptr [r13 + 0x18]
0x168d2:    mov     rdi, qword ptr [rbp + 0x18]
0x168d6:    call   0x4b30
0x168db:    test    eax, eax
0x168dd:    je      0x16908
0x168df:    add     rsp, 8
0x168e3:    pop     rbx
0x168e4:    pop     rbp
0x168e5:    pop     r12
0x168e7:    pop     r13
0x168e9:    pop     r14
0x168eb:    pop     r15
0x168ed:    ret
0x168ee:    nop
0x168f0:    lea     rax, [rbx + rbx*4]
0x168f4:    shl     rax, 4
0x168f8:    add     rax, qword ptr [rbp + 0x38]
0x168fc:    mov     byte ptr [rax + 0x49], 0

```



```

0x16900:      jmp      0x167f8
0x167f8
0x16905:      nop      dword ptr [rax]
0x16908:      xor      ecx, ecx
0x1690a:      xor      edx, edx
0x1690c:      mov      r8d, 1
0x16912:      mov      esi, 0x22
0x16917:      mov      rdi, r13
0x1691a:      call     0x13e30
0x1691f:      xor      eax, eax
0x16921:      mov      rdi, r13
0x16924:      call     0x10ec0
0x16929:      xor      eax, eax
0x1692b:      mov      rdi, r13
0x1692e:      call     0x10160
0x16933:      cmp      byte ptr [r13 + 0x6d], 0
0x16938:      jne      0x16968
0x1693a:      xor      eax, eax
0x1693c:      mov      rdi, r13
0x1693f:      call     0x10240
0x16944:      cmp      byte ptr [r13 + 0x6d], 0
0x16949:      je       0x168df
0x1694b:      add      rsp, 8
0x1694f:      mov      rdi, r13
0x16952:      xor      eax, eax
0x16954:      pop      rbx
0x16955:      pop      rbp
0x16956:      pop      r12
0x16958:      pop      r13
0x1695a:      pop      r14
0x1695c:      pop      r15
0x1695e:      jmp      0x10d70
0x10d70
0x16963:      nop      dword ptr [rax + rax]
0x16968:      mov      rdi, r13
0x1696b:      xor      eax, eax
0x1696d:      call     0xec50
0x16972:      jmp      0x1693a
0x1693a
0x16974:      nop      word ptr cs:[rax + rax]
0x1697e:      nop
0x16980:      push     r15
0x16982:      push     r14
0x16984:      push     r13
0x16986:      push     r12
0x16988:      push     rbp
0x16989:      push     rbx
0x1698a:      sub      rsp, 0x8c8

```

```

0x16991:      mov     rdi, qword ptr [rip + 0x14ca8]
0x16998:      mov     r13, qword ptr [rip + 0x13cc1]
0x1699f:      mov     rax, qword ptr fs:[0x28]
0x169a8:      mov     qword ptr [rsp + 0x8b8], rax
0x169b0:      xor     eax, eax
0x169b2:      lea     rax, [rip + 0x14c8f]
0x169b9:      mov     qword ptr [rip + 0x13ca0], rax
0x169c0:      test    rdi, rdi
0x169c3:      je      0x16a78
0x169c9:      lea     rsi, [rsp + 0x20]
0x169ce:      call    0x18f40
0x169d3:      cmp     eax, -1
0x169d6:      je      0x16a78
0x169dc:      mov     rax, qword ptr [rsp + 0x68]
0x169e1:      mov     rdi, qword ptr [rip + 0x14c58]
0x169e8:      lea     rsi, [rip + 0xb5da]
0x169ef:      mov     qword ptr [rsp + 0x10], rax
0x169f4:      mov     rax, qword ptr [rsp + 0x78]
0x169f9:      mov     qword ptr [rsp + 0x18], rax
0x169fe:      call    0x4da0
0x16a03:      mov     r14, rax
0x16a06:      test    rax, rax
0x16a09:      je      0x16a78
0x16a0b:      mov     dword ptr [rsp + 4], 0
0x16a13:      xor     ebp, ebp
0x16a15:      xor     ebx, ebx
0x16a17:      xor     r12d, r12d
0x16a1a:      lea     r15, [rsp + 0xb0]
0x16a22:      nop     word ptr [rax + rax]
0x16a28:      mov     rdx, r14
0x16a2b:      mov     esi, 0x800
0x16a30:      mov     rdi, r15
0x16a33:      call    0x4b00
0x16a38:      test    rax, rax
0x16a3b:      je      0x16ad0
0x16a41:      cmp     byte ptr [rsp + 0xb0], 0
0x16a49:      je      0x16aa8
0x16a4b:      cmp     dword ptr [r15], 0x6d6f7246
0x16a52:      je      0x16ab0
0x16a54:      test    ebp, ebp
0x16a56:      jne     0x16a28
0x16a58:      test    ebx, ebx
0x16a5a:      je      0x16a28
0x16a5c:      cmp     dword ptr [r15], 0x74617453
0x16a63:      je      0x16b60
0x16a69:      xor     ebp, ebp
0x16a6b:      mov     ebx, 1
0x16a70:      jmp     0x16a28

```

```

0x16a28
0x16a72:    nop        word ptr [rax + rax]
0x16a78:    xor        eax, eax
0x16a7a:    mov        rdx, qword ptr [rsp + 0x8b8]
0x16a82:    sub        rdx, qword ptr fs:[0x28]
0x16a8b:    jne        0x16c02
0x16a91:    add        rsp, 0x8c8
0x16a98:    pop        rbx
0x16a99:    pop        rbp
0x16a9a:    pop        r12
0x16a9c:    pop        r13
0x16a9e:    pop        r14
0x16aa0:    pop        r15
0x16aa2:    ret
0x16aa3:    nop        dword ptr [rax + rax]
0x16aa8:    xor        ebx, ebx
0x16aaa:    jmp        0x16a28
0x16a28
0x16aaf:    nop
0x16ab0:    cmp        byte ptr [r15 + 4], 0x20
0x16ab5:    jne        0x16a54
0x16ab7:    add        r12d, 1
0x16abb:    xor        ebp, ebp
0x16abd:    mov        ebx, 1
0x16ac2:    jmp        0x16a28
0x16a28
0x16ac7:    nop        word ptr [rax + rax]
0x16ad0:    mov        rdi, r14
0x16ad3:    call       0x4940
0x16ad8:    mov        rdi, qword ptr [rip + 0x14b61]
0x16adf:    lea        rsi, [rsp + 0x10]
0x16ae4:    call       0x47f0
0x16ae9:    mov        rbp, qword ptr [rip + 0x13b70]
0x16af0:    test       r12d, r12d
0x16af3:    je         0x16b0a
0x16af5:    cmp        r12d, dword ptr [rsp + 4]
0x16afa:    je         0x16b48
0x16afc:    lea        rbp, [rip + 0x13b1d]
0x16b03:    mov        qword ptr [rip + 0x13b56], rbp
0x16b0a:    mov        rdi, rbp
0x16b0d:    call       0x49b0
0x16b12:    mov        rsi, r13
0x16b15:    mov        rdi, rbp
0x16b18:    mov        r8, rax
0x16b1b:    mov        eax, dword ptr [rip + 0x14b2f]
0x16b21:    add        eax, dword ptr [rip + 0x14b2d]
0x16b27:    lea        eax, [rax + r8 + 0x15]
0x16b2c:    mov        dword ptr [rip + 0x14b1a], eax

```

```

0x16b32:      call    0x4b30
0x16b37:      test    eax, eax
0x16b39:      setne   al
0x16b3c:      movzx   eax, al
0x16b3f:      jmp     0x16a7a
0x16a7a
0x16b44:      nop     dword ptr [rax]
0x16b48:      lea     rbp, [rip + 0x13adc]
0x16b4f:      mov     qword ptr [rip + 0x13b0a], rbp
0x16b56:      jmp     0x16b0a
0x16b0a
0x16b58:      nop     dword ptr [rax + rax]
0x16b60:      cmp     word ptr [r15 + 4], 0x7375
0x16b67:      jne     0x16a69
0x16b6d:      cmp     byte ptr [r15 + 6], 0x3a
0x16b72:      jne     0x16a69
0x16b78:      mov     rdi, r15
0x16b7b:      mov     esi, 0x3a
0x16b80:      call    0x4a00
0x16b85:      movsx   rbx, byte ptr [rax + 1]
0x16b8a:      lea     rdi, [rax + 1]
0x16b8e:      test    bl, bl
0x16b90:      je      0x16bf8
0x16b92:      mov     qword ptr [rsp + 8], rdi
0x16b97:      call    0x4f00
0x16b9c:      mov     rdi, qword ptr [rsp + 8]
0x16ba1:      mov     rax, qword ptr [rax]
0x16ba4:      jmp     0x16bbd
0x16bbd
0x16ba6:      nop     word ptr cs:[rax + rax]
0x16bb0:      movsx   rbx, byte ptr [rdi + 1]
0x16bb5:      add     rdi, 1
0x16bb9:      test    bl, bl
0x16bbb:      je      0x16bf8
0x16bbd:      test    byte ptr [rax + rbx*2 + 1], 0x20
0x16bc2:      jne     0x16bb0
0x16bc4:      mov     esi, 0x52
0x16bc9:      mov     ebp, 1
0x16bce:      mov     ebx, 1
0x16bd3:      call    0x4a00
0x16bd8:      test    rax, rax
0x16bdb:      je      0x16a28
0x16be1:      add     dword ptr [rsp + 4], 1
0x16be6:      mov     ebp, 1
0x16beb:      mov     ebx, 1
0x16bf0:      jmp     0x16a28
0x16a28
0x16bf5:      nop     dword ptr [rax]

```

```

0x16bf8:      mov     ebx, 1
0x16bfd:      jmp     0x16a28
0x16a28:
0x16c02:      call    0x49d0
0x16c07:      nop     word ptr [rax + rax]
0x16c10:      endbr64
0x16c14:      sub     rsp, 0xa8
0x16c1b:      mov     edi, 0x1d
0x16c20:      mov     rax, qword ptr fs:[0x28]
0x16c29:      mov     qword ptr [rsp + 0x98], rax
0x16c31:      xor     eax, eax
0x16c33:      call    0x1d670
0x16c38:      lea     r9, [rip + 0x9950]
0x16c3f:      lea     r8, [rip + 0x991f]
0x16c46:      mov     rdx, -1
0x16c4d:      lea     rcx, [rip + 0xb219]
0x16c54:      mov     esi, 1
0x16c59:      mov     rdi, rax
0x16c5c:      mov     qword ptr [rip + 0x149fd], rax
0x16c63:      xor     eax, eax
0x16c65:      call    0x4f10
0x16c6a:      mov     rdi, qword ptr [rip + 0x149ef]
0x16c71:      call    0x49b0
0x16c76:      mov     rdi, qword ptr [rip + 0x14e1b]
0x16c7d:      mov     dword ptr [rip + 0x149d5], eax
0x16c83:      call    0x49b0
0x16c88:      mov     rdi, qword ptr [rip + 0x13d01]
0x16c8f:      mov     dword ptr [rip + 0x149bf], eax
0x16c95:      call    0x49b0
0x16c9a:      lea     rdi, [rip + 0x9999]
0x16ca1:      mov     dword ptr [rip + 0x149a9], eax
0x16ca7:      call    0x47e0
0x16cac:      mov     qword ptr [rip + 0x1498d], rax
0x16cb3:      test    rax, rax
0x16cb6:      je      0x16cc3
0x16cb8:      mov     rdi, rax
0x16cbb:      mov     rsi, rsp
0x16cbe:      call    0x18f40
0x16cc3:      xor     eax, eax
0x16cc5:      call    0x16980
0x16cca:      mov     rdi, qword ptr [rip + 0x1398f]
0x16cd1:      call    0x49b0
0x16cd6:      mov     r8, rax
0x16cd9:      mov     eax, dword ptr [rip + 0x14971]
0x16cdf:      add     eax, dword ptr [rip + 0x1496f]
0x16ce5:      lea     eax, [rax + r8 + 0x15]
0x16cea:      mov     dword ptr [rip + 0x1495c], eax
0x16cf0:      xor     eax, eax

```

```

0x16cf2:      call    0xc590
0x16cf7:      mov     qword ptr [rip + 0x1493a], rax
0x16cfe:      mov     rax, qword ptr [rsp + 0x98]
0x16d06:      sub     rax, qword ptr fs:[0x28]
0x16d0f:      jne     0x16d19
0x16d11:      add     rsp, 0xa8
0x16d18:      ret
0x16d19:      call    0x49d0
0x16d1e:      nop
0x16d20:      endbr64
0x16d24:      mov     rdi, qword ptr [rip + 0x1490d]
0x16d2b:      jmp     0xc5c0
0xc5c0
0x16d30:      endbr64
0x16d34:      mov     r8d, edi
0x16d37:      mov     rdi, qword ptr [rip + 0x148fa]
0x16d3e:      mov     edx, esi
0x16d40:      mov     ecx, 1
0x16d45:      xor     esi, esi
0x16d47:      jmp     0xc5e0
0xc5e0
0x16d4c:      nop     dword ptr [rax]
0x16d50:      endbr64
0x16d54:      push    r14
0x16d56:      push    r13
0x16d58:      push    r12
0x16d5a:      push    rbp
0x16d5b:      push    rbx
0x16d5c:      sub     rsp, 0x10
0x16d60:      mov     rax, qword ptr fs:[0x28]
0x16d69:      mov     qword ptr [rsp + 8], rax
0x16d6e:      xor     eax, eax
0x16d70:      lea     rbp, [rsp + 7]
0x16d75:      mov     rdi, rbp
0x16d78:      call    0xa6a0
0x16d7d:      mov     edx, dword ptr [rip + 0x138c1]
0x16d83:      mov     esi, dword ptr [rip + 0x138b7]
0x16d89:      mov     edi, dword ptr [rip + 0x138b9]
0x16d8f:      call    0xa570
0x16d94:      mov     rdi, qword ptr [rip + 0x1489d]
0x16d9b:      xor     edx, edx
0x16d9d:      xor     esi, esi
0x16d9f:      call    0xc680
0x16da4:      mov     edx, dword ptr [rip + 0x148ae]
0x16daa:      mov     rsi, qword ptr [rip + 0x148af]
0x16db1:      mov     rdi, qword ptr [rip + 0x14880]
0x16db8:      call    0xc600
0x16dbd:      mov     rax, qword ptr [rip + 0x14874]

```

```

0x16dc4:      mov     edi, dword ptr [rax + 0xc]
0x16dc7:      add     edi, 1
0x16dca:      movsxd   rdi, edi
0x16dcd:      call    0x1d670
0x16dd2:      mov     r13, rax
0x16dd5:      xor     eax, eax
0x16dd7:      call    0x16980
0x16ddc:      mov     r14, qword ptr [rip + 0x14855]
0x16de3:      mov     eax, dword ptr [rip + 0x1486f]
0x16de9:      mov     ecx, dword ptr [rip + 0x1485d]
0x16def:      mov     edx, dword ptr [r14 + 0xc]
0x16df3:      lea     esi, [rax + rcx + 2]
0x16df7:      cmp     esi, edx
0x16df9:      jl      0x16e58
0x16dfb:      cmp     eax, edx
0x16dfd:      jl      0x16e30
0x16dff:      mov     rdi, r13
0x16e02:      call    0x18d10
0x16e07:      mov     rdi, rbp
0x16e0a:      call    0xa6b0
0x16e0f:      mov     rax, qword ptr [rsp + 8]
0x16e14:      sub     rax, qword ptr fs:[0x28]
0x16e1d:      jne     0x16fc7
0x16e23:      add     rsp, 0x10
0x16e27:      pop     rbx
0x16e28:      pop     rbp
0x16e29:      pop     r12
0x16e2b:      pop     r13
0x16e2d:      pop     r14
0x16e2f:      ret
0x16e30:      sub     edx, eax
0x16e32:      mov     rdi, r13
0x16e35:      mov     esi, 0x20
0x16e3a:      mov     r12d, edx
0x16e3d:      movsxd   rdx, edx
0x16e40:      call    0x4a50
0x16e45:      mov     edx, r12d
0x16e48:      mov     rsi, r13
0x16e4b:      mov     rdi, r14
0x16e4e:      call    0xc600
0x16e53:      jmp     0x16dff
0x16dff:
0x16e55:      nop     dword ptr [rax]
0x16e58:      sub     edx, eax
0x16e5a:      mov     rdi, r13
0x16e5d:      mov     esi, 0x20
0x16e62:      mov     ebx, edx
0x16e64:      sub     ebx, ecx

```

```

0x16e66:      movsxd  rdx, ebx
0x16e69:      call   0x4a50
0x16e6e:      mov     edx, ebx
0x16e70:      mov     rsi, r13
0x16e73:      mov     rdi, r14
0x16e76:      call   0xc600
0x16e7b:      add     ebx, dword ptr [rip + 0x147d7]
0x16e81:      mov     rdi, qword ptr [rip + 0x147b0]
0x16e88:      xor     esi, esi
0x16e8a:      mov     edx, ebx
0x16e8c:      call   0xc680
0x16e91:      mov     rdi, qword ptr [rip + 0x147a0]
0x16e98:      mov     edx, 5
0x16e9d:      lea     rsi, [rip + 0x13793]
0x16ea4:      call   0xc600
0x16ea9:      mov     rdi, qword ptr [rip + 0x14788]
0x16eb0:      mov     esi, 0x20
0x16eb5:      call   0xc650
0x16eba:      mov     edi, dword ptr [rip + 0x1378c]
0x16ec0:      call   0xa4f0
0x16ec5:      mov     edx, dword ptr [rip + 0x14789]
0x16ecb:      mov     rsi, qword ptr [rip + 0x14bc6]
0x16ed2:      mov     rdi, qword ptr [rip + 0x1475f]
0x16ed9:      call   0xc600
0x16ede:      mov     rdi, qword ptr [rip + 0x14753]
0x16ee5:      mov     esi, 0x20
0x16eea:      call   0xc650
0x16eef:      mov     r12, qword ptr [rip + 0x1376a]
0x16ef6:      mov     rdi, r12
0x16ef9:      call   0x49b0
0x16efe:      mov     rdi, qword ptr [rip + 0x14733]
0x16f05:      mov     rsi, r12
0x16f08:      mov     edx, eax
0x16f0a:      call   0xc600
0x16f0f:      mov     rdi, qword ptr [rip + 0x14722]
0x16f16:      mov     esi, 0x20
0x16f1b:      call   0xc650
0x16f20:      mov     edi, dword ptr [rip + 0x1371a]
0x16f26:      call   0xa4f0
0x16f2b:      mov     rdi, qword ptr [rip + 0x14706]
0x16f32:      mov     edx, 4
0x16f37:      lea     rsi, [rip + 0x136f4]
0x16f3e:      call   0xc600
0x16f43:      mov     rdi, qword ptr [rip + 0x146ee]
0x16f4a:      mov     esi, 0x20
0x16f4f:      call   0xc650
0x16f54:      mov     edi, dword ptr [rip + 0x136f6]
0x16f5a:      call   0xa4f0

```



```

0x16f5f:    mov     edx, dword ptr [rip + 0x146eb]
0x16f65:    mov     rsi, qword ptr [rip + 0x13a24]
0x16f6c:    mov     rdi, qword ptr [rip + 0x146c5]
0x16f73:    call    0xc600
0x16f78:    mov     edi, dword ptr [rip + 0x136c2]
0x16f7e:    call    0xa4f0
0x16f83:    mov     rdi, qword ptr [rip + 0x146ae]
0x16f8a:    mov     esi, 0x20
0x16f8f:    call    0xc650
0x16f94:    xor     edi, edi
0x16f96:    xor     eax, eax
0x16f98:    call    0x16fd0
0x16f9d:    mov     rdi, qword ptr [rip + 0x14694]
0x16fa4:    xor     esi, esi
0x16fa6:    mov     eax, dword ptr [rdi + 0xc]
0x16fa9:    lea     edx, [rax - 1]
0x16fac:    call    0xc680
0x16fb1:    mov     rdi, qword ptr [rip + 0x14680]
0x16fb8:    mov     esi, 0x20
0x16fbd:    call    0xc650
0x16fc2:    jmp     0x16dff
0x16dff:
0x16fc7:    call    0x49d0
0x16fcc:    nop     dword ptr [rax]
0x16fd0:    endbr64
0x16fd4:    push    r12
0x16fd6:    push    rbp
0x16fd7:    push    rbx
0x16fd8:    mov     ebx, edi
0x16fda:    sub     rsp, 0x30
0x16fde:    mov     rax, qword ptr fs:[0x28]
0x16fe7:    mov     qword ptr [rsp + 0x28], rax
0x16fec:    xor     eax, eax
0x16fee:    call    0x7870
0x16ff3:    test    eax, eax
0x16ff5:    jne     0x17114
0x16ffb:    mov     eax, dword ptr [rip + 0x1464b]
0x17001:    mov     edx, dword ptr [rip + 0x14651]
0x17007:    lea     edx, [rdx + rax + 2]
0x1700b:    mov     rax, qword ptr [rip + 0x14626]
0x17012:    cmp     edx, dword ptr [rax + 0xc]
0x17015:    jge     0x17114
0x1701b:    test    ebx, ebx
0x1701d:    jne     0x17140
0x17023:    lea     r12, [rsp + 7]
0x17028:    xor     eax, eax
0x1702a:    call    0x18330
0x1702f:    mov     rdi, r12

```

```

0x17032:    mov     rbp, rax
0x17035:    call   0xa6a0
0x1703a:    lea     rsi, [rsp + 0xc]
0x1703f:    lea     rdi, [rsp + 8]
0x17044:    call   0xa4d0
0x17049:    xor     edi, edi
0x1704b:    call   0xa660
0x17050:    movsxd  r8, dword ptr [rbp + 8]
0x17054:    mov     r9d, dword ptr [rbp + 4]
0x17058:    lea     rbp, [rsp + 0x10]
0x1705d:    lea     rcx, [rip + 0xa0d4]
0x17064:    mov     esi, 1
0x17069:    mov     rdi, rbp
0x1706c:    mov     rax, r8
0x1706f:    imul    r8, r8, 0x2aaaaaab
0x17076:    cdq
0x17077:    sar     r8, 0x21
0x1707b:    sub     r8d, edx
0x1707e:    lea     edx, [r8 + r8*2]
0x17082:    mov     r8d, eax
0x17085:    shl     edx, 2
0x17088:    sub     r8d, edx
0x1708b:    mov     edx, 0xc
0x17090:    cmovbe  r8d, edx
0x17094:    cmp     eax, 0xc
0x17097:    mov     edx, 0x70
0x1709c:    mov     eax, 0x61
0x170a1:    cmovge  eax, edx
0x170a4:    sub     rsp, 8
0x170a8:    mov     edx, 0x10
0x170ad:    push    rax
0x170ae:    xor     eax, eax
0x170b0:    call   0x4f10
0x170b5:    mov     rdi, qword ptr [rip + 0x1457c]
0x170bc:    xor     esi, esi
0x170be:    mov     eax, dword ptr [rdi + 0xc]
0x170c1:    lea     edx, [rax - 7]
0x170c4:    call   0xc680
0x170c9:    mov     edx, dword ptr [rip + 0x13589]
0x170cf:    mov     esi, dword ptr [rip + 0x1357f]
0x170d5:    mov     edi, dword ptr [rip + 0x13581]
0x170db:    call   0xa570
0x170e0:    mov     rdi, rbp
0x170e3:    call   0x49b0
0x170e8:    mov     rdi, qword ptr [rip + 0x14549]
0x170ef:    mov     rsi, rbp
0x170f2:    mov     edx, eax
0x170f4:    call   0xc600

```

```

0x170f9:      mov     esi, dword ptr [rsp + 0x1c]
0x170fd:      mov     edi, dword ptr [rsp + 0x18]
0x17101:      call    0xa4b0
0x17106:      mov     rdi, r12
0x17109:      call    0xa6b0
0x1710e:      pop     rax
0x1710f:      pop     rdx
0x17110:      test    ebx, ebx
0x17112:      jne     0x17130
0x17114:      mov     rax, qword ptr [rsp + 0x28]
0x17119:      sub     rax, qword ptr fs:[0x28]
0x17122:      jne     0x1715b
0x17124:      add     rsp, 0x30
0x17128:      pop     rbx
0x17129:      pop     rbp
0x1712a:      pop     r12
0x1712c:      ret
0x1712d:      nop     dword ptr [rax]
0x17130:      xor     eax, eax
0x17132:      call    0xa820
0x17137:      jmp     0x17114
0x17114:
0x17139:      nop     dword ptr [rax]
0x17140:      xor     eax, eax
0x17142:      call    0x16980
0x17147:      test    eax, eax
0x17149:      je      0x17023
0x1714f:      xor     eax, eax
0x17151:      call    0x16d50
0x17156:      jmp     0x17023
0x17023:
0x1715b:      call    0x49d0
0x17160:      push    rbp
0x17161:      mov     esi, 0x400
0x17166:      push    rbx
0x17167:      lea     rbx, [rip + 0x14512]
0x1716e:      mov     rdi, rbx
0x17171:      sub     rsp, 8
0x17175:      mov     rdx, qword ptr [rip + 0x1490c]
0x1717c:      call    0x4b00
0x17181:      test    rax, rax
0x17184:      je      0x171d8
0x17186:      mov     rdi, rbx
0x17189:      call    0x49b0
0x1718e:      mov     rbp, rax
0x17191:      cmp     rax, 0x3ff
0x17197:      je      0x171e8
0x17199:      mov     esi, 0x23

```

```

0x1719e:    mov     rdi, rbx
0x171a1:    call   0x4a00
0x171a6:    test    rax, rax
0x171a9:    je      0x171c0
0x171ab:    mov     byte ptr [rax], 0
0x171ae:    mov     eax, 1
0x171b3:    add     rsp, 8
0x171b7:    pop     rbx
0x171b8:    pop     rbp
0x171b9:    ret
0x171ba:    nop     word ptr [rax + rax]
0x171c0:    cmp     byte ptr [rbx + rbp - 1], 0xa
0x171c5:    mov     eax, 1
0x171ca:    jne     0x171b3
0x171cc:    mov     byte ptr [rbx + rbp - 1], 0
0x171d1:    jmp     0x171b3
0x171b3:
0x171d3:    nop     dword ptr [rax + rax]
0x171d8:    add     rsp, 8
0x171dc:    xor     eax, eax
0x171de:    pop     rbx
0x171df:    pop     rbp
0x171e0:    ret
0x171e1:    nop     dword ptr [rax]
0x171e8:    mov     rcx, qword ptr [rip + 0x136f9]
0x171ef:    mov     rdi, qword ptr [rip + 0x1368a]
0x171f6:    lea     rdx, [rip + 0xaceb]
0x171fd:    xor     eax, eax
0x171ff:    mov     esi, 1
0x17204:    call   0x4e60
0x17209:    nop     dword ptr [rax]
0x17210:    mov     rdi, qword ptr [rip + 0x14871]
0x17217:    call   0x4a90
0x1721c:    cmp     eax, 0xa
0x1721f:    je      0x17199
0x17225:    cmp     eax, -1
0x17228:    jne     0x17210
0x1722a:    jmp     0x17199
0x17199:
0x1722f:    nop
0x17230:    endbr64
0x17234:    test    rdi, rdi
0x17237:    je      0x17270
0x17239:    sub     rsp, 8
0x1723d:    lea     rsi, [rip + 0xad85]
0x17244:    call   0x4da0
0x17249:    mov     qword ptr [rip + 0x14838], rax
0x17250:    test    rax, rax

```

```

0x17253:      je      0x17269
0x17255:      mov     dword ptr [rip + 0x14821], 0xffffffff
0x1725f:      mov     eax, 1
0x17264:      add     rsp, 8
0x17268:      ret
0x17269:      xor     eax, eax
0x1726b:      add     rsp, 8
0x1726f:      ret
0x17270:      xor     eax, eax
0x17272:      ret
0x17273:      nop     word ptr cs:[rax + rax]
0x1727e:      nop
0x17280:      endbr64
0x17284:      mov     rdi, qword ptr [rip + 0x147fd]
0x1728b:      test    rdi, rdi
0x1728e:      je      0x17298
0x17290:      jmp     0x4940
0x4940
0x17295:      nop     dword ptr [rax]
0x17298:      ret
0x17299:      nop     dword ptr [rax]
0x172a0:      endbr64
0x172a4:      push    rbp
0x172a5:      xor     edx, edx
0x172a7:      xor     esi, esi
0x172a9:      lea     rbp, [rip + 0x143d0]
0x172b0:      push    rbx
0x172b1:      mov     rbx, rdi
0x172b4:      sub     rsp, 8
0x172b8:      mov     rdi, qword ptr [rip + 0x147c9]
0x172bf:      call    0x4c80
0x172c4:      jmp     0x172df
0x172df
0x172c6:      nop     word ptr cs:[rax + rax]
0x172d0:      mov     rsi, rbp
0x172d3:      mov     rdi, rbx
0x172d6:      call    0x4b30
0x172db:      test    eax, eax
0x172dd:      je      0x17300
0x172df:      xor     eax, eax
0x172e1:      call    0x17160
0x172e6:      test    eax, eax
0x172e8:      jne     0x172d0
0x172ea:      mov     eax, 0xffffffff
0x172ef:      mov     dword ptr [rip + 0x1478b], eax
0x172f5:      add     rsp, 8
0x172f9:      pop     rbx
0x172fa:      pop     rbp

```

```

0x172fb:    ret
0x172fc:    nop     dword ptr [rax]
0x17300:    mov     rdi, qword ptr [rip + 0x14781]
0x17307:    call    0x4b80
0x1730c:    mov     dword ptr [rip + 0x1476e], eax
0x17312:    add     rsp, 8
0x17316:    pop     rbx
0x17317:    pop     rbp
0x17318:    ret
0x17319:    nop     dword ptr [rax]
0x17320:    endbr64
0x17324:    push    r15
0x17326:    push    r14
0x17328:    push    r13
0x1732a:    mov     r13, rdi
0x1732d:    push    r12
0x1732f:    push    rbp
0x17330:    push    rbx
0x17331:    mov     ebx, ecx
0x17333:    sub     rsp, 0x438
0x1733a:    mov     qword ptr [rsp + 0x10], rsi
0x1733f:    mov     dword ptr [rsp + 0xc], edx
0x17343:    mov     rax, qword ptr fs:[0x28]
0x1734c:    mov     qword ptr [rsp + 0x428], rax
0x17354:    xor     eax, eax
0x17356:    test    ecx, ecx
0x17358:    jne     0x17618
0x1735e:    cmp     dword ptr [rsp + 0xc], 1
0x17363:    jne     0x175e0
0x17369:    mov     rax, qword ptr [rsp + 0x10]
0x1736e:    mov     qword ptr [rax], 0
0x17375:    lea     rbp, [rip + 0x14304]
0x1737c:    lea     r12, [rsp + 0x20]
0x17381:    lea     r14, [rip + 0xa092]
0x17388:    nop     dword ptr [rax + rax]
0x17390:    xor     eax, eax
0x17392:    call    0x17160
0x17397:    test    eax, eax
0x17399:    je      0x17600
0x1739f:    cmp     byte ptr [rbp], 0
0x173a3:    je      0x17600
0x173a9:    mov     rdx, r12
0x173ac:    mov     rsi, r14
0x173af:    mov     rdi, rbp
0x173b2:    xor     eax, eax
0x173b4:    mov     byte ptr [rsp + 0x20], 0
0x173b9:    call    0x4c60
0x173be:    call    0x4f00

```

```

0x173c3:    mov     r15, rax
0x173c6:    test    ebx, ebx
0x173c8:    jne     0x17410
0x173ca:    movsx   rcx, byte ptr [rsp + 0x20]
0x173d0:    mov     rax, qword ptr [rax]
0x173d3:    mov     byte ptr [rsp + 0x40], 0
0x173d8:    test    byte ptr [rax + rcx*2 + 1], 0x40
0x173dd:    jne     0x17433
0x173df:    mov     rax, qword ptr [rsp + 0x428]
0x173e7:    sub     rax, qword ptr fs:[0x28]
0x173f0:    jne     0x17657
0x173f6:    add     rsp, 0x438
0x173fd:    pop     rbx
0x173fe:    pop     rbp
0x173ff:    pop     r12
0x17401:    pop     r13
0x17403:    pop     r14
0x17405:    pop     r15
0x17407:    ret
0x17408:    nop     dword ptr [rax + rax]
0x17410:    movsx   rcx, byte ptr [rsp + 0x20]
0x17416:    mov     rax, qword ptr [rax]
0x17419:    test    byte ptr [rax + rcx*2 + 1], 0x40
0x1741e:    je      0x173df
0x17420:    mov     rsi, r12
0x17423:    mov     rdi, r13
0x17426:    call    0x4b30
0x1742b:    test    eax, eax
0x1742d:    jne     0x17390
0x17433:    mov     esi, 0x3d
0x17438:    mov     rdi, rbp
0x1743b:    call    0x4a00
0x17440:    test    rax, rax
0x17443:    je      0x174e3
0x17449:    cmp     byte ptr [rax + 1], 0
0x1744d:    je      0x174e3
0x17453:    mov     rdx, qword ptr [rsp + 0x10]
0x17458:    lea     rdi, [rax + 1]
0x1745c:    mov     qword ptr [rdx], rdi
0x1745f:    movzx   ecx, byte ptr [rax + 1]
0x17463:    test    cl, cl
0x17465:    je      0x1763f
0x1746b:    cmp     dword ptr [rsp + 0xc], 1
0x17470:    jle     0x1763f
0x17476:    mov     r8d, 1
0x1747c:    jmp     0x1748e
0x1748e:
0x1747e:    nop

```

```

0x17480:    mov     ecx, eax
0x17482:    cmp     r8d, dword ptr [rsp + 0xc]
0x17487:    jge     0x174d1
0x17489:    test    sil, sil
0x1748c:    je      0x174d1
0x1748e:    movzx   eax, byte ptr [rdi + 1]
0x17492:    test    al, al
0x17494:    setne   sil
0x17498:    add     rdi, 1
0x1749c:    cmp     cl, 0x3b
0x1749f:    jne     0x17480
0x174a1:    mov     byte ptr [rdi - 1], 0
0x174a5:    cmp     al, 0x3b
0x174a7:    je      0x17638
0x174ad:    test    sil, sil
0x174b0:    je      0x17638
0x174b6:    mov     rdx, qword ptr [rsp + 0x10]
0x174bb:    movsxd  rax, r8d
0x174be:    mov     qword ptr [rdx + rax*8], rdi
0x174c2:    movzx   ecx, byte ptr [rdi]
0x174c5:    test    cl, cl
0x174c7:    setne   sil
0x174cb:    add     r8d, 1
0x174cf:    jmp     0x17482
0x17482:
0x174d1:    mov     esi, 0x3b
0x174d6:    call    0x4a00
0x174db:    test    rax, rax
0x174de:    je      0x174ef
0x174e0:    mov     byte ptr [rax], 0
0x174e3:    mov     eax, dword ptr [rsp + 0xc]
0x174e7:    test    eax, eax
0x174e9:    jle     0x175c0
0x174ef:    mov     rsi, qword ptr [rsp + 0x10]
0x174f4:    xor     r8d, r8d
0x174f7:    jmp     0x17513
0x17513:
0x174f9:    nop     dword ptr [rax]
0x17500:    add     r8d, 1
0x17504:    add     rsi, 8
0x17508:    cmp     dword ptr [rsp + 0xc], r8d
0x1750d:    jle     0x175c0
0x17513:    mov     rbp, qword ptr [rsi]
0x17516:    test    rbp, rbp
0x17519:    je      0x17500
0x1751b:    mov     r14, qword ptr [r15]
0x1751e:    movsx   rax, byte ptr [rbp]
0x17523:    test    byte ptr [r14 + rax*2 + 1], 0x20

```



```

0x17529:      je      0x17547
0x1752b:      lea     rax, [rbp + 1]
0x1752f:      nop
0x17530:      mov     rbp, rax
0x17533:      mov     qword ptr [rsi], rax
0x17536:      add     rax, 1
0x1753a:      movsx   rcx, byte ptr [rbp]
0x1753f:      test    byte ptr [r14 + rcx*2 + 1], 0x20
0x17545:      jne     0x17530
0x17547:      mov     rdi, rbp
0x1754a:      mov     qword ptr [rsp + 0x18], rsi
0x1754f:      mov     dword ptr [rsp + 0x10], r8d
0x17554:      call    0x49b0
0x17559:      mov     r8d, dword ptr [rsp + 0x10]
0x1755e:      mov     rsi, qword ptr [rsp + 0x18]
0x17563:      lea     rax, [rbp + rax - 1]
0x17568:      movsx   rcx, byte ptr [rax]
0x1756c:      test    byte ptr [r14 + rcx*2 + 1], 0x20
0x17572:      jne     0x17591
0x17574:      jmp     0x17596
0x17596:
0x17576:      nop     word ptr cs:[rax + rax]
0x17580:      movsx   rcx, byte ptr [rax - 1]
0x17585:      sub     rax, 1
0x17589:      test    byte ptr [r14 + rcx*2 + 1], 0x20
0x1758f:      je      0x17596
0x17591:      cmp     rax, rbp
0x17594:      jae     0x17580
0x17596:      mov     byte ptr [rax + 1], 0
0x1759a:      mov     rax, qword ptr [rsi]
0x1759d:      cmp     byte ptr [rax], 0
0x175a0:      jne     0x17500
0x175a6:      mov     qword ptr [rsi], 0
0x175ad:      add     r8d, 1
0x175b1:      add     rsi, 8
0x175b5:      cmp     dword ptr [rsp + 0xc], r8d
0x175ba:      jg      0x17513
0x175c0:      test    ebx, ebx
0x175c2:      jne     0x173df
0x175c8:      mov     rsi, r12
0x175cb:      mov     rdi, r13
0x175ce:      call    0x48b0
0x175d3:      jmp     0x173df
0x173df:
0x175d8:      nop     dword ptr [rax + rax]
0x175e0:      movsxd  rdx, dword ptr [rsp + 0xc]
0x175e5:      mov     rdi, qword ptr [rsp + 0x10]
0x175ea:      xor     esi, esi

```

```

0x175ec:      shl     rdx, 3
0x175f0:      call    0x4a50
0x175f5:      jmp     0x17375
0x17375
0x175fa:      nop     word ptr [rax + rax]
0x17600:      test    ebx, ebx
0x17602:      jne     0x173df
0x17608:      mov     byte ptr [r13], 0
0x1760d:      jmp     0x173df
0x173df
0x17612:      nop     word ptr [rax + rax]
0x17618:      movsxd   rsi, dword ptr [rip + 0x14461]
0x1761f:      mov     rdi, qword ptr [rip + 0x14462]
0x17626:      xor     edx, edx
0x17628:      call    0x4c80
0x1762d:      jmp     0x1735e
0x1735e
0x17632:      nop     word ptr [rax + rax]
0x17638:      mov     ecx, eax
0x1763a:      jmp     0x174cb
0x174cb
0x1763f:      mov     esi, 0x3b
0x17644:      call    0x4a00
0x17649:      test    rax, rax
0x1764c:      jne     0x174e0
0x17652:      jmp     0x174e3
0x174e3
0x17657:      call    0x49d0
0x1765c:      nop     dword ptr [rax]
0x17660:      endbr64
0x17664:      push    r12
0x17666:      mov     edi, 0xf
0x1766b:      lea     r12, [rip + 0xa8b4]
0x17672:      call    0x1d670
0x17677:      mov     rcx, r12
0x1767a:      lea     r9, [rip + 0xa895]
0x17681:      lea     r8, [rip + 0xa893]
0x17688:      mov     rdi, rax
0x1768b:      mov     rdx, -1
0x17692:      mov     esi, 1
0x17697:      mov     qword ptr [rip + 0x14402], rax
0x1769e:      xor     eax, eax
0x176a0:      call    0x4f10
0x176a5:      mov     edi, 0x17
0x176aa:      call    0x1d670
0x176af:      mov     rcx, r12
0x176b2:      pop     r12
0x176b4:      lea     r9, [rip + 0xa870]

```

```

0x176bb:    mov     qword ptr [rip + 0x143e6], rax
0x176c2:    mov     rdi, rax
0x176c5:    xor     eax, eax
0x176c7:    lea     r8, [rip + 0xa84d]
0x176ce:    mov     rdx, -1
0x176d5:    mov     esi, 1
0x176da:    jmp     0x4f10
0x4f10:
0x176df:    nop
0x176e0:    endbr64
0x176e4:    push    rbp
0x176e5:    mov     rbp, rdi
0x176e8:    xor     edi, edi
0x176ea:    push    rbx
0x176eb:    sub     rsp, 0x18
0x176ef:    mov     rax, qword ptr fs:[0x28]
0x176f8:    mov     qword ptr [rsp + 8], rax
0x176fd:    xor     eax, eax
0x176ff:    call    0x4be0
0x17704:    mov     rdi, rsp
0x17707:    mov     qword ptr [rsp], rax
0x1770b:    call    0x4840
0x17710:    mov     rbx, rax
0x17713:    call    0x4920
0x17718:    movsxd  rcx, dword ptr [rbx + 0x10]
0x1771c:    sub     rsp, 8
0x17720:    lea     rdx, [rip + 0x12fd9]
0x17727:    mov     r9d, dword ptr [rbx + 8]
0x1772b:    mov     r8d, dword ptr [rbx + 0xc]
0x1772f:    mov     esi, 1
0x17734:    mov     rcx, qword ptr [rdx + rcx*8]
0x17738:    push    rbp
0x17739:    lea     rdx, [rip + 0xa8d0]
0x17740:    push    rax
0x17741:    mov     rdi, qword ptr [rip + 0x13138]
0x17748:    push    qword ptr [rip + 0x1319a]
0x1774e:    mov     eax, dword ptr [rbx]
0x17750:    push    rax
0x17751:    mov     eax, dword ptr [rbx + 4]
0x17754:    push    rax
0x17755:    xor     eax, eax
0x17757:    call    0x4e60
0x1775c:    add     rsp, 0x30
0x17760:    mov     rax, qword ptr [rsp + 8]
0x17765:    sub     rax, qword ptr fs:[0x28]
0x1776e:    jne     0x17777
0x17770:    add     rsp, 0x18
0x17774:    pop     rbx

```

```

0x17775:      pop      rbp
0x17776:      ret
0x17777:      call     0x49d0
0x1777c:      nop      dword ptr [rax]
0x17780:      endbr64
0x17784:      push     rbp
0x17785:      xor      eax, eax
0x17787:      mov      ebp, edi
0x17789:      call     0x8ad0
0x1778e:      cmp      ebp, 0xb
0x17791:      je       0x177d3
0x17793:      jg       0x177c0
0x17795:      lea      eax, [rbp - 1]
0x17798:      cmp      eax, 1
0x1779b:      ja       0x17802
0x1779d:      cmp      ebp, 1
0x177a0:      lea      rdi, [rip + 0xa791]
0x177a7:      lea      rax, [rip + 0xa78e]
0x177ae:      cmovne   rdi, rax
0x177b2:      call     0x176e0
0x177b7:      mov      edi, ebp
0x177b9:      call     0x4e20
0x177be:      nop
0x177c0:      cmp      ebp, 0xf
0x177c3:      jne      0x17802
0x177c5:      lea      rdi, [rip + 0x8dbe]
0x177cc:      call     0x176e0
0x177d1:      jmp      0x177b7
0x177b7:
0x177d3:      lea      rdi, [rip + 0xa766]
0x177da:      call     0x176e0
0x177df:      mov      rcx, qword ptr [rip + 0x13102]
0x177e6:      mov      rdi, qword ptr [rip + 0x13093]
0x177ed:      lea      rdx, [rip + 0xa88c]
0x177f4:      xor      eax, eax
0x177f6:      mov      esi, 1
0x177fb:      call     0x4e60
0x17800:      jmp      0x177b7
0x177b7:
0x17802:      mov      rcx, qword ptr [rip + 0x130df]
0x17809:      mov      rdi, qword ptr [rip + 0x13070]
0x17810:      mov      r8d, ebp
0x17813:      xor      eax, eax
0x17815:      lea      rdx, [rip + 0xa82c]
0x1781c:      mov      esi, 1
0x17821:      call     0x4e60
0x17826:      jmp      0x177df
0x177df:

```

```

0x17828:      nop      dword ptr [rax + rax]
0x17830:      endbr64
0x17834:      push     rbx
0x17835:      mov      rcx, qword ptr [rip + 0x130ac]
0x1783c:      mov      rbx, rdi
0x1783f:      mov      esi, 1
0x17844:      mov      rdi, qword ptr [rip + 0x13035]
0x1784b:      lea      rdx, [rip + 0xa6f3]
0x17852:      xor      eax, eax
0x17854:      call     0x4e60
0x17859:      mov      r8, rbx
0x1785c:      mov      rcx, qword ptr [rip + 0x13085]
0x17863:      pop      rbx
0x17864:      mov      rdi, qword ptr [rip + 0x13015]
0x1786b:      lea      rdx, [rip + 0xa836]
0x17872:      mov      esi, 1
0x17877:      xor      eax, eax
0x17879:      jmp      0x4e60
0x4e60
0x1787e:      nop
0x17880:      endbr64
0x17884:      sub      rsp, 8
0x17888:      mov      r8, rdi
0x1788b:      mov      rcx, qword ptr [rip + 0x13056]
0x17892:      xor      eax, eax
0x17894:      mov      rdi, qword ptr [rip + 0x12fe5]
0x1789b:      lea      rdx, [rip + 0xa836]
0x178a2:      mov      esi, 1
0x178a7:      call     0x4e60
0x178ac:      mov      rcx, qword ptr [rip + 0x13035]
0x178b3:      mov      esi, 1
0x178b8:      xor      eax, eax
0x178ba:      mov      rdi, qword ptr [rip + 0x12fbf]
0x178c1:      lea      rdx, [rip + 0xa840]
0x178c8:      call     0x4e60
0x178cd:      mov      rcx, qword ptr [rip + 0x13014]
0x178d4:      mov      esi, 1
0x178d9:      xor      eax, eax
0x178db:      mov      rdi, qword ptr [rip + 0x12f9e]
0x178e2:      lea      rdx, [rip + 0xa84f]
0x178e9:      call     0x4e60
0x178ee:      mov      rcx, qword ptr [rip + 0x12ff3]
0x178f5:      mov      esi, 1
0x178fa:      xor      eax, eax
0x178fc:      mov      rdi, qword ptr [rip + 0x12f7d]
0x17903:      lea      rdx, [rip + 0xa86e]
0x1790a:      call     0x4e60
0x1790f:      mov      rcx, qword ptr [rip + 0x12fd2]

```

```

0x17916:      mov     esi, 1
0x1791b:      xor     eax, eax
0x1791d:      mov     rdi, qword ptr [rip + 0x12f5c]
0x17924:      lea     rdx, [rip + 0xa885]
0x1792b:      call    0x4e60
0x17930:      mov     rcx, qword ptr [rip + 0x12fb1]
0x17937:      mov     esi, 1
0x1793c:      xor     eax, eax
0x1793e:      mov     rdi, qword ptr [rip + 0x12f3b]
0x17945:      lea     rdx, [rip + 0xa8a4]
0x1794c:      call    0x4e60
0x17951:      mov     rcx, qword ptr [rip + 0x12f90]
0x17958:      mov     esi, 1
0x1795d:      xor     eax, eax
0x1795f:      mov     rdi, qword ptr [rip + 0x12f1a]
0x17966:      lea     rdx, [rip + 0xa8c3]
0x1796d:      call    0x4e60
0x17972:      mov     rcx, qword ptr [rip + 0x12f6f]
0x17979:      mov     esi, 1
0x1797e:      xor     eax, eax
0x17980:      mov     rdi, qword ptr [rip + 0x12ef9]
0x17987:      lea     rdx, [rip + 0xa8ea]
0x1798e:      add     rsp, 8
0x17992:      jmp     0x4e60
0x4e60
0x17997:      nop     word ptr [rax + rax]
0x179a0:      endbr64
0x179a4:      push    r12
0x179a6:      lea     r12, [rip + 0x12d33]
0x179ad:      push    rbp
0x179ae:      sub     rsp, 8
0x179b2:      mov     rdi, qword ptr [rip + 0x140ef]
0x179b9:      call    0x49b0
0x179be:      mov     rdi, r12
0x179c1:      mov     rbp, rax
0x179c4:      call    0x49b0
0x179c9:      lea     rdi, [rbp + rax + 9]
0x179ce:      call    0x1d670
0x179d3:      mov     rsi, qword ptr [rip + 0x140ce]
0x179da:      mov     rdi, rax
0x179dd:      mov     rbp, rax
0x179e0:      call    0x4980
0x179e5:      mov     rsi, r12
0x179e8:      mov     rdi, rax
0x179eb:      call    0x4980
0x179f0:      mov     edx, 0x6e6f
0x179f5:      mov     rdi, rbp
0x179f8:      mov     dword ptr [rax], 0x6d6d6f63

```

```

0x179fe:      mov     word ptr [rax + 4], dx
0x17a02:      mov     byte ptr [rax + 6], 0
0x17a06:      call    0x17230
0x17a0b:      test    eax, eax
0x17a0d:      je      0x17a17
0x17a0f:      add     rsp, 8
0x17a13:      pop     rbp
0x17a14:      pop     r12
0x17a16:      ret
0x17a17:      mov     rdi, rbp
0x17a1a:      call    0x17830
0x17a1f:      mov     edi, 1
0x17a24:      call    0x4e20
0x17a29:      nop     dword ptr [rax]
0x17a30:      endbr64
0x17a34:      push    r13
0x17a36:      push    r12
0x17a38:      lea     r12, [rip + 0x12c91]
0x17a3f:      push    rbp
0x17a40:      push    rbx
0x17a41:      sub     rsp, 8
0x17a45:      mov     rdi, qword ptr [rip + 0x12ea4]
0x17a4c:      call    0x49b0
0x17a51:      mov     rdi, r12
0x17a54:      mov     rbx, rax
0x17a57:      call    0x49b0
0x17a5c:      mov     rdi, qword ptr [rip + 0x12eed]
0x17a63:      mov     rbp, rax
0x17a66:      call    0x49b0
0x17a6b:      lea     rdi, [rbx + rbp + 2]
0x17a70:      add     rdi, rax
0x17a73:      call    0x1d670
0x17a78:      mov     rsi, qword ptr [rip + 0x12e71]
0x17a7f:      mov     rdi, rax
0x17a82:      mov     rbp, rax
0x17a85:      call    0x4980
0x17a8a:      mov     rsi, r12
0x17a8d:      mov     rdi, rax
0x17a90:      call    0x4980
0x17a95:      mov     rsi, qword ptr [rip + 0x12eb4]
0x17a9c:      mov     rdi, rax
0x17a9f:      call    0x48b0
0x17aa4:      mov     rdi, rbp
0x17aa7:      call    0x17230
0x17aac:      test    eax, eax
0x17aae:      jne     0x17b2a
0x17ab0:      mov     rdi, rbp
0x17ab3:      lea     r12, [rip + 0x12c06]

```

```

0x17aba:    call    0x18d10
0x17abf:    mov     rdi, qword ptr [rip + 0x12e2a]
0x17ac6:    call    0x49b0
0x17acb:    mov     rdi, r12
0x17ace:    mov     rbp, rax
0x17ad1:    call    0x49b0
0x17ad6:    mov     rdi, qword ptr [rip + 0x12e73]
0x17add:    mov     rbx, rax
0x17ae0:    call    0x49b0
0x17ae5:    lea     rdi, [rbp + rbx + 2]
0x17aea:    add     rdi, rax
0x17aed:    call    0x1d670
0x17af2:    mov     rsi, qword ptr [rip + 0x12df7]
0x17af9:    mov     rdi, rax
0x17afc:    mov     rbp, rax
0x17aff:    call    0x4980
0x17b04:    mov     rsi, r12
0x17b07:    mov     rdi, rax
0x17b0a:    call    0x4980
0x17b0f:    mov     rsi, qword ptr [rip + 0x12e3a]
0x17b16:    mov     rdi, rax
0x17b19:    call    0x48b0
0x17b1e:    mov     rdi, rbp
0x17b21:    call    0x17230
0x17b26:    test    eax, eax
0x17b28:    je      0x17b50
0x17b2a:    mov     rdi, rbp
0x17b2d:    mov     r12d, 1
0x17b33:    call    0x18d10
0x17b38:    add     rsp, 8
0x17b3c:    mov     eax, r12d
0x17b3f:    pop     rbx
0x17b40:    pop     rbp
0x17b41:    pop     r12
0x17b43:    pop     r13
0x17b45:    ret
0x17b46:    nop     word ptr cs:[rax + rax]
0x17b50:    mov     rdi, rbp
0x17b53:    lea     r13, [rip + 0x12b86]
0x17b5a:    call    0x18d10
0x17b5f:    mov     rdi, qword ptr [rip + 0x13f42]
0x17b66:    call    0x49b0
0x17b6b:    mov     rdi, r13
0x17b6e:    mov     rbp, rax
0x17b71:    call    0x49b0
0x17b76:    mov     rdi, qword ptr [rip + 0x12dd3]
0x17b7d:    mov     rbx, rax
0x17b80:    call    0x49b0

```



```

0x17b85:    lea     rdi, [rbp + rbx + 2]
0x17b8a:    add     rdi, rax
0x17b8d:    call    0x1d670
0x17b92:    mov     rsi, qword ptr [rip + 0x13f0f]
0x17b99:    mov     rdi, rax
0x17b9c:    mov     rbp, rax
0x17b9f:    call    0x4980
0x17ba4:    mov     rsi, r13
0x17ba7:    mov     rdi, rax
0x17baa:    call    0x4980
0x17baf:    mov     rsi, qword ptr [rip + 0x12d9a]
0x17bb6:    mov     rdi, rax
0x17bb9:    call    0x48b0
0x17bbe:    mov     rdi, rbp
0x17bc1:    call    0x17230
0x17bc6:    mov     r12d, eax
0x17bc9:    test    eax, eax
0x17bcb:    jne     0x17b2a
0x17bd1:    xor     eax, eax
0x17bd3:    mov     rdi, rbp
0x17bd6:    call    0x17880
0x17bdb:    mov     rdi, rbp
0x17bde:    call    0x18d10
0x17be3:    mov     rdi, qword ptr [rip + 0x13ebe]
0x17bea:    call    0x49b0
0x17bef:    mov     rdi, r13
0x17bf2:    mov     rbx, rax
0x17bf5:    call    0x49b0
0x17bfa:    lea     rdi, [rbx + rax + 0xa]
0x17bff:    call    0x1d670
0x17c04:    mov     rsi, qword ptr [rip + 0x13e9d]
0x17c0b:    mov     rdi, rax
0x17c0e:    mov     rbp, rax
0x17c11:    call    0x4980
0x17c16:    mov     rsi, r13
0x17c19:    mov     rdi, rax
0x17c1c:    call    0x4980
0x17c21:    mov     rdi, rbp
0x17c24:    movabs  rdx, 0x636972656e6567
0x17c2e:    mov     qword ptr [rax], rdx
0x17c31:    call    0x17230
0x17c36:    test    eax, eax
0x17c38:    jne     0x17b38
0x17c3e:    mov     rdi, rbp
0x17c41:    call    0x17830
0x17c46:    mov     edi, 1
0x17c4b:    call    0x4e20
0x17c50:    endbr64

```

```

0x17c54:    push    rbp
0x17c55:    push    rbx
0x17c56:    mov     rbx, rdi
0x17c59:    sub     rsp, 8
0x17c5d:    call    0x172a0
0x17c62:    cmp     eax, -1
0x17c65:    je      0x17c70
0x17c67:    add     rsp, 8
0x17c6b:    pop     rbx
0x17c6c:    pop     rbp
0x17c6d:    ret
0x17c6e:    nop
0x17c70:    mov     edx, 7
0x17c75:    lea     rsi, [rip + 0xa2e5]
0x17c7c:    mov     rdi, rbx
0x17c7f:    call    0x4890
0x17c84:    test    eax, eax
0x17c86:    jne     0x17cd4
0x17c88:    lea     rdi, [rbx + 5]
0x17c8c:    call    0x49b0
0x17c91:    lea     rdi, [rax + 6]
0x17c95:    call    0x1d670
0x17c9a:    lea     rsi, [rbx + 7]
0x17c9e:    mov     dword ptr [rax], 0x5449475b
0x17ca4:    mov     rbp, rax
0x17ca7:    lea     rdi, [rax + 5]
0x17cab:    mov     byte ptr [rax + 4], 0x2d
0x17caf:    call    0x48b0
0x17cb4:    mov     rdi, rbp
0x17cb7:    call    0x172a0
0x17cbc:    mov     rdi, rbp
0x17cbf:    cmp     eax, -1
0x17cc2:    je      0x17ccf
0x17cc4:    add     rsp, 8
0x17cc8:    pop     rbx
0x17cc9:    pop     rbp
0x17cca:    jmp     0x18d10
0x18d10:
0x17ccf:    call    0x18d10
0x17cd4:    mov     rdi, qword ptr [rip + 0x12ba5]
0x17cdb:    mov     rcx, qword ptr [rip + 0x12c06]
0x17ce2:    mov     r8, rbx
0x17ce5:    xor     eax, eax
0x17ce7:    lea     rdx, [rip + 0xa5ba]
0x17cee:    mov     esi, 1
0x17cf3:    call    0x4e60
0x17cf8:    mov     edi, 1
0x17cfd:    call    0x4e20

```

```

0x17d02:      nop      word ptr cs:[rax + rax]
0x17d0d:      nop      dword ptr [rax]
0x17d10:      endbr64
0x17d14:      push     rbx
0x17d15:      mov      ecx, 1
0x17d1a:      mov      ebx, esi
0x17d1c:      mov      edx, 1
0x17d21:      sub      rsp, 0x10
0x17d25:      mov      rax, qword ptr fs:[0x28]
0x17d2e:      mov      qword ptr [rsp + 8], rax
0x17d33:      xor      eax, eax
0x17d35:      mov      rsi, rsp
0x17d38:      call     0x17320
0x17d3d:      mov      rdi, qword ptr [rsp]
0x17d41:      mov      eax, ebx
0x17d43:      test     rdi, rdi
0x17d46:      je       0x17d54
0x17d48:      mov      edx, 0xa
0x17d4d:      xor      esi, esi
0x17d4f:      call     0x4ba0
0x17d54:      mov      rdx, qword ptr [rsp + 8]
0x17d59:      sub      rdx, qword ptr fs:[0x28]
0x17d62:      jne      0x17d6a
0x17d64:      add      rsp, 0x10
0x17d68:      pop      rbx
0x17d69:      ret
0x17d6a:      call     0x49d0
0x17d6f:      nop
0x17d70:      endbr64
0x17d74:      push     r15
0x17d76:      push     r14
0x17d78:      mov      r14, rdi
0x17d7b:      push     r13
0x17d7d:      mov      r13d, ecx
0x17d80:      mov      ecx, 1
0x17d85:      push     r12
0x17d87:      mov      r12d, r13d
0x17d8a:      push     rbp
0x17d8b:      mov      ebp, edx
0x17d8d:      mov      edx, 1
0x17d92:      push     rbx
0x17d93:      mov      rbx, rsi
0x17d96:      sub      rsp, 0x18
0x17d9a:      mov      rax, qword ptr fs:[0x28]
0x17da3:      mov      qword ptr [rsp + 8], rax
0x17da8:      xor      eax, eax
0x17daa:      mov      rsi, rsp
0x17dad:      call     0x17320

```

```

0x17db2:      mov     r15, qword ptr [rsp]
0x17db6:      test    r15, r15
0x17db9:      je      0x17de4
0x17dbb:      test    ebp, ebp
0x17dbd:      jle     0x17e3c
0x17dbf:      xor     r12d, r12d
0x17dc2:      jmp     0x17dd5
0x17dd5:
0x17dc4:      nop     dword ptr [rax]
0x17dc8:      add     r12d, 1
0x17dcc:      add     rbx, 8
0x17dd0:      cmp     ebp, r12d
0x17dd3:      je      0x17e10
0x17dd5:      mov     rsi, qword ptr [rbx]
0x17dd8:      mov     rdi, r15
0x17ddb:      call    0x4b30
0x17de0:      test    eax, eax
0x17de2:      jne     0x17dc8
0x17de4:      mov     rax, qword ptr [rsp + 8]
0x17de9:      sub     rax, qword ptr fs:[0x28]
0x17df2:      jne     0x17e43
0x17df4:      add     rsp, 0x18
0x17df8:      mov     eax, r12d
0x17dfb:      pop     rbx
0x17dfc:      pop     rbp
0x17dfd:      pop     r12
0x17dff:      pop     r13
0x17e01:      pop     r14
0x17e03:      pop     r15
0x17e05:      ret
0x17e06:      nop     word ptr cs:[rax + rax]
0x17e10:      mov     rcx, qword ptr [rip + 0x12ad1]
0x17e17:      mov     r9, r15
0x17e1a:      mov     r8, r14
0x17e1d:      xor     eax, eax
0x17e1f:      mov     rdi, qword ptr [rip + 0x12a5a]
0x17e26:      lea     rdx, [rip + 0xa13c]
0x17e2d:      mov     esi, 1
0x17e32:      mov     r12d, r13d
0x17e35:      call    0x4e60
0x17e3a:      jmp     0x17de4
0x17de4:
0x17e3c:      je      0x17e10
0x17e3e:      xor     r12d, r12d
0x17e41:      jmp     0x17de4
0x17de4:
0x17e43:      call    0x49d0
0x17e48:      nop     dword ptr [rax + rax]

```

```

0x17e50:      endbr64
0x17e54:      push    r12
0x17e56:      mov     ecx, 1
0x17e5b:      mov     edx, 1
0x17e60:      mov     r12, rdi
0x17e63:      push    rbp
0x17e64:      push    rbx
0x17e65:      mov     ebx, esi
0x17e67:      sub     rsp, 0x10
0x17e6b:      mov     rax, qword ptr fs:[0x28]
0x17e74:      mov     qword ptr [rsp + 8], rax
0x17e79:      xor     eax, eax
0x17e7b:      mov     rsi, rsp
0x17e7e:      call    0x17320
0x17e83:      mov     rbp, qword ptr [rsp]
0x17e87:      mov     eax, ebx
0x17e89:      test    rbp, rbp
0x17e8c:      je      0x17ea7
0x17e8e:      cmp     byte ptr [rbp], 0x4f
0x17e92:      je      0x17ec0
0x17e94:      lea     rsi, [rip + 0x8e4b]
0x17e9b:      mov     rdi, rbp
0x17e9e:      call    0x4b30
0x17ea3:      test    eax, eax
0x17ea5:      jne     0x17ed8
0x17ea7:      mov     rdx, qword ptr [rsp + 8]
0x17eac:      sub     rdx, qword ptr fs:[0x28]
0x17eb5:      jne     0x17f03
0x17eb7:      add     rsp, 0x10
0x17ebb:      pop     rbx
0x17ebc:      pop     rbp
0x17ebd:      pop     r12
0x17ebf:      ret
0x17ec0:      cmp     byte ptr [rbp + 1], 0x4e
0x17ec4:      jne     0x17e94
0x17ec6:      cmp     byte ptr [rbp + 2], 0
0x17eca:      mov     eax, 1
0x17ecf:      je      0x17ea7
0x17ed1:      jmp     0x17e94
0x17e94:
0x17ed3:      nop     dword ptr [rax + rax]
0x17ed8:      mov     rcx, qword ptr [rip + 0x12a09]
0x17edf:      mov     r9, rbp
0x17ee2:      mov     r8, r12
0x17ee5:      xor     eax, eax
0x17ee7:      mov     rdi, qword ptr [rip + 0x12992]
0x17eee:      lea     rdx, [rip + 0xa074]
0x17ef5:      mov     esi, 1

```

```

0x17efa:      call    0x4e60
0x17eff:      mov     eax, ebx
0x17f01:      jmp     0x17ea7
0x17ea7:
0x17f03:      call    0x49d0
0x17f08:      nop     dword ptr [rax + rax]
0x17f10:      endbr64
0x17f14:      push    r12
0x17f16:      mov     ecx, 1
0x17f1b:      mov     r12, rsi
0x17f1e:      mov     edx, 1
0x17f23:      sub     rsp, 0x10
0x17f27:      mov     rax, qword ptr fs:[0x28]
0x17f30:      mov     qword ptr [rsp + 8], rax
0x17f35:      xor     eax, eax
0x17f37:      mov     rsi, rsp
0x17f3a:      call    0x17320
0x17f3f:      mov     rdi, qword ptr [rsp]
0x17f43:      test    rdi, rdi
0x17f46:      je      0x17f50
0x17f48:      call    0x1d7a0
0x17f4d:      mov     r12, rax
0x17f50:      mov     rax, qword ptr [rsp + 8]
0x17f55:      sub     rax, qword ptr fs:[0x28]
0x17f5e:      jne     0x17f6a
0x17f60:      add     rsp, 0x10
0x17f64:      mov     rax, r12
0x17f67:      pop     r12
0x17f69:      ret
0x17f6a:      call    0x49d0
0x17f6f:      nop
0x17f70:      endbr64
0x17f74:      push    r14
0x17f76:      push    r13
0x17f78:      push    r12
0x17f7a:      push    rbp
0x17f7b:      push    rbx
0x17f7c:      sub     rsp, 0x10
0x17f80:      mov     rax, qword ptr fs:[0x28]
0x17f89:      mov     qword ptr [rsp + 8], rax
0x17f8e:      xor     eax, eax
0x17f90:      test    edx, edx
0x17f92:      jle     0x17fe8
0x17f94:      lea     eax, [rdx - 1]
0x17f97:      mov     rbx, rsi
0x17f9a:      mov     rbp, rdi
0x17f9d:      mov     r12, rsp
0x17fa0:      lea     r13, [rsi + rax*8 + 8]

```

```

0x17fa5:    lea     r14, [rip + 0x9fbd]
0x17fac:    nop     dword ptr [rax]
0x17fb0:    mov     rdi, qword ptr [rbx]
0x17fb3:    mov     ecx, 1
0x17fb8:    mov     edx, 1
0x17fbd:    mov     rsi, r12
0x17fc0:    call    0x17320
0x17fc5:    mov     rdi, qword ptr [rsp]
0x17fc9:    test    rdi, rdi
0x17fcc:    je      0x17fdb
0x17fce:    call    0xbd00
0x17fd3:    cmp     eax, -1
0x17fd6:    je      0x18008
0x17fd8:    mov     dword ptr [rbp], eax
0x17fdb:    add     rbx, 8
0x17fdf:    add     rbp, 4
0x17fe3:    cmp     rbx, r13
0x17fe6:    jne     0x17fb0
0x17fe8:    mov     rax, qword ptr [rsp + 8]
0x17fed:    sub     rax, qword ptr fs:[0x28]
0x17ff6:    jne     0x1802e
0x17ff8:    add     rsp, 0x10
0x17ffc:    pop     rbx
0x17ffd:    pop     rbp
0x17ffe:    pop     r12
0x18000:    pop     r13
0x18002:    pop     r14
0x18004:    ret
0x18005:    nop     dword ptr [rax]
0x18008:    mov     r9, qword ptr [rsp]
0x1800c:    mov     r8, qword ptr [rbx]
0x1800f:    mov     rdx, r14
0x18012:    mov     esi, 1
0x18017:    mov     rcx, qword ptr [rip + 0x128ca]
0x1801e:    mov     rdi, qword ptr [rip + 0x1285b]
0x18025:    xor     eax, eax
0x18027:    call    0x4e60
0x1802c:    jmp     0x17fdb
0x17fdb
0x1802e:    call    0x49d0
0x18033:    nop     word ptr cs:[rax + rax]
0x1803e:    nop
0x18040:    endbr64
0x18044:    push    r12
0x18046:    cmp     byte ptr [rdi], 0x2f
0x18049:    mov     r12, rdi
0x1804c:    lea     rax, [rdi + 1]
0x18050:    je      0x18061

```

```

0x18052:    lea     rdi, [rip + 0x9f26]
0x18059:    call   0x8b00
0x1805e:    mov     rax, r12
0x18061:    movzx   edx, byte ptr [rax]
0x18064:    mov     rsi, rax
0x18067:    test    dl, dl
0x18069:    jne     0x1808c
0x1806b:    jmp     0x180f0
0x180f0:
0x18070:    cmp     dl, 0x2e
0x18073:    jne     0x18118
0x18079:    movzx   ecx, byte ptr [rax + 1]
0x1807d:    cmp     cl, 0x2f
0x18080:    je      0x18086
0x18082:    test    cl, cl
0x18084:    jne     0x180a0
0x18086:    add     rax, 1
0x1808a:    mov     edx, ecx
0x1808c:    cmp     dl, 0x2f
0x1808f:    jne     0x18070
0x18091:    movzx   ecx, byte ptr [rax + 1]
0x18095:    jmp     0x18086
0x18086:
0x18097:    nop     word ptr [rax + rax]
0x180a0:    cmp     cl, 0x2e
0x180a3:    jne     0x180c0
0x180a5:    movzx   edx, byte ptr [rax + 2]
0x180a9:    cmp     dl, 0x2f
0x180ac:    je      0x18120
0x180ae:    test    dl, dl
0x180b0:    je      0x18120
0x180b2:    mov     edx, 0x2e
0x180b7:    nop     word ptr [rax + rax]
0x180c0:    mov     rdi, rax
0x180c3:    mov     rcx, rsi
0x180c6:    add     rax, 1
0x180ca:    add     rsi, 1
0x180ce:    mov     byte ptr [rsi - 1], dl
0x180d1:    movzx   edx, byte ptr [rax]
0x180d4:    test    dl, dl
0x180d6:    je      0x180f0
0x180d8:    cmp     dl, 0x2f
0x180db:    jne     0x180c0
0x180dd:    test    dl, dl
0x180df:    je      0x180f0
0x180e1:    mov     byte ptr [rsi], 0x2f
0x180e4:    movzx   edx, byte ptr [rdi + 1]
0x180e8:    lea     rsi, [rcx + 2]

```



```

0x180ec:    test    dl, dl
0x180ee:    jne     0x1808c
0x180f0:    cmp     byte ptr [rsi - 1], 0x2f
0x180f4:    jne     0x18109
0x180f6:    mov     rax, rsi
0x180f9:    sub     rax, r12
0x180fc:    cmp     rax, 1
0x18100:    setg    al
0x18103:    movzx   eax, al
0x18106:    sub     rsi, rax
0x18109:    mov     rax, r12
0x1810c:    mov     byte ptr [rsi], 0
0x1810f:    pop     r12
0x18111:    ret
0x18112:    nop     word ptr [rax + rax]
0x18118:    test    dl, dl
0x1811a:    jne     0x180c0
0x1811c:    jmp     0x180f0
0x180f0:
0x1811e:    nop
0x18120:    lea     rdi, [rsi - 1]
0x18124:    cmp     rdi, r12
0x18127:    jne     0x1812f
0x18129:    add     rax, 2
0x1812d:    jmp     0x180ec
0x180ec:
0x1812f:    cmp     byte ptr [rsi - 2], 0x2f
0x18133:    lea     rcx, [rsi - 2]
0x18137:    je      0x1814e
0x18139:    nop     dword ptr [rax]
0x18140:    mov     rsi, rcx
0x18143:    sub     rcx, 1
0x18147:    cmp     byte ptr [rcx], 0x2f
0x1814a:    jne     0x18140
0x1814c:    jmp     0x18129
0x18129:
0x1814e:    mov     rsi, rdi
0x18151:    jmp     0x18129
0x18129:
0x18153:    nop     word ptr cs:[rax + rax]
0x1815e:    nop
0x18160:    endbr64
0x18164:    push    rbx
0x18165:    call    0x4a60
0x1816a:    mov     edi, eax
0x1816c:    mov     ebx, eax
0x1816e:    call    0x4970
0x18173:    test    rax, rax

```

```

0x18176:      je      0x18189
0x18178:      mov     rdi, qword ptr [rax]
0x1817b:      call    0x1d7a0
0x18180:      pop     rbx
0x18181:      mov     qword ptr [rip + 0x13910], rax
0x18188:      ret
0x18189:      mov     rcx, qword ptr [rip + 0x12758]
0x18190:      mov     r8d, ebx
0x18193:      mov     rdi, qword ptr [rip + 0x126e6]
0x1819a:      lea     rdx, [rip + 0xa13f]
0x181a1:      mov     esi, 1
0x181a6:      call    0x4e60
0x181ab:      mov     rcx, qword ptr [rip + 0x12736]
0x181b2:      xor     eax, eax
0x181b4:      mov     rdi, qword ptr [rip + 0x126c5]
0x181bb:      lea     r8, [rip + 0x9dd7]
0x181c2:      lea     rdx, [rip + 0xa14f]
0x181c9:      mov     esi, 1
0x181ce:      call    0x4e60
0x181d3:      mov     rdi, qword ptr [rip + 0x126a6]
0x181da:      mov     esi, 1
0x181df:      xor     eax, eax
0x181e1:      mov     rcx, qword ptr [rip + 0x12700]
0x181e8:      lea     rdx, [rip + 0xa161]
0x181ef:      call    0x4e60
0x181f4:      mov     edi, 1
0x181f9:      call    0x4e20
0x181fe:      nop
0x18200:      endbr64
0x18204:      push    r12
0x18206:      push    rbp
0x18207:      mov     rbp, rsi
0x1820a:      push    rbx
0x1820b:      movsxd  rbx, edx
0x1820e:      cmp     ebx, 2
0x18211:      je      0x18250
0x18213:      mov     r12, rdi
0x18216:      jg      0x18230
0x18218:      test    ebx, ebx
0x1821a:      je      0x18224
0x1821c:      cmp     ebx, 1
0x1821f:      jne     0x18260
0x18221:      mov     byte ptr [rsi], 0x20
0x18224:      pop     rbx
0x18225:      pop     rbp
0x18226:      pop     r12
0x18228:      ret
0x18229:      nop     dword ptr [rax]

```

```

0x18230:      cmp     ebx, 3
0x18233:      jne     0x18260
0x18235:      mov     edx, 0x2020
0x1823a:      mov     byte ptr [rsi + 2], 0x20
0x1823e:      mov     word ptr [rsi], dx
0x18241:      pop     rbx
0x18242:      pop     rbp
0x18243:      pop     r12
0x18245:      ret
0x18246:      nop     word ptr cs:[rax + rax]
0x18250:      mov     ecx, 0x2020
0x18255:      mov     word ptr [rsi], cx
0x18258:      pop     rbx
0x18259:      pop     rbp
0x1825a:      pop     r12
0x1825c:      ret
0x1825d:      nop     dword ptr [rax]
0x18260:      mov     rdi, r12
0x18263:      call    0x49b0
0x18268:      movsxd  rdx, eax
0x1826b:      cmp     ebx, eax
0x1826d:      jge     0x182a0
0x1826f:      mov     eax, 0x2e2e
0x18274:      lea     r8d, [rbx - 3]
0x18278:      mov     byte ptr [rbp + 2], 0x2e
0x1827c:      sub     rdx, rbx
0x1827f:      mov     word ptr [rbp], ax
0x18283:      movsxd  r8, r8d
0x18286:      lea     rsi, [r12 + rdx + 3]
0x1828b:      pop     rbx
0x1828c:      lea     rdi, [rbp + 3]
0x18290:      mov     rdx, r8
0x18293:      pop     rbp
0x18294:      pop     r12
0x18296:      jmp     0x4bb0
0x4bb0
0x1829b:      nop     dword ptr [rax + rax]
0x182a0:      pop     rbx
0x182a1:      mov     rsi, r12
0x182a4:      mov     rdi, rbp
0x182a7:      pop     rbp
0x182a8:      pop     r12
0x182aa:      jmp     0x4bb0
0x4bb0
0x182af:      nop
0x182b0:      endbr64
0x182b4:      push    r12
0x182b6:      mov     r12, rsi

```

```

0x182b9:    push    rbp
0x182ba:    mov     rbp, rdi
0x182bd:    push    rbx
0x182be:    lea     ebx, [rdx - 1]
0x182c1:    mov     edx, ebx
0x182c3:    call    0x18200
0x182c8:    mov     rdi, rbp
0x182cb:    call    0x49b0
0x182d0:    movsxd  rcx, ebx
0x182d3:    cmp     ebx, eax
0x182d5:    movsxd  rdx, eax
0x182d8:    mov     rax, rcx
0x182db:    pop     rbx
0x182dc:    cmovg   rax, rdx
0x182e0:    pop     rbp
0x182e1:    mov     byte ptr [r12 + rax], 0
0x182e6:    mov     rax, r12
0x182e9:    pop     r12
0x182eb:    ret
0x182ec:    nop     dword ptr [rax]
0x182f0:    endbr64
0x182f4:    push    r13
0x182f6:    mov     edx, 1
0x182fb:    xor     esi, esi
0x182fd:    push    r12
0x182ff:    push    rbp
0x18300:    mov     ebp, edi
0x18302:    call    0x4e50
0x18307:    mov     edi, ebp
0x18309:    mov     edx, 2
0x1830e:    xor     esi, esi
0x18310:    mov     r13, rax
0x18313:    call    0x4e50
0x18318:    mov     rsi, r13
0x1831b:    mov     edi, ebp
0x1831d:    xor     edx, edx
0x1831f:    mov     r12, rax
0x18322:    call    0x4e50
0x18327:    mov     rax, r12
0x1832a:    pop     rbp
0x1832b:    pop     r12
0x1832d:    pop     r13
0x1832f:    ret
0x18330:    endbr64
0x18334:    sub     rsp, 0x18
0x18338:    xor     edi, edi
0x1833a:    mov     rax, qword ptr fs:[0x28]
0x18343:    mov     qword ptr [rsp + 8], rax

```

```

0x18348:    xor     eax, eax
0x1834a:    call   0x4be0
0x1834f:    mov     rdi, rsp
0x18352:    mov     qword ptr [rsp], rax
0x18356:    call   0x4840
0x1835b:    mov     rdx, qword ptr [rsp + 8]
0x18360:    sub     rdx, qword ptr fs:[0x28]
0x18369:    jne     0x18370
0x1836b:    add     rsp, 0x18
0x1836f:    ret
0x18370:    call   0x49d0
0x18375:    nop     word ptr cs:[rax + rax]
0x18380:    endbr64
0x18384:    push    r12
0x18386:    push    rbp
0x18387:    mov     rbp, rsi
0x1838a:    push    rbx
0x1838b:    mov     rbx, rdi
0x1838e:    call   0x49b0
0x18393:    mov     rdi, rbp
0x18396:    mov     r12, rax
0x18399:    call   0x49b0
0x1839e:    lea     rdi, [r12 + rax + 2]
0x183a3:    call   0x1d670
0x183a8:    mov     r9, rbp
0x183ab:    mov     r8, rbx
0x183ae:    mov     esi, 1
0x183b3:    mov     r12, rax
0x183b6:    mov     rdi, rax
0x183b9:    xor     eax, eax
0x183bb:    mov     rdx, -1
0x183c2:    lea     rcx, [rip + 0x9be5]
0x183c9:    call   0x4f10
0x183ce:    mov     rdi, r12
0x183d1:    call   0x4b20
0x183d6:    mov     r12d, eax
0x183d9:    cmp     eax, -1
0x183dc:    je      0x183f0
0x183de:    mov     eax, r12d
0x183e1:    pop     rbx
0x183e2:    pop     rbp
0x183e3:    pop     r12
0x183e5:    ret
0x183e6:    nop     word ptr cs:[rax + rax]
0x183f0:    mov     rcx, qword ptr [rip + 0x124f1]
0x183f7:    mov     rdi, qword ptr [rip + 0x12482]
0x183fe:    mov     r8, rbx
0x18401:    xor     eax, eax

```

```

0x18403:    lea     rdx, [rip + 0x9f6e]
0x1840a:    mov     esi, 1
0x1840f:    call    0x4e60
0x18414:    mov     eax, r12d
0x18417:    pop     rbx
0x18418:    pop     rbp
0x18419:    pop     r12
0x1841b:    ret
0x1841c:    nop     dword ptr [rax]
0x18420:    endbr64
0x18424:    push    r13
0x18426:    push    r12
0x18428:    lea     r12, [rip + 0x9b85]
0x1842f:    push    rbp
0x18430:    mov     rdi, r12
0x18433:    push    rbx
0x18434:    sub     rsp, 8
0x18438:    call    0x47e0
0x1843d:    mov     r13, qword ptr [rip + 0x1365c]
0x18444:    test    rax, rax
0x18447:    je      0x184b0
0x18449:    mov     rdi, rax
0x1844c:    mov     rbx, rax
0x1844f:    call    0x49b0
0x18454:    mov     rdi, r13
0x18457:    mov     rbp, rax
0x1845a:    call    0x49b0
0x1845f:    lea     rdi, [rbp + rax + 2]
0x18464:    call    0x1d670
0x18469:    mov     r9, qword ptr [rip + 0x13630]
0x18470:    mov     r8, rbx
0x18473:    lea     rcx, [rip + 0x9b3f]
0x1847a:    mov     rbp, rax
0x1847d:    mov     rdi, rax
0x18480:    mov     esi, 1
0x18485:    xor     eax, eax
0x18487:    mov     rdx, -1
0x1848e:    call    0x4f10
0x18493:    mov     rdi, r12
0x18496:    mov     rsi, rbp
0x18499:    call    0x18380
0x1849e:    add     rsp, 8
0x184a2:    mov     rdi, rbp
0x184a5:    pop     rbx
0x184a6:    pop     rbp
0x184a7:    pop     r12
0x184a9:    pop     r13
0x184ab:    jmp     0x18d10

```

```

0x18d10
0x184b0:      add     rsp, 8
0x184b4:      mov     rsi, r13
0x184b7:      mov     rdi, r12
0x184ba:      pop     rbx
0x184bb:      pop     rbp
0x184bc:      pop     r12
0x184be:      pop     r13
0x184c0:      jmp     0x18380
0x18380
0x184c5:      nop     word ptr cs:[rax + rax]
0x184d0:      endbr64
0x184d4:      test    rsi, rsi
0x184d7:      je      0x18520
0x184d9:      push    rbp
0x184da:      mov     rbp, rsi
0x184dd:      push    rbx
0x184de:      mov     rbx, rdi
0x184e1:      sub     rsp, 8
0x184e5:      call    0x4f00
0x184ea:      mov     rdi, rbx
0x184ed:      lea     rsi, [rbx + rbp]
0x184f1:      nop     dword ptr [rax]
0x184f8:      movsx   rcx, byte ptr [rdi]
0x184fc:      mov     rdx, qword ptr [rax]
0x184ff:      test    byte ptr [rdx + rcx*2 + 1], 0x40
0x18504:      jne     0x18509
0x18506:      mov     byte ptr [rdi], 0x3f
0x18509:      add     rdi, 1
0x1850d:      cmp     rsi, rdi
0x18510:      jne     0x184f8
0x18512:      add     rsp, 8
0x18516:      pop     rbx
0x18517:      pop     rbp
0x18518:      ret
0x18519:      nop     dword ptr [rax]
0x18520:      ret
0x18521:      nop     word ptr cs:[rax + rax]
0x1852c:      nop     dword ptr [rax]
0x18530:      endbr64
0x18534:      test    rsi, rsi
0x18537:      je      0x185c2
0x1853d:      push    rbp
0x1853e:      mov     rbp, rsi
0x18541:      push    rbx
0x18542:      mov     rbx, rdi
0x18545:      sub     rsp, 8
0x18549:      call    0x4f00

```

```

0x1854e:      mov     rdi, rbx
0x18551:      lea     rsi, [rbx + rbp]
0x18555:      movabs  r9, 0x68000000c0000001
0x1855f:      movabs  r8, 0x2400f82000000000
0x18569:      mov     rcx, qword ptr [rax]
0x1856c:      nop     dword ptr [rax]
0x18570:      movsx   rdx, byte ptr [rdi]
0x18574:      mov     rax, rdx
0x18577:      test    byte ptr [rcx + rdx*2], 8
0x1857b:      jne     0x185b0
0x1857d:      cmp     dl, 0x3d
0x18580:      jg      0x185a0
0x18582:      cmp     dl, 0x24
0x18585:      jle     0x1858d
0x18587:      bt      r8, rdx
0x1858b:      jb      0x185b0
0x1858d:      add     rsp, 8
0x18591:      mov     eax, 1
0x18596:      pop     rbx
0x18597:      pop     rbp
0x18598:      ret
0x18599:      nop     dword ptr [rax]
0x185a0:      sub     eax, 0x40
0x185a3:      cmp     al, 0x3e
0x185a5:      ja      0x1858d
0x185a7:      bt      r9, rax
0x185ab:      jae     0x1858d
0x185ad:      nop     dword ptr [rax]
0x185b0:      add     rdi, 1
0x185b4:      cmp     rsi, rdi
0x185b7:      jne     0x18570
0x185b9:      add     rsp, 8
0x185bd:      xor     eax, eax
0x185bf:      pop     rbx
0x185c0:      pop     rbp
0x185c1:      ret
0x185c2:      xor     eax, eax
0x185c4:      ret
0x185c5:      nop     word ptr cs:[rax + rax]
0x185d0:      endbr64
0x185d4:      push    rbx
0x185d5:      mov     rbx, rdi
0x185d8:      call    0x49b0
0x185dd:      sub     eax, 1
0x185e0:      js      0x18618
0x185e2:      cdqe
0x185e4:      jmp     0x18602
0x18602:

```



```

0x185e6:      nop      word ptr cs:[rax + rax]
0x185f0:      cmp      dl, 0x20
0x185f3:      je       0x185fa
0x185f5:      cmp      dl, 9
0x185f8:      jne      0x18618
0x185fa:      sub      rax, 1
0x185fe:      test     eax, eax
0x18600:      js       0x18618
0x18602:      movzx    edx, byte ptr [rbx + rax]
0x18606:      cmp      dl, 0x26
0x18609:      jne      0x185f0
0x1860b:      mov      eax, 1
0x18610:      pop      rbx
0x18611:      ret
0x18612:      nop      word ptr [rax + rax]
0x18618:      xor      eax, eax
0x1861a:      pop      rbx
0x1861b:      ret
0x1861c:      nop      dword ptr [rax]
0x18620:      endbr64
0x18624:      jmp      0x1863c
0x1863c:
0x18626:      nop      word ptr cs:[rax + rax]
0x18630:      cmp      al, 0x20
0x18632:      je       0x18638
0x18634:      cmp      al, 0x3b
0x18636:      jne      0x18650
0x18638:      add      rdi, 1
0x1863c:      movzx    eax, byte ptr [rdi]
0x1863f:      test     al, al
0x18641:      jne      0x18630
0x18643:      mov      eax, 1
0x18648:      ret
0x18649:      nop      dword ptr [rax]
0x18650:      xor      eax, eax
0x18652:      ret
0x18653:      nop      word ptr cs:[rax + rax]
0x1865e:      nop
0x18660:      endbr64
0x18664:      push     r15
0x18666:      push     r14
0x18668:      push     r13
0x1866a:      push     r12
0x1866c:      push     rbp
0x1866d:      push     rbx
0x1866e:      sub      rsp, 0x98
0x18675:      mov      rax, qword ptr fs:[0x28]
0x1867e:      mov      qword ptr [rsp + 0x88], rax

```

```

0x18686:    xor     eax, eax
0x18688:    lea     r15, [rsp + 0x10]
0x1868d:    lea     r13, [rsp + 0x30]
0x18692:    mov     qword ptr [rsp + 8], 0
0x1869b:    mov     qword ptr [rsp], 0
0x186a3:    jmp     0x186d3
0x186d3:
0x186a5:    nop     dword ptr [rax]
0x186a8:    mov     rax, qword ptr [rsp]
0x186ac:    mov     qword ptr [rax + 0x18], r14
0x186b0:    mov     rdi, r13
0x186b3:    call    0x1d7a0
0x186b8:    mov     dword ptr [r14 + 8], r12d
0x186bc:    mov     qword ptr [r14], rax
0x186bf:    mov     dword ptr [r14 + 0xc], ebp
0x186c3:    mov     dword ptr [r14 + 0x10], ebx
0x186c7:    mov     qword ptr [r14 + 0x18], 0
0x186cf:    mov     qword ptr [rsp], r14
0x186d3:    xor     ecx, ecx
0x186d5:    mov     edx, 3
0x186da:    mov     rsi, r15
0x186dd:    mov     rdi, r13
0x186e0:    call    0x17320
0x186e5:    cmp     byte ptr [rsp + 0x30], 0
0x186ea:    je      0x18758
0x186ec:    mov     rdi, qword ptr [rsp + 0x10]
0x186f1:    mov     r12d, 0xffffffff
0x186f7:    test    rdi, rdi
0x186fa:    je      0x18704
0x186fc:    call    0xbd00
0x18701:    mov     r12d, eax
0x18704:    mov     rdi, qword ptr [rsp + 0x18]
0x18709:    mov     ebp, 0xffffffff
0x1870e:    test    rdi, rdi
0x18711:    je      0x1871a
0x18713:    call    0xbd00
0x18718:    mov     ebp, eax
0x1871a:    mov     rdi, qword ptr [rsp + 0x20]
0x1871f:    mov     ebx, 0xffffffff
0x18724:    test    rdi, rdi
0x18727:    je      0x18730
0x18729:    call    0xbd00
0x1872e:    mov     ebx, eax
0x18730:    mov     edi, 0x20
0x18735:    call    0x1d670
0x1873a:    cmp     qword ptr [rsp + 8], 0
0x18740:    mov     r14, rax
0x18743:    jne     0x186a8

```

```

0x18749:      mov     qword ptr [rsp + 8], rax
0x1874e:      jmp     0x186b0
0x186b0:
0x18753:      nop     dword ptr [rax + rax]
0x18758:      mov     rdx, qword ptr [rsp + 8]
0x1875d:      test    rdx, rdx
0x18760:      je      0x1877d
0x18762:      mov     rax, qword ptr [rip + 0x13327]
0x18769:      test    rax, rax
0x1876c:      je      0x187a8
0x1876e:      mov     rcx, qword ptr [rsp]
0x18772:      mov     qword ptr [rip + 0x13317], rdx
0x18779:      mov     qword ptr [rcx + 0x18], rax
0x1877d:      mov     rax, qword ptr [rsp + 0x88]
0x18785:      sub     rax, qword ptr fs:[0x28]
0x1878e:      jne     0x187b6
0x18790:      add     rsp, 0x98
0x18797:      pop     rbx
0x18798:      pop     rbp
0x18799:      pop     r12
0x1879b:      pop     r13
0x1879d:      pop     r14
0x1879f:      pop     r15
0x187a1:      ret
0x187a2:      nop     word ptr [rax + rax]
0x187a8:      mov     rax, qword ptr [rsp + 8]
0x187ad:      mov     qword ptr [rip + 0x132dc], rax
0x187b4:      jmp     0x1877d
0x1877d:
0x187b6:      call    0x49d0
0x187bb:      nop     dword ptr [rax + rax]
0x187c0:      endbr64
0x187c4:      sub     rsp, 8
0x187c8:      lea     rsi, [rip - 0xf]
0x187cf:      call    0x4b50
0x187d4:      mov     dword ptr [rip + 0x132d2], 1
0x187de:      add     rsp, 8
0x187e2:      ret
0x187e3:      nop     word ptr cs:[rax + rax]
0x187ee:      nop
0x187f0:      cmp     edi, 0x14
0x187f3:      je      0x18850
0x187f5:      jg      0x18818
0x187f7:      cmp     edi, 0xe
0x187fa:      je      0x18838
0x187fc:      cmp     edi, 0x12
0x187ff:      jne     0x18861
0x18801:      lea     rsi, [rip + 0x148]

```

```

0x18808:      mov     edi, 0x12
0x1880d:      jmp     0x4b50
0x4b50
0x18812:      nop     word ptr [rax + rax]
0x18818:      cmp     edi, 0x1c
0x1881b:      jne     0x18830
0x1881d:      lea     rsi, [rip + 0x1cc]
0x18824:      mov     edi, 0x1c
0x18829:      jmp     0x4b50
0x4b50
0x1882e:      nop
0x18830:      ret
0x18831:      nop     dword ptr [rax]
0x18838:      lea     rsi, [rip + 0x31]
0x1883f:      mov     edi, 0xe
0x18844:      jmp     0x4b50
0x4b50
0x18849:      nop     dword ptr [rax]
0x18850:      lea     rsi, [rip + 0x89]
0x18857:      mov     edi, 0x14
0x1885c:      jmp     0x4b50
0x4b50
0x18861:      ret
0x18862:      nop     word ptr cs:[rax + rax]
0x1886d:      nop     dword ptr [rax]
0x18870:      endbr64
0x18874:      mov     edx, dword ptr [rip + 0x1323a]
0x1887a:      push    rbp
0x1887b:      mov     eax, 1
0x18880:      mov     ebp, edi
0x18882:      test    edx, edx
0x18884:      jne     0x188b0
0x18886:      mov     edi, ebp
0x18888:      mov     dword ptr [rip + 0x1322e], eax
0x1888e:      call    0x187f0
0x18893:      xor     eax, eax
0x18895:      call    0x18330
0x1889a:      mov     edi, 0x3c
0x1889f:      pop     rbp
0x188a0:      sub     edi, dword ptr [rax]
0x188a2:      jmp     0x4aa0
0x4aa0
0x188a7:      nop     word ptr [rax + rax]
0x188b0:      xor     eax, eax
0x188b2:      call    0x18330
0x188b7:      mov     eax, dword ptr [rax]
0x188b9:      test    eax, eax
0x188bb:      je      0x188d0

```

```

0x188bd:    xor     eax, eax
0x188bf:    call    0xb4e0
0x188c4:    xor     eax, eax
0x188c6:    jmp     0x18886
0x18886:
0x188c8:    nop     dword ptr [rax + rax]
0x188d0:    mov     edi, ebp
0x188d2:    call    0x16fd0
0x188d7:    jmp     0x188bd
0x188bd:
0x188d9:    nop     dword ptr [rax]
0x188e0:    endbr64
0x188e4:    mov     eax, dword ptr [rip + 0x131ca]
0x188ea:    push    rbp
0x188eb:    mov     ebp, edi
0x188ed:    test    eax, eax
0x188ef:    jne     0x18908
0x188f1:    mov     eax, 1
0x188f6:    mov     edi, ebp
0x188f8:    pop     rbp
0x188f9:    mov     dword ptr [rip + 0x131c5], eax
0x188ff:    jmp     0x187f0
0x187f0:
0x18904:    nop     dword ptr [rax]
0x18908:    xor     eax, eax
0x1890a:    call    0x9c90
0x1890f:    cmp     eax, 1
0x18912:    sete    al
0x18915:    movzx   eax, al
0x18918:    mov     dword ptr [rip + 0x1319a], eax
0x1891e:    xor     eax, eax
0x18920:    call    0x9610
0x18925:    call    0x4920
0x1892a:    mov     esi, 0x13
0x1892f:    mov     edi, eax
0x18931:    call    0x4bd0
0x18936:    xor     eax, eax
0x18938:    mov     edi, ebp
0x1893a:    pop     rbp
0x1893b:    mov     dword ptr [rip + 0x13183], eax
0x18941:    jmp     0x187f0
0x187f0:
0x18946:    nop     word ptr cs:[rax + rax]
0x18950:    endbr64
0x18954:    mov     eax, dword ptr [rip + 0x1315e]
0x1895a:    push    rbx
0x1895b:    mov     ebx, edi
0x1895d:    test    eax, eax

```

```

0x1895f:      jne      0x189a0
0x18961:      cmp      ebx, 0x14
0x18964:      je       0x189b4
0x18966:      jg       0x18988
0x18968:      cmp      ebx, 0xe
0x1896b:      je       0x189d8
0x1896d:      cmp      ebx, 0x12
0x18970:      jne      0x189d0
0x18972:      lea      rsi, [rip - 0x29]
0x18979:      mov      edi, 0x12
0x1897e:      pop      rbx
0x1897f:      jmp      0x4b50
0x4b50
0x18984:      nop      dword ptr [rax]
0x18988:      cmp      ebx, 0x1c
0x1898b:      jne      0x189d0
0x1898d:      lea      rsi, [rip + 0x5c]
0x18994:      mov      edi, 0x1c
0x18999:      pop      rbx
0x1899a:      jmp      0x4b50
0x4b50
0x1899f:      nop
0x189a0:      call     0x7bd0
0x189a5:      mov      dword ptr [rip + 0x13111], 0
0x189af:      cmp      ebx, 0x14
0x189b2:      jne      0x18966
0x189b4:      lea      rsi, [rip - 0xdb]
0x189bb:      mov      edi, 0x14
0x189c0:      pop      rbx
0x189c1:      jmp      0x4b50
0x4b50
0x189c6:      nop      word ptr cs:[rax + rax]
0x189d0:      pop      rbx
0x189d1:      ret
0x189d2:      nop      word ptr [rax + rax]
0x189d8:      lea      rsi, [rip - 0x16f]
0x189df:      mov      edi, 0xe
0x189e4:      pop      rbx
0x189e5:      jmp      0x4b50
0x4b50
0x189ea:      nop      word ptr [rax + rax]
0x189f0:      endbr64
0x189f4:      mov      eax, dword ptr [rip + 0x130ba]
0x189fa:      push     rbx
0x189fb:      mov      ebx, edi
0x189fd:      test     eax, eax
0x189ff:      jne      0x18a50
0x18a01:      mov      eax, 1

```

```

0x18a06:    mov     dword ptr [rip + 0x130b4], eax
0x18a0c:    cmp     ebx, 0x14
0x18a0f:    je      0x18a72
0x18a11:    jg      0x18a30
0x18a13:    cmp     ebx, 0xe
0x18a16:    je      0x18a90
0x18a18:    cmp     ebx, 0x12
0x18a1b:    jne     0x18a88
0x18a1d:    lea     rsi, [rip - 0xd4]
0x18a24:    mov     edi, 0x12
0x18a29:    pop     rbx
0x18a2a:    jmp     0x4b50
0x4b50:
0x18a2f:    nop
0x18a30:    cmp     ebx, 0x1c
0x18a33:    jne     0x18a88
0x18a35:    lea     rsi, [rip - 0x4c]
0x18a3c:    mov     edi, 0x1c
0x18a41:    pop     rbx
0x18a42:    jmp     0x4b50
0x4b50:
0x18a47:    nop     word ptr [rax + rax]
0x18a50:    xor     eax, eax
0x18a52:    call    0xa6c0
0x18a57:    xor     eax, eax
0x18a59:    call    0x9d80
0x18a5e:    mov     edi, ebx
0x18a60:    call    0x7bd0
0x18a65:    xor     eax, eax
0x18a67:    mov     dword ptr [rip + 0x13053], eax
0x18a6d:    cmp     ebx, 0x14
0x18a70:    jne     0x18a11
0x18a72:    lea     rsi, [rip - 0x199]
0x18a79:    mov     edi, 0x14
0x18a7e:    pop     rbx
0x18a7f:    jmp     0x4b50
0x4b50:
0x18a84:    nop     dword ptr [rax]
0x18a88:    pop     rbx
0x18a89:    ret
0x18a8a:    nop     word ptr [rax + rax]
0x18a90:    lea     rsi, [rip - 0x227]
0x18a97:    mov     edi, 0xe
0x18a9c:    pop     rbx
0x18a9d:    jmp     0x4b50
0x4b50:
0x18aa2:    nop     word ptr cs:[rax + rax]
0x18aad:    nop     dword ptr [rax]

```

```

0x18ab0:    endbr64
0x18ab4:    sub     rsp, 8
0x18ab8:    mov     ecx, dword ptr [rip + 0x13006]
0x18abe:    test    ecx, ecx
0x18ac0:    jne     0x18af0
0x18ac2:    mov     edx, dword ptr [rip + 0x12ff8]
0x18ac8:    test    edx, edx
0x18aca:    jne     0x18b30
0x18acc:    mov     eax, dword ptr [rip + 0x12fea]
0x18ad2:    test    eax, eax
0x18ad4:    jne     0x18ae0
0x18ad6:    add     rsp, 8
0x18ada:    ret
0x18adb:    nop     dword ptr [rax + rax]
0x18ae0:    mov     dword ptr [rip + 0x12fd2], 0
0x18aea:    add     rsp, 8
0x18aee:    ret
0x18aef:    nop
0x18af0:    xor     eax, eax
0x18af2:    call    0x9610
0x18af7:    call    0x4920
0x18afc:    mov     esi, 0x13
0x18b01:    mov     edi, eax
0x18b03:    call    0x4bd0
0x18b08:    mov     dword ptr [rip + 0x12fb2], 0
0x18b12:    mov     dword ptr [rip + 0x12fa4], 0
0x18b1c:    mov     dword ptr [rip + 0x12f96], 0
0x18b26:    add     rsp, 8
0x18b2a:    ret
0x18b2b:    nop     dword ptr [rax + rax]
0x18b30:    xor     eax, eax
0x18b32:    call    0xa6c0
0x18b37:    xor     eax, eax
0x18b39:    call    0x9d80
0x18b3e:    mov     edi, 0x1c
0x18b43:    call    0x7bd0
0x18b48:    mov     dword ptr [rip + 0x12f6e], 0
0x18b52:    mov     dword ptr [rip + 0x12f60], 0
0x18b5c:    add     rsp, 8
0x18b60:    ret
0x18b61:    nop     word ptr cs:[rax + rax]
0x18b6c:    nop     dword ptr [rax]
0x18b70:    endbr64
0x18b74:    mov     dword ptr [rip + 0x12f3a], edi
0x18b7a:    test    edi, edi
0x18b7c:    jne     0x18b80
0x18b7e:    ret
0x18b7f:    nop

```



```

0x18b80:    xor     eax, eax
0x18b82:    jmp     0x18ab0
0x18ab0:
0x18b87:    nop     word ptr [rax + rax]
0x18b90:    endbr64
0x18b94:    sub     rsp, 8
0x18b98:    mov     eax, dword ptr [rip + 0x11bfa]
0x18b9e:    cmp     edi, 1
0x18ba1:    je      0x18bf0
0x18ba3:    test    eax, eax
0x18ba5:    jne     0x18bd0
0x18ba7:    mov     esi, 1
0x18bac:    mov     edi, 0x1c
0x18bb1:    call    0x4b50
0x18bb6:    mov     esi, 1
0x18bbb:    mov     edi, 0xe
0x18bc0:    add     rsp, 8
0x18bc4:    jmp     0x4b50
0x4b50:
0x18bc9:    nop     dword ptr [rax]
0x18bd0:    mov     esi, 1
0x18bd5:    mov     edi, 0x14
0x18bda:    call    0x4b50
0x18bdf:    mov     esi, 1
0x18be4:    mov     edi, 0x12
0x18be9:    call    0x4b50
0x18bee:    jmp     0x18ba7
0x18ba7:
0x18bf0:    test    eax, eax
0x18bf2:    jne     0x18c20
0x18bf4:    lea     rsi, [rip - 0x20b]
0x18bfb:    mov     edi, 0x1c
0x18c00:    call    0x4b50
0x18c05:    lea     rsi, [rip - 0x39c]
0x18c0c:    mov     edi, 0xe
0x18c11:    add     rsp, 8
0x18c15:    jmp     0x4b50
0x4b50:
0x18c1a:    nop     word ptr [rax + rax]
0x18c20:    lea     rsi, [rip - 0x347]
0x18c27:    mov     edi, 0x14
0x18c2c:    call    0x4b50
0x18c31:    lea     rsi, [rip - 0x2e8]
0x18c38:    mov     edi, 0x12
0x18c3d:    call    0x4b50
0x18c42:    jmp     0x18bf4
0x18bf4:
0x18c44:    nop     word ptr cs:[rax + rax]

```

```

0x18c4f:      nop
0x18c50:      endbr64
0x18c54:      push    rbp
0x18c55:      mov     esi, 1
0x18c5a:      mov     edi, 0x14
0x18c5f:      call    0x4b50
0x18c64:      mov     esi, 1
0x18c69:      mov     edi, 0x12
0x18c6e:      cmp     rax, 1
0x18c72:      setne   al
0x18c75:      movzx   eax, al
0x18c78:      mov     dword ptr [rip + 0x11b1a], eax
0x18c7e:      call    0x4b50
0x18c83:      mov     esi, 1
0x18c88:      mov     edi, 0x1c
0x18c8d:      call    0x4b50
0x18c92:      lea     rbp, [rip - 0x1519]
0x18c99:      mov     edi, 0xb
0x18c9e:      mov     rsi, rbp
0x18ca1:      call    0x4b50
0x18ca6:      mov     rsi, rbp
0x18ca9:      mov     edi, 1
0x18cae:      call    0x4b50
0x18cb3:      mov     rsi, rbp
0x18cb6:      mov     edi, 0xf
0x18cbb:      call    0x4b50
0x18cc0:      mov     esi, 1
0x18cc5:      mov     edi, 3
0x18cca:      call    0x4b50
0x18ccf:      lea     rsi, [rip - 0x516]
0x18cd6:      mov     edi, 2
0x18cdb:      pop     rbp
0x18cdc:      jmp     0x4b50
0x4b50:
0x18ce1:      nop     word ptr cs:[rax + rax]
0x18ceb:      nop     dword ptr [rax + rax]
0x18cf0:      endbr64
0x18cf4:      push    rax
0x18cf5:      pop     rax
0x18cf6:      lea     rdi, [rip + 0x96a8]
0x18cfd:      sub     rsp, 8
0x18d01:      call    0x8b00
0x18d06:      call    0x4850
0x18d0b:      nop     dword ptr [rax + rax]
0x18d10:      endbr64
0x18d14:      test    rdi, rdi
0x18d17:      je      0x18d20
0x18d19:      jmp     0x4800

```

```

0x4800
0x18d1e:      nop
0x18d20:      lea     rdi, [rip + 0x9697]
0x18d27:      jmp     0x8b00
0x8b00
0x18d2c:      nop     dword ptr [rax]
0x18d30:      endbr64
0x18d34:      xor     r8d, r8d
0x18d37:      test    rdx, rdx
0x18d3a:      je      0x18d90
0x18d3c:      push    r14
0x18d3e:      push    r13
0x18d40:      push    r12
0x18d42:      mov     r12, rsi
0x18d45:      push    rbp
0x18d46:      mov     ebp, edi
0x18d48:      push    rbx
0x18d49:      mov     rbx, rdx
0x18d4c:      call    0x4860
0x18d51:      mov     r13d, dword ptr [rax]
0x18d54:      mov     r14, rax
0x18d57:      jmp     0x18d66
0x18d66
0x18d59:      nop     dword ptr [rax]
0x18d60:      cmp     dword ptr [r14], 4
0x18d64:      jne     0x18d7d
0x18d66:      mov     dword ptr [r14], r13d
0x18d69:      mov     rdx, rbx
0x18d6c:      mov     rsi, r12
0x18d6f:      mov     edi, ebp
0x18d71:      call    0x47a0
0x18d76:      mov     r8d, eax
0x18d79:      test    eax, eax
0x18d7b:      js      0x18d60
0x18d7d:      pop     rbx
0x18d7e:      mov     eax, r8d
0x18d81:      pop     rbp
0x18d82:      pop     r12
0x18d84:      pop     r13
0x18d86:      pop     r14
0x18d88:      ret
0x18d89:      nop     dword ptr [rax]
0x18d90:      mov     eax, r8d
0x18d93:      ret
0x18d94:      nop     word ptr cs:[rax + rax]
0x18d9f:      nop
0x18da0:      endbr64
0x18da4:      xor     r8d, r8d

```

```

0x18da7:    test    rdx, rdx
0x18daa:    je      0x18e00
0x18dac:    push    r14
0x18dae:    push    r13
0x18db0:    push    r12
0x18db2:    mov     r12, rsi
0x18db5:    push    rbp
0x18db6:    mov     ebp, edi
0x18db8:    push    rbx
0x18db9:    mov     rbx, rdx
0x18dbc:    call    0x4860
0x18dc1:    mov     r13d, dword ptr [rax]
0x18dc4:    mov     r14, rax
0x18dc7:    jmp     0x18dd6
0x18dd6:
0x18dc9:    nop     dword ptr [rax]
0x18dd0:    cmp     dword ptr [r14], 4
0x18dd4:    jne     0x18ded
0x18dd6:    mov     dword ptr [r14], r13d
0x18dd9:    mov     rdx, rbx
0x18ddc:    mov     rsi, r12
0x18ddf:    mov     edi, ebp
0x18de1:    call    0x4790
0x18de6:    mov     r8d, eax
0x18de9:    test    eax, eax
0x18deb:    js      0x18dd0
0x18ded:    pop     rbx
0x18dee:    mov     eax, r8d
0x18df1:    pop     rbp
0x18df2:    pop     r12
0x18df4:    pop     r13
0x18df6:    pop     r14
0x18df8:    ret
0x18df9:    nop     dword ptr [rax]
0x18e00:    mov     eax, r8d
0x18e03:    ret
0x18e04:    nop     word ptr cs:[rax + rax]
0x18e0f:    nop
0x18e10:    endbr64
0x18e14:    xor     r8d, r8d
0x18e17:    test    rdx, rdx
0x18e1a:    je      0x18e70
0x18e1c:    push    r14
0x18e1e:    push    r13
0x18e20:    push    r12
0x18e22:    mov     r12, rsi
0x18e25:    push    rbp
0x18e26:    mov     rbp, rdi

```

```

0x18e29:    push    rbx
0x18e2a:    mov     rbx, rdx
0x18e2d:    call    0x4860
0x18e32:    mov     r13d, dword ptr [rax]
0x18e35:    mov     r14, rax
0x18e38:    jmp     0x18e46
0x18e46:
0x18e3a:    nop     word ptr [rax + rax]
0x18e40:    cmp     dword ptr [r14], 4
0x18e44:    jne     0x18e5e
0x18e46:    mov     dword ptr [r14], r13d
0x18e49:    mov     rdx, rbx
0x18e4c:    mov     rsi, r12
0x18e4f:    mov     rdi, rbp
0x18e52:    call    0x4910
0x18e57:    mov     r8d, eax
0x18e5a:    test    eax, eax
0x18e5c:    js      0x18e40
0x18e5e:    pop     rbx
0x18e5f:    mov     eax, r8d
0x18e62:    pop     rbp
0x18e63:    pop     r12
0x18e65:    pop     r13
0x18e67:    pop     r14
0x18e69:    ret
0x18e6a:    nop     word ptr [rax + rax]
0x18e70:    mov     eax, r8d
0x18e73:    ret
0x18e74:    nop     word ptr cs:[rax + rax]
0x18e7f:    nop
0x18e80:    endbr64
0x18e84:    push    r13
0x18e86:    mov     r13, rdi
0x18e89:    push    r12
0x18e8b:    push    rbp
0x18e8c:    push    rbx
0x18e8d:    mov     ebx, 0x64
0x18e92:    sub     rsp, 0x18
0x18e96:    jmp     0x18eaa
0x18eaa:
0x18e98:    nop     dword ptr [rax + rax]
0x18ea0:    mov     rdi, rbp
0x18ea3:    add     ebx, ebx
0x18ea5:    call    0x18d10
0x18eaa:    movsxd  r12, ebx
0x18ead:    mov     rdi, r12
0x18eb0:    call    0x1d670
0x18eb5:    mov     rdx, r12

```

```

0x18eb8:    mov     rdi, r13
0x18ebb:    mov     rbp, rax
0x18ebe:    mov     rsi, rax
0x18ec1:    xor     eax, eax
0x18ec3:    call    0x18e10
0x18ec8:    cmp     ebx, eax
0x18eca:    jle     0x18ea0
0x18ecc:    mov     rdi, rbp
0x18ecf:    mov     dword ptr [rsp + 0xc], eax
0x18ed3:    call    0x18d10
0x18ed8:    mov     eax, dword ptr [rsp + 0xc]
0x18edc:    add     rsp, 0x18
0x18ee0:    pop     rbx
0x18ee1:    pop     rbp
0x18ee2:    pop     r12
0x18ee4:    pop     r13
0x18ee6:    ret
0x18ee7:    nop     word ptr [rax + rax]
0x18ef0:    endbr64
0x18ef4:    push    r13
0x18ef6:    push    r12
0x18ef8:    mov     r12d, edi
0x18efb:    push    rbp
0x18efc:    mov     rbp, rsi
0x18eff:    push    rbx
0x18f00:    sub     rsp, 8
0x18f04:    call    0x4860
0x18f09:    mov     r13d, dword ptr [rax]
0x18f0c:    mov     rbx, rax
0x18f0f:    jmp     0x18f1d
0x18f1d:
0x18f11:    nop     dword ptr [rax]
0x18f18:    cmp     dword ptr [rbx], 4
0x18f1b:    jne     0x18f2f
0x18f1d:    mov     dword ptr [rbx], r13d
0x18f20:    mov     rsi, rbp
0x18f23:    mov     edi, r12d
0x18f26:    call    0x4eb0
0x18f2b:    test    eax, eax
0x18f2d:    js      0x18f18
0x18f2f:    add     rsp, 8
0x18f33:    pop     rbx
0x18f34:    pop     rbp
0x18f35:    pop     r12
0x18f37:    pop     r13
0x18f39:    ret
0x18f3a:    nop     word ptr [rax + rax]
0x18f40:    endbr64

```

```

0x18f44:    push    r13
0x18f46:    push    r12
0x18f48:    mov     r12, rdi
0x18f4b:    push    rbp
0x18f4c:    mov     rbp, rsi
0x18f4f:    push    rbx
0x18f50:    sub     rsp, 8
0x18f54:    call    0x4860
0x18f59:    mov     r13d, dword ptr [rax]
0x18f5c:    mov     rbx, rax
0x18f5f:    jmp     0x18f6d
0x18f6d:
0x18f61:    nop     dword ptr [rax]
0x18f68:    cmp     dword ptr [rbx], 4
0x18f6b:    jne     0x18f7f
0x18f6d:    mov     dword ptr [rbx], r13d
0x18f70:    mov     rsi, rbp
0x18f73:    mov     rdi, r12
0x18f76:    call    0x4b90
0x18f7b:    test    eax, eax
0x18f7d:    js      0x18f68
0x18f7f:    add     rsp, 8
0x18f83:    pop     rbx
0x18f84:    pop     rbp
0x18f85:    pop     r12
0x18f87:    pop     r13
0x18f89:    ret
0x18f8a:    nop     word ptr [rax + rax]
0x18f90:    endbr64
0x18f94:    push    r13
0x18f96:    push    r12
0x18f98:    mov     r12, rdi
0x18f9b:    push    rbp
0x18f9c:    mov     rbp, rsi
0x18f9f:    push    rbx
0x18fa0:    sub     rsp, 8
0x18fa4:    call    0x4860
0x18fa9:    mov     r13d, dword ptr [rax]
0x18fac:    mov     rbx, rax
0x18faf:    jmp     0x18fbd
0x18fbd:
0x18fb1:    nop     dword ptr [rax]
0x18fb8:    cmp     dword ptr [rbx], 4
0x18fbb:    jne     0x18fcf
0x18fbd:    mov     dword ptr [rbx], r13d
0x18fc0:    mov     rsi, rbp
0x18fc3:    mov     rdi, r12
0x18fc6:    call    0x4ad0

```

```

0x18fcb:    test    eax, eax
0x18fcd:    js      0x18fb8
0x18fcf:    add     rsp, 8
0x18fd3:    pop     rbx
0x18fd4:    pop     rbp
0x18fd5:    pop     r12
0x18fd7:    pop     r13
0x18fd9:    ret
0x18fda:    nop     word ptr [rax + rax]
0x18fe0:    endbr64
0x18fe4:    push    r12
0x18fe6:    push    rbp
0x18fe7:    push    rbx
0x18fe8:    mov     ebx, 0x40
0x18fed:    call    0x4860
0x18ff2:    mov     edi, 0x40
0x18ff7:    mov     dword ptr [rax], 0
0x18ffd:    mov     r12, rax
0x19000:    call    0x1d670
0x19005:    mov     rbp, rax
0x19008:    jmp     0x19031
0x19031:
0x1900a:    nop     word ptr [rax + rax]
0x19010:    cmp     dword ptr [r12], 0x22
0x19015:    jne     0x19050
0x19017:    add     rbx, 0x40
0x1901b:    mov     rdi, rbp
0x1901e:    mov     rsi, rbx
0x19021:    call    0x1d690
0x19026:    mov     dword ptr [r12], 0
0x1902e:    mov     rbp, rax
0x19031:    mov     rsi, rbx
0x19034:    mov     rdi, rbp
0x19037:    call    0x4a80
0x1903c:    test    rax, rax
0x1903f:    je      0x19010
0x19041:    mov     rax, rbp
0x19044:    pop     rbx
0x19045:    pop     rbp
0x19046:    pop     r12
0x19048:    ret
0x19049:    nop     dword ptr [rax]
0x19050:    mov     rdi, rbp
0x19053:    xor     ebp, ebp
0x19055:    call    0x18d10
0x1905a:    mov     rax, rbp
0x1905d:    pop     rbx
0x1905e:    pop     rbp

```



```

0x1905f:    pop     r12
0x19061:    ret
0x19062:    nop     word ptr cs:[rax + rax]
0x1906d:    nop     dword ptr [rax]
0x19070:    endbr64
0x19074:    push    rbp
0x19075:    mov     esi, 0x2f
0x1907a:    mov     rbp, rdi
0x1907d:    push    rbx
0x1907e:    sub     rsp, 8
0x19082:    call    0x4a20
0x19087:    test    rax, rax
0x1908a:    je      0x190a8
0x1908c:    mov     rbx, rax
0x1908f:    mov     rdi, rbp
0x19092:    call    0x1d7a0
0x19097:    sub     rbx, rbp
0x1909a:    mov     byte ptr [rax + rbx], 0x30
0x1909e:    add     rsp, 8
0x190a2:    pop     rbx
0x190a3:    pop     rbp
0x190a4:    ret
0x190a5:    nop     dword ptr [rax]
0x190a8:    add     rsp, 8
0x190ac:    lea     rdi, [rip + 0x862b]
0x190b3:    pop     rbx
0x190b4:    pop     rbp
0x190b5:    jmp     0x1d7a0
0x1d7a0
0x190ba:    nop     word ptr [rax + rax]
0x190c0:    endbr64
0x190c4:    push    rbx
0x190c5:    mov     rbx, rdi
0x190c8:    call    0x49b0
0x190cd:    lea     rax, [rbx + rax - 1]
0x190d2:    cmp     byte ptr [rax], 0x2f
0x190d5:    jne     0x190da
0x190d7:    mov     byte ptr [rax], 0
0x190da:    mov     rdi, rbx
0x190dd:    mov     esi, 0x2f
0x190e2:    call    0x4a20
0x190e7:    lea     rdx, [rax + 1]
0x190eb:    test    rax, rax
0x190ee:    mov     rax, rbx
0x190f1:    pop     rbx
0x190f2:    cmovne  rax, rdx
0x190f6:    ret
0x190f7:    nop     word ptr [rax + rax]

```

```

0x19100:      endbr64
0x19104:      push    rbx
0x19105:      mov     esi, 0x10
0x1910a:      mov     ebx, edi
0x1910c:      mov     edi, 1
0x19111:      call    0x1d740
0x19116:      mov     dword ptr [rax + 8], ebx
0x19119:      pop     rbx
0x1911a:      ret
0x1911b:      nop     dword ptr [rax + rax]
0x19120:      endbr64
0x19124:      test    rdi, rdi
0x19127:      je      0x19148
0x19129:      push    rbp
0x1912a:      mov     rbp, rdi
0x1912d:      mov     rdi, qword ptr [rdi]
0x19130:      test    rdi, rdi
0x19133:      je      0x1913a
0x19135:      call    0x18d10
0x1913a:      mov     rdi, rbp
0x1913d:      pop     rbp
0x1913e:      jmp     0x18d10
0x18d10
0x19143:      nop     dword ptr [rax + rax]
0x19148:      ret
0x19149:      nop     dword ptr [rax]
0x19150:      endbr64
0x19154:      push    rbp
0x19155:      mov     rbp, rsi
0x19158:      push    rbx
0x19159:      mov     rbx, rdi
0x1915c:      sub     rsp, 8
0x19160:      mov     eax, dword ptr [rdi + 0xc]
0x19163:      lea     esi, [rax + 1]
0x19166:      mov     dword ptr [rdi + 0xc], esi
0x19169:      imul    esi, dword ptr [rdi + 8]
0x1916d:      mov     rdi, qword ptr [rdi]
0x19170:      movsxd  rsi, esi
0x19173:      call    0x1d690
0x19178:      mov     ecx, dword ptr [rbx + 0xc]
0x1917b:      movsxd  rdx, dword ptr [rbx + 8]
0x1917f:      mov     rsi, rbp
0x19182:      mov     qword ptr [rbx], rax
0x19185:      add     rsp, 8
0x19189:      sub     ecx, 1
0x1918c:      pop     rbx
0x1918d:      pop     rbp
0x1918e:      imul    ecx, edx

```

```

0x19191:      movsxd  rcx, ecx
0x19194:      lea     rdi, [rax + rcx]
0x19198:      jmp     0x4bb0
0x4bb0:
0x1919d:      nop     dword ptr [rax]
0x191a0:      endbr64
0x191a4:      mov     eax, dword ptr [rdi + 0xc]
0x191a7:      test    eax, eax
0x191a9:      je      0x191f8
0x191ab:      push    rbp
0x191ac:      sub     eax, 1
0x191af:      mov     rbp, rsi
0x191b2:      push    rbx
0x191b3:      mov     rbx, rdi
0x191b6:      sub     rsp, 8
0x191ba:      movsxd  rdx, dword ptr [rdi + 8]
0x191be:      mov     dword ptr [rdi + 0xc], eax
0x191c1:      mov     rsi, rdx
0x191c4:      imul    esi, eax
0x191c7:      movsxd  rsi, esi
0x191ca:      add     rsi, qword ptr [rdi]
0x191cd:      mov     rdi, rbp
0x191d0:      call    0x4bb0
0x191d5:      mov     esi, dword ptr [rbx + 0xc]
0x191d8:      imul    esi, dword ptr [rbx + 8]
0x191dc:      mov     rdi, qword ptr [rbx]
0x191df:      movsxd  rsi, esi
0x191e2:      call    0x1d690
0x191e7:      mov     qword ptr [rbx], rax
0x191ea:      add     rsp, 8
0x191ee:      mov     rax, rbp
0x191f1:      pop     rbx
0x191f2:      pop     rbp
0x191f3:      ret
0x191f4:      nop     dword ptr [rax]
0x191f8:      xor     eax, eax
0x191fa:      ret
0x191fb:      nop     dword ptr [rax + rax]
0x19200:      endbr64
0x19204:      mov     ecx, dword ptr [rdi + 0xc]
0x19207:      cmp     ecx, edx
0x19209:      jl      0x19240
0x1920b:      test    ecx, ecx
0x1920d:      je      0x19240
0x1920f:      sub     rsp, 8
0x19213:      movsxd  r9, dword ptr [rdi + 8]
0x19217:      sub     ecx, edx
0x19219:      mov     r8, rsi

```

```

0x1921c:    imul    ecx, r9d
0x19220:    mov     rdx, r9
0x19223:    movsxd  rsi, ecx
0x19226:    add     rsi, qword ptr [rdi]
0x19229:    mov     rdi, r8
0x1922c:    call    0x4bb0
0x19231:    add     rsp, 8
0x19235:    ret
0x19236:    nop     word ptr cs:[rax + rax]
0x19240:    xor     eax, eax
0x19242:    ret
0x19243:    nop     word ptr cs:[rax + rax]
0x1924e:    nop
0x19250:    endbr64
0x19254:    push    rbx
0x19255:    mov     rbx, rdi
0x19258:    mov     dword ptr [rdi + 0xc], esi
0x1925b:    imul    esi, dword ptr [rdi + 8]
0x1925f:    mov     rdi, qword ptr [rdi]
0x19262:    movsxd  rsi, esi
0x19265:    call    0x1d690
0x1926a:    mov     qword ptr [rbx], rax
0x1926d:    pop     rbx
0x1926e:    ret
0x1926f:    nop
0x19270:    endbr64
0x19274:    mov     eax, dword ptr [rdi + 0xc]
0x19277:    ret
0x19278:    nop     dword ptr [rax + rax]
0x19280:    push    r12
0x19282:    mov     r12, rdi
0x19285:    mov     edi, 2
0x1928a:    push    rbp
0x1928b:    push    rbx
0x1928c:    mov     ebx, esi
0x1928e:    xor     esi, esi
0x19290:    call    0x4b50
0x19295:    xor     esi, esi
0x19297:    mov     edi, 3
0x1929c:    call    0x4b50
0x192a1:    test    ebx, ebx
0x192a3:    jne     0x192e0
0x192a5:    mov     r9, qword ptr [rip + 0x115b4]
0x192ac:    xor     r8d, r8d
0x192af:    mov     rcx, r12
0x192b2:    xor     eax, eax
0x192b4:    lea     rdi, [rip + 0x7363]
0x192bb:    lea     rdx, [rip + 0x911e]

```

```

0x192c2:    lea     rsi, [rip + 0x735a]
0x192c9:    call    0x4e30
0x192ce:    mov     edi, 0x7f
0x192d3:    call    0x48a0
0x192d8:    nop     dword ptr [rax + rax]
0x192e0:    mov     edi, 1
0x192e5:    lea     rbp, [rip + 0x90ed]
0x192ec:    call    0x4ab0
0x192f1:    mov     edi, 2
0x192f6:    call    0x4ab0
0x192fb:    mov     rdi, qword ptr [rip + 0x127ce]
0x19302:    mov     rsi, rbp
0x19305:    call    0x4da0
0x1930a:    mov     rdi, qword ptr [rip + 0x127b7]
0x19311:    mov     rsi, rbp
0x19314:    call    0x4da0
0x19319:    mov     edx, 4
0x1931e:    lea     rsi, [rip + 0x90b6]
0x19325:    mov     rdi, r12
0x19328:    call    0x4890
0x1932d:    test    eax, eax
0x1932f:    je      0x192a5
0x19335:    xor     edi, edi
0x19337:    call    0x4ab0
0x1933c:    jmp     0x192a5
0x192a5:
0x19341:    nop     word ptr cs:[rax + rax]
0x1934c:    nop     dword ptr [rax]
0x19350:    endbr64
0x19354:    push    r12
0x19356:    push    rbp
0x19357:    push    rbx
0x19358:    sub     rsp, 0x10
0x1935c:    mov     rax, qword ptr fs:[0x28]
0x19365:    mov     qword ptr [rsp + 8], rax
0x1936a:    xor     eax, eax
0x1936c:    test    rdi, rdi
0x1936f:    je      0x193d0
0x19371:    mov     rbp, rdi
0x19374:    mov     r12d, esi
0x19377:    call    0x4ed0
0x1937c:    mov     ebx, eax
0x1937e:    test    eax, eax
0x19380:    js      0x193c0
0x19382:    je      0x193dc
0x19384:    lea     rbp, [rsp + 4]
0x19389:    nop     dword ptr [rax]
0x19390:    mov     rdi, rbp

```

```

0x19393:      call    0x4ec0
0x19398:      cmp     eax, ebx
0x1939a:      jne     0x19390
0x1939c:      mov     eax, dword ptr [rsp + 4]
0x193a0:      mov     rdx, qword ptr [rsp + 8]
0x193a5:      sub     rdx, qword ptr fs:[0x28]
0x193ae:      jne     0x193d7
0x193b0:      add     rsp, 0x10
0x193b4:      pop     rbx
0x193b5:      pop     rbp
0x193b6:      pop     r12
0x193b8:      ret
0x193b9:      nop     dword ptr [rax]
0x193c0:      mov     eax, 0xffffffff
0x193c5:      jmp     0x193a0
0x193a0
0x193c7:      nop     word ptr [rax + rax]
0x193d0:      mov     eax, 1
0x193d5:      jmp     0x193a0
0x193a0
0x193d7:      call    0x49d0
0x193dc:      mov     esi, r12d
0x193df:      mov     rdi, rbp
0x193e2:      xor     eax, eax
0x193e4:      call    0x19280
0x193e9:      nop     dword ptr [rax]
0x193f0:      endbr64
0x193f4:      push    r13
0x193f6:      mov     r13, rdi
0x193f9:      push    r12
0x193fb:      push    rbp
0x193fc:      push    rbx
0x193fd:      sub     rsp, 0x18
0x19401:      mov     rax, qword ptr fs:[0x28]
0x1940a:      mov     qword ptr [rsp + 8], rax
0x1940f:      xor     eax, eax
0x19411:      test    esi, esi
0x19413:      je      0x19488
0x19415:      mov     edi, 1
0x1941a:      mov     ebp, esi
0x1941c:      call    0x18b70
0x19421:      test    r13, r13
0x19424:      je      0x19520
0x1942a:      call    0x4ed0
0x1942f:      mov     r12d, eax
0x19432:      test    eax, eax
0x19434:      js      0x19550
0x1943a:      je      0x19560

```

```

0x19440:    lea     rbx, [rsp + 4]
0x19445:    nop     dword ptr [rax]
0x19448:    mov     rdi, rbx
0x1944b:    call    0x4ec0
0x19450:    cmp     r12d, eax
0x19453:    jne     0x19448
0x19455:    mov     r12d, dword ptr [rsp + 4]
0x1945a:    xor     edi, edi
0x1945c:    call    0x18b70
0x19461:    mov     rax, qword ptr [rsp + 8]
0x19466:    sub     rax, qword ptr fs:[0x28]
0x1946f:    jne     0x1955b
0x19475:    add     rsp, 0x18
0x19479:    mov     eax, r12d
0x1947c:    pop     rbx
0x1947d:    pop     rbp
0x1947e:    pop     r12
0x19480:    pop     r13
0x19482:    ret
0x19483:    nop     dword ptr [rax + rax]
0x19488:    xor     edi, edi
0x1948a:    call    0x9b00
0x1948f:    xor     eax, eax
0x19491:    call    0xa6c0
0x19496:    mov     rdi, qword ptr [rip + 0x1142b]
0x1949d:    call    0xbb40
0x194a2:    xor     edi, edi
0x194a4:    call    0x18b90
0x194a9:    test    r13, r13
0x194ac:    je      0x19530
0x194b2:    call    0x4ed0
0x194b7:    mov     ebp, eax
0x194b9:    test    eax, eax
0x194bb:    js      0x19540
0x194c1:    je      0x1956c
0x194c7:    lea     rbx, [rsp + 4]
0x194cc:    nop     dword ptr [rax]
0x194d0:    mov     rdi, rbx
0x194d3:    call    0x4ec0
0x194d8:    cmp     ebp, eax
0x194da:    jne     0x194d0
0x194dc:    mov     r12d, dword ptr [rsp + 4]
0x194e1:    mov     edi, 1
0x194e6:    call    0x18b90
0x194eb:    mov     edx, 2
0x194f0:    mov     edi, 1
0x194f5:    lea     rsi, [rip + 0x8ee7]
0x194fc:    call    0x18da0

```

```

0x19501:      mov     edi, 1
0x19506:      call    0x9b00
0x1950b:      xor     eax, eax
0x1950d:      call    0xa6c0
0x19512:      xor     edi, edi
0x19514:      call    0x7890
0x19519:      jmp     0x19461
0x19461
0x1951e:      nop
0x19520:      mov     r12d, 1
0x19526:      jmp     0x1945a
0x1945a
0x1952b:      nop     dword ptr [rax + rax]
0x19530:      mov     r12d, 1
0x19536:      jmp     0x194e1
0x194e1
0x19538:      nop     dword ptr [rax + rax]
0x19540:      mov     r12d, 0xffffffff
0x19546:      jmp     0x194e1
0x194e1
0x19548:      nop     dword ptr [rax + rax]
0x19550:      mov     r12d, 0xffffffff
0x19556:      jmp     0x1945a
0x1945a
0x1955b:      call    0x49d0
0x19560:      mov     esi, ebp
0x19562:      mov     rdi, r13
0x19565:      xor     eax, eax
0x19567:      call    0x19280
0x1956c:      xor     esi, esi
0x1956e:      mov     rdi, r13
0x19571:      xor     eax, eax
0x19573:      call    0x19280
0x19578:      nop     dword ptr [rax + rax]
0x19580:      endbr64
0x19584:      mov     rdi, qword ptr [rip + 0x12545]
0x1958b:      test    rdi, rdi
0x1958e:      je      0x195b8
0x19590:      sub     rsp, 8
0x19594:      call    0x4870
0x19599:      mov     rdi, qword ptr [rip + 0x12528]
0x195a0:      test    rdi, rdi
0x195a3:      je      0x195b0
0x195a5:      add     rsp, 8
0x195a9:      jmp     0x4870
0x4870
0x195ae:      nop
0x195b0:      add     rsp, 8

```



```

0x195b4:      ret
0x195b5:      nop        dword ptr [rax]
0x195b8:      mov        rdi, qword ptr [rip + 0x12509]
0x195bf:      test       rdi, rdi
0x195c2:      je         0x195d0
0x195c4:      jmp        0x4870
0x4870
0x195c9:      nop        dword ptr [rax]
0x195d0:      ret
0x195d1:      nop        word ptr cs:[rax + rax]
0x195dc:      nop        dword ptr [rax]
0x195e0:      endbr64
0x195e4:      push       r12
0x195e6:      lea        rsi, [rip + 0x89dc]
0x195ed:      push       rbp
0x195ee:      mov        rbp, rdi
0x195f1:      sub        rsp, 8
0x195f5:      mov        rdi, qword ptr [rip + 0x124cc]
0x195fc:      call       0x4da0
0x19601:      test       rax, rax
0x19604:      je         0x19676
0x19606:      mov        edi, 0x801
0x1960b:      mov        r12, rax
0x1960e:      call       0x1d670
0x19613:      mov        rbp, rax
0x19616:      jmp        0x1964a
0x1964a
0x19618:      nop        dword ptr [rax + rax]
0x19620:      mov        rdi, rbp
0x19623:      call       0x49b0
0x19628:      cdqe
0x1962a:      lea        rax, [rbp + rax - 1]
0x1962f:      cmp        byte ptr [rax], 0xa
0x19632:      jne        0x19637
0x19634:      mov        byte ptr [rax], 0
0x19637:      xor        esi, esi
0x19639:      mov        edx, 0xa
0x1963e:      mov        rdi, rbp
0x19641:      call       0x8070
0x19646:      test       al, al
0x19648:      je         0x1965f
0x1964a:      mov        rdx, r12
0x1964d:      mov        esi, 0x801
0x19652:      mov        rdi, rbp
0x19655:      call       0x4b00
0x1965a:      test       rax, rax
0x1965d:      jne        0x19620
0x1965f:      mov        rdi, rbp

```

```

0x19662:      call    0x18d10
0x19667:      add     rsp, 8
0x1966b:      mov     rdi, r12
0x1966e:      pop     rbp
0x1966f:      pop     r12
0x19671:      jmp     0x4940
0x4940
0x19676:      mov     rdi, rbp
0x19679:      call    0x49b0
0x1967e:      mov     rdi, qword ptr [rip + 0x12443]
0x19685:      mov     r12, rax
0x19688:      call    0x49b0
0x1968d:      lea     rdi, [r12 + rax + 0x20]
0x19692:      call    0x1d670
0x19697:      mov     r9, qword ptr [rip + 0x1242a]
0x1969e:      mov     r8, rbp
0x196a1:      or      rdx, 0xffffffffffffffff
0x196a5:      mov     r12, rax
0x196a8:      mov     rdi, rax
0x196ab:      mov     esi, 1
0x196b0:      xor     eax, eax
0x196b2:      lea     rcx, [rip + 0x8d2d]
0x196b9:      call    0x4f10
0x196be:      mov     rdi, r12
0x196c1:      mov     edx, 0x1e
0x196c6:      xor     esi, esi
0x196c8:      call    0x8070
0x196cd:      pop     rax
0x196ce:      mov     rdi, r12
0x196d1:      pop     rbp
0x196d2:      pop     r12
0x196d4:      jmp     0x18d10
0x18d10
0x196d9:      nop     dword ptr [rax]
0x196e0:      endbr64
0x196e4:      push    r15
0x196e6:      push    r14
0x196e8:      push    r13
0x196ea:      push    r12
0x196ec:      push    rbp
0x196ed:      push    rbx
0x196ee:      sub     rsp, 8
0x196f2:      test    rdi, rdi
0x196f5:      je      0x19878
0x196fb:      mov     rbp, rdi
0x196fe:      call    0x49b0
0x19703:      lea     rdi, [rax + 1]
0x19707:      call    0x1d670

```

```

0x1970c:      mov     rsi, rbp
0x1970f:      mov     rdi, rax
0x19712:      call    0x48b0
0x19717:      cmp     byte ptr [rax], 0x7e
0x1971a:      mov     rbp, rax
0x1971d:      mov     r12, rax
0x19720:      jne     0x19800
0x19726:      movzx   eax, byte ptr [rax + 1]
0x1972a:      lea     r13, [rbp + 1]
0x1972e:      test    al, al
0x19730:      je      0x19818
0x19736:      cmp     al, 0x2f
0x19738:      je      0x19818
0x1973e:      mov     rdi, rbp
0x19741:      call    0x49b0
0x19746:      mov     rdi, rax
0x19749:      call    0x1d670
0x1974e:      movzx   ecx, byte ptr [rbp + 1]
0x19752:      mov     r14, rax
0x19755:      cmp     cl, 0x2f
0x19758:      je      0x19938
0x1975e:      test    cl, cl
0x19760:      je      0x19938
0x19766:      mov     edx, 2
0x1976b:      nop     dword ptr [rax + rax]
0x19770:      mov     byte ptr [r14 + rdx - 2], cl
0x19775:      movzx   ecx, byte ptr [rbp + rdx]
0x1977a:      lea     r13, [rbp + rdx]
0x1977f:      mov     rax, rdx
0x19782:      add     rdx, 1
0x19786:      test    cl, cl
0x19788:      je      0x1978f
0x1978a:      cmp     cl, 0x2f
0x1978d:      jne     0x19770
0x1978f:      lea     rax, [r14 + rax - 1]
0x19794:      mov     byte ptr [rax], 0
0x19797:      mov     rdi, r14
0x1979a:      call    0x4b60
0x1979f:      mov     rbx, rax
0x197a2:      test    rax, rax
0x197a5:      je      0x19880
0x197ab:      mov     rdi, qword ptr [rax + 0x20]
0x197af:      call    0x49b0
0x197b4:      mov     rdi, r13
0x197b7:      mov     r15, rax
0x197ba:      call    0x49b0
0x197bf:      lea     edi, [r15 + 1]
0x197c3:      movsxd  rdi, edi

```

```

0x197c6:      add     rdi, rax
0x197c9:      call    0x1d670
0x197ce:      mov     rsi, qword ptr [rbx + 0x20]
0x197d2:      mov     rdi, rax
0x197d5:      mov     r12, rax
0x197d8:      call    0x48b0
0x197dd:      movsxd  rdi, r15d
0x197e0:      mov     rsi, r13
0x197e3:      add     rdi, r12
0x197e6:      call    0x48b0
0x197eb:      mov     rdi, rbp
0x197ee:      call    0x18d10
0x197f3:      call    0x4c70
0x197f8:      mov     rdi, r14
0x197fb:      call    0x18d10
0x19800:      add     rsp, 8
0x19804:      mov     rax, r12
0x19807:      pop     rbx
0x19808:      pop     rbp
0x19809:      pop     r12
0x1980b:      pop     r13
0x1980d:      pop     r14
0x1980f:      pop     r15
0x19811:      ret
0x19812:      nop     word ptr [rax + rax]
0x19818:      lea     rdi, [rip + 0x6d41]
0x1981f:      call    0x47e0
0x19824:      mov     r14, rax
0x19827:      test    rax, rax
0x1982a:      je      0x198f8
0x19830:      mov     rdi, r14
0x19833:      call    0x49b0
0x19838:      mov     rdi, r13
0x1983b:      mov     rbx, rax
0x1983e:      call    0x49b0
0x19843:      movsxd  rbx, ebx
0x19846:      lea     rdi, [rbx + rax + 1]
0x1984b:      call    0x1d670
0x19850:      mov     rsi, r14
0x19853:      mov     r12, rax
0x19856:      mov     rdi, rax
0x19859:      call    0x48b0
0x1985e:      lea     rdi, [r12 + rbx]
0x19862:      mov     rsi, r13
0x19865:      call    0x48b0
0x1986a:      mov     rdi, rbp
0x1986d:      call    0x18d10
0x19872:      jmp     0x19800

```

```

0x19800
0x19874:      nop      dword ptr [rax]
0x19878:      xor      r12d, r12d
0x1987b:      jmp      0x19800
0x19800
0x1987d:      nop      dword ptr [rax]
0x19880:      mov      rdx, qword ptr [rip + 0x12251]
0x19887:      test     rdx, rdx
0x1988a:      je       0x197f3
0x19890:      mov      rdi, r14
0x19893:      xor      eax, eax
0x19895:      call     rdx
0x19897:      mov      r15, rax
0x1989a:      test     rax, rax
0x1989d:      je       0x197f3
0x198a3:      mov      rdi, rax
0x198a6:      call     0x49b0
0x198ab:      mov      rdi, r13
0x198ae:      mov      rbx, rax
0x198b1:      call     0x49b0
0x198b6:      lea      edi, [rbx + 1]
0x198b9:      movsxd   rdi, edi
0x198bc:      add      rdi, rax
0x198bf:      call     0x1d670
0x198c4:      mov      rsi, r15
0x198c7:      mov      rdi, rax
0x198ca:      mov      r12, rax
0x198cd:      call     0x48b0
0x198d2:      movsxd   rdi, ebx
0x198d5:      mov      rsi, r13
0x198d8:      add      rdi, r12
0x198db:      call     0x48b0
0x198e0:      mov      rdi, r15
0x198e3:      call     0x18d10
0x198e8:      mov      rdi, rbp
0x198eb:      call     0x18d10
0x198f0:      jmp      0x197f3
0x197f3
0x198f5:      nop      dword ptr [rax]
0x198f8:      call     0x49e0
0x198fd:      mov      edi, eax
0x198ff:      call     0x4970
0x19904:      test     rax, rax
0x19907:      je       0x19916
0x19909:      mov      r14, qword ptr [rax + 0x20]
0x1990d:      test     r14, r14
0x19910:      jne      0x19830
0x19916:      mov      rdi, r13

```

```

0x19919:      call    0x49b0
0x1991e:      lea     rdi, [rax + 1]
0x19922:      call    0x1d670
0x19927:      mov     r12, rax
0x1992a:      mov     rdi, rax
0x1992d:      jmp     0x19862
0x19862:
0x19932:      nop     word ptr [rax + rax]
0x19938:      mov     rax, r14
0x1993b:      jmp     0x19794
0x19794:
0x19940:      endbr64
0x19944:      push    r15
0x19946:      push    r14
0x19948:      mov     r14, rdi
0x1994b:      push    r13
0x1994d:      push    r12
0x1994f:      push    rbp
0x19950:      push    rbx
0x19951:      xor     ebx, ebx
0x19953:      sub     rsp, 0x48
0x19957:      mov     dword ptr [rsp + 0x1c], 0
0x1995f:      mov     qword ptr [rsp + 0x10], 0
0x19968:      nop     dword ptr [rax + rax]
0x19970:      mov     r15, qword ptr [rip + 0x10e31]
0x19977:      mov     rdi, r14
0x1997a:      mov     qword ptr [rsp + 8], r15
0x1997f:      call    0x49b0
0x19984:      mov     qword ptr [rsp + 0x20], rax
0x19989:      mov     rcx, rax
0x1998c:      movzx   eax, byte ptr [r14]
0x19990:      test    al, al
0x19992:      je      0x19bd0
0x19998:      cmp     al, 0x7e
0x1999a:      je      0x19bd0
0x199a0:      mov     dword ptr [rsp + 0x18], ecx
0x199a4:      mov     rax, rcx
0x199a7:      test    r15, r15
0x199aa:      je      0x19c50
0x199b0:      mov     edx, dword ptr [rsp + 0x20]
0x199b4:      test    edx, edx
0x199b6:      jle     0x19c4a
0x199bc:      mov     rax, qword ptr [rsp + 8]
0x199c1:      mov     dword ptr [rsp + 0x3c], ebx
0x199c5:      xor     r12d, r12d
0x199c8:      mov     rax, qword ptr [rax]
0x199cb:      mov     qword ptr [rsp + 0x28], rax
0x199d0:      mov     rax, qword ptr [rsp + 0x20]

```

```

0x199d5:      sub     eax, 1
0x199d8:      mov     qword ptr [rsp + 0x30], rax
0x199dd:      mov     rax, qword ptr [rsp + 8]
0x199e2:      mov     dword ptr [rsp + 0x38], r12d
0x199e7:      lea     rbp, [r14 + r12]
0x199eb:      lea     rbx, [rax + 8]
0x199ef:      mov     rax, qword ptr [rsp + 0x28]
0x199f4:      mov     r13, rax
0x199f7:      test    rax, rax
0x199fa:      jne     0x19a18
0x199fc:      jmp     0x19bb0
0x19bb0:
0x19a01:      nop     dword ptr [rax]
0x19a08:      mov     r13, qword ptr [rbx]
0x19a0b:      add     rbx, 8
0x19a0f:      test    r13, r13
0x19a12:      je      0x19bb0
0x19a18:      mov     rdi, r13
0x19a1b:      call    0x49b0
0x19a20:      mov     rsi, r13
0x19a23:      mov     rdi, rbp
0x19a26:      mov     rdx, rax
0x19a29:      mov     r15, rax
0x19a2c:      call    0x4890
0x19a31:      test    eax, eax
0x19a33:      jne     0x19a08
0x19a35:      mov     eax, dword ptr [rsp + 0x38]
0x19a39:      mov     ebx, dword ptr [rsp + 0x3c]
0x19a3d:      lea     eax, [rax + r15 - 1]
0x19a42:      lea     ecx, [rbx + rax]
0x19a45:      mov     dword ptr [rsp + 0x18], eax
0x19a49:      movsxd  r13, eax
0x19a4c:      mov     dword ptr [rsp + 8], ecx
0x19a50:      lea     r15, [r14 + r13]
0x19a54:      mov     ecx, dword ptr [rsp + 8]
0x19a58:      cmp     dword ptr [rsp + 0x1c], ecx
0x19a5c:      jle     0x19bf0
0x19a62:      movsxd  rbx, ebx
0x19a65:      add     rbx, qword ptr [rsp + 0x10]
0x19a6a:      mov     rdx, r13
0x19a6d:      mov     rsi, r14
0x19a70:      mov     rdi, rbx
0x19a73:      mov     qword ptr [rsp + 0x28], rbx
0x19a78:      call    0x4880
0x19a7d:      mov     rdi, r15
0x19a80:      mov     r13, qword ptr [rip + 0x10d19]
0x19a87:      call    0x49b0
0x19a8c:      mov     dword ptr [rsp + 0x20], eax

```

```

0x19a90:    test    eax, eax
0x19a92:    jle     0x19c64
0x19a98:    mov     rbx, r15
0x19a9b:    xor     ebp, ebp
0x19a9d:    movzx   eax, byte ptr [rbx]
0x19aa0:    cmp     al, 0x2f
0x19aa2:    je      0x19b80
0x19aa8:    test    al, al
0x19aaa:    je      0x19b80
0x19ab0:    test    r13, r13
0x19ab3:    je      0x19aea
0x19ab5:    mov     r14, r13
0x19ab8:    jmp     0x19ae2
0x19ae2:
0x19aba:    nop     word ptr [rax + rax]
0x19ac0:    mov     rdi, r12
0x19ac3:    add     r14, 8
0x19ac7:    call    0x49b0
0x19acc:    mov     rsi, r12
0x19acf:    mov     rdi, rbx
0x19ad2:    mov     rdx, rax
0x19ad5:    call    0x4890
0x19ada:    test    eax, eax
0x19adc:    je      0x19b80
0x19ae2:    mov     r12, qword ptr [r14]
0x19ae5:    test    r12, r12
0x19ae8:    jne     0x19ac0
0x19aea:    add     ebp, 1
0x19aed:    add     rbx, 1
0x19af1:    cmp     dword ptr [rsp + 0x20], ebp
0x19af5:    jne     0x19a9d
0x19af7:    nop     word ptr [rax + rax]
0x19b00:    lea     edi, [rbp + 1]
0x19b03:    movsxd  rbp, ebp
0x19b06:    movsxd  rdi, edi
0x19b09:    lea     r14, [r15 + rbp]
0x19b0d:    call    0x1d670
0x19b12:    mov     rdx, rbp
0x19b15:    mov     rsi, r15
0x19b18:    mov     r13, rax
0x19b1b:    mov     rdi, rax
0x19b1e:    call    0x4880
0x19b23:    mov     byte ptr [r13 + rbp], 0
0x19b29:    mov     rdi, r13
0x19b2c:    xor     eax, eax
0x19b2e:    call    0x196e0
0x19b33:    mov     rdi, r13
0x19b36:    mov     rbp, rax

```



```

0x19b39:      call    0x18d10
0x19b3e:      mov     rdi, rbp
0x19b41:      call    0x49b0
0x19b46:      mov     ecx, dword ptr [rsp + 8]
0x19b4a:      lea     ebx, [rcx + rax]
0x19b4d:      cmp     ebx, dword ptr [rsp + 0x1c]
0x19b51:      jge     0x19c20
0x19b57:      movsxd  rdi, dword ptr [rsp + 8]
0x19b5c:      mov     rsi, rbp
0x19b5f:      add     rdi, qword ptr [rsp + 0x10]
0x19b64:      call    0x48b0
0x19b69:      mov     rdi, rbp
0x19b6c:      call    0x18d10
0x19b71:      jmp     0x19970
0x19970:
0x19b76:      nop     word ptr cs:[rax + rax]
0x19b80:      or      dword ptr [rsp + 0x18], ebp
0x19b84:      mov     eax, dword ptr [rsp + 0x18]
0x19b88:      test    eax, eax
0x19b8a:      jne     0x19b00
0x19b90:      mov     rax, qword ptr [rsp + 0x28]
0x19b95:      mov     byte ptr [rax], 0
0x19b98:      mov     rax, qword ptr [rsp + 0x10]
0x19b9d:      add     rsp, 0x48
0x19ba1:      pop     rbx
0x19ba2:      pop     rbp
0x19ba3:      pop     r12
0x19ba5:      pop     r13
0x19ba7:      pop     r14
0x19ba9:      pop     r15
0x19bab:      ret
0x19bac:      nop     dword ptr [rax]
0x19bb0:      lea     rax, [r12 + 1]
0x19bb5:      cmp     qword ptr [rsp + 0x30], r12
0x19bba:      je      0x19c46
0x19bc0:      mov     r12, rax
0x19bc3:      jmp     0x199dd
0x199dd:
0x19bc8:      nop     dword ptr [rax + rax]
0x19bd0:      mov     dword ptr [rsp + 8], ebx
0x19bd4:      mov     r15, r14
0x19bd7:      mov     ecx, dword ptr [rsp + 8]
0x19bdb:      xor     r13d, r13d
0x19bde:      mov     dword ptr [rsp + 0x18], 0
0x19be6:      cmp     dword ptr [rsp + 0x1c], ecx
0x19bea:      jg      0x19a62
0x19bf0:      mov     eax, dword ptr [rsp + 0x1c]
0x19bf4:      mov     ecx, dword ptr [rsp + 0x18]

```

```

0x19bf8:      mov     rdi, qword ptr [rsp + 0x10]
0x19bfd:      lea     eax, [rax + rcx + 0x14]
0x19c01:      lea     esi, [rax + 1]
0x19c04:      mov     dword ptr [rsp + 0x1c], eax
0x19c08:      movsxd  rsi, esi
0x19c0b:      call    0x1d690
0x19c10:      mov     qword ptr [rsp + 0x10], rax
0x19c15:      jmp     0x19a62
0x19a62
0x19c1a:      nop     word ptr [rax + rax]
0x19c20:      mov     ecx, dword ptr [rsp + 0x1c]
0x19c24:      mov     rdi, qword ptr [rsp + 0x10]
0x19c29:      lea     eax, [rcx + rax + 0x14]
0x19c2d:      lea     esi, [rax + 1]
0x19c30:      mov     dword ptr [rsp + 0x1c], eax
0x19c34:      movsxd  rsi, esi
0x19c37:      call    0x1d690
0x19c3c:      mov     qword ptr [rsp + 0x10], rax
0x19c41:      jmp     0x19b57
0x19b57
0x19c46:      mov     ebx, dword ptr [rsp + 0x3c]
0x19c4a:      mov     rax, qword ptr [rsp + 0x20]
0x19c4f:      nop
0x19c50:      mov     ecx, eax
0x19c52:      movsxd  r13, eax
0x19c55:      add     ecx, ebx
0x19c57:      lea     r15, [r14 + r13]
0x19c5b:      mov     dword ptr [rsp + 8], ecx
0x19c5f:      jmp     0x19a54
0x19a54
0x19c64:      xor     ebp, ebp
0x19c66:      jmp     0x19b84
0x19b84
0x19c6b:      nop     dword ptr [rax + rax]
0x19c70:      push    r15
0x19c72:      mov     r15, rsi
0x19c75:      push    r14
0x19c77:      mov     r14d, edx
0x19c7a:      push    r13
0x19c7c:      push    r12
0x19c7e:      mov     r12, rdi
0x19c81:      push    rbp
0x19c82:      push    rbx
0x19c83:      sub     rsp, 0x18
0x19c87:      movsxd  rsi, dword ptr [rsi]
0x19c8a:      lea     r13, [rdi + rsi]
0x19c8e:      mov     edi, ecx
0x19c90:      mov     ebx, esi

```

```

0x19c92:    mov     rbp, rsi
0x19c95:    movzx   eax, byte ptr [r13]
0x19c9a:    and     edi, 1
0x19c9d:    mov     rdx, rsi
0x19ca0:    test    al, al
0x19ca2:    jne     0x19cdc
0x19ca4:    jmp     0x19db9
0x19db9:
0x19ca9:    nop     dword ptr [rax]
0x19cb0:    cmp     al, 0x5c
0x19cb2:    jne     0x19cc5
0x19cb4:    movsx   eax, byte ptr [r12 + rdx + 1]
0x19cba:    cmp     eax, r14d
0x19cbd:    sete    al
0x19cc0:    movzx   eax, al
0x19cc3:    add     ebx, eax
0x19cc5:    add     ebx, 1
0x19cc8:    movsxd  rdx, ebx
0x19ccb:    lea     r13, [r12 + rdx]
0x19ccf:    movzx   eax, byte ptr [r13]
0x19cd4:    test    al, al
0x19cd6:    je      0x19da0
0x19cdc:    movsx   ecx, al
0x19cdf:    cmp     ecx, r14d
0x19ce2:    jne     0x19cb0
0x19ce4:    cmp     ebp, ebx
0x19ce6:    jl      0x19d10
0x19ce8:    test    dil, dil
0x19ceb:    jne     0x19d10
0x19ced:    movzx   eax, byte ptr [r13]
0x19cf2:    xor     r9d, r9d
0x19cf5:    cmp     al, 1
0x19cf7:    sbb     ebx, -1
0x19cfa:    mov     dword ptr [r15], ebx
0x19cfd:    add     rsp, 0x18
0x19d01:    mov     rax, r9
0x19d04:    pop     rbx
0x19d05:    pop     rbp
0x19d06:    pop     r12
0x19d08:    pop     r13
0x19d0a:    pop     r14
0x19d0c:    pop     r15
0x19d0e:    ret
0x19d0f:    nop
0x19d10:    mov     edi, ebx
0x19d12:    mov     qword ptr [rsp], r8
0x19d16:    sub     edi, ebp
0x19d18:    mov     qword ptr [rsp + 8], rsi

```

```

0x19d1d:      add     edi, 1
0x19d20:      movsxd  rdi, edi
0x19d23:      call    0x1d670
0x19d28:      cmp     ebp, ebx
0x19d2a:      mov     r8, qword ptr [rsp]
0x19d2e:      mov     r9, rax
0x19d31:      jge     0x19dd6
0x19d37:      mov     rsi, qword ptr [rsp + 8]
0x19d3c:      mov     eax, 1
0x19d41:      jmp     0x19d5d
0x19d5d
0x19d43:      nop     dword ptr [rax + rax]
0x19d48:      add     ebp, 1
0x19d4b:      mov     byte ptr [r9 + rax - 1], dl
0x19d50:      mov     edx, eax
0x19d52:      add     rax, 1
0x19d56:      cmp     ebp, ebx
0x19d58:      jge     0x19d80
0x19d5a:      movsxd  rsi, ebp
0x19d5d:      movzx   edx, byte ptr [r12 + rsi]
0x19d62:      cmp     dl, 0x5c
0x19d65:      jne     0x19d48
0x19d67:      movsx   esi, byte ptr [r12 + rsi + 1]
0x19d6d:      cmp     esi, r14d
0x19d70:      jne     0x19d48
0x19d72:      add     ebp, 1
0x19d75:      mov     edx, esi
0x19d77:      jmp     0x19d48
0x19d48
0x19d79:      nop     dword ptr [rax]
0x19d80:      movsxd  rax, edx
0x19d83:      add     rax, r9
0x19d86:      mov     byte ptr [rax], 0
0x19d89:      mov     dword ptr [r8], edx
0x19d8c:      movzx   eax, byte ptr [r13]
0x19d91:      jmp     0x19cf5
0x19cf5
0x19d96:      nop     word ptr cs:[rax + rax]
0x19da0:      cmp     ebp, ebx
0x19da2:      jl      0x19d10
0x19da8:      test    dil, dil
0x19dab:      jne     0x19d10
0x19db1:      xor     r9d, r9d
0x19db4:      jmp     0x19cfa
0x19cfa
0x19db9:      test    ecx, ecx
0x19dbb:      je      0x19db1
0x19dbd:      mov     edi, 1

```

```

0x19dc2:    mov     qword ptr [rsp], r8
0x19dc6:    call    0x1d670
0x19dcb:    mov     r8, qword ptr [rsp]
0x19dcf:    xor     edx, edx
0x19dd1:    mov     r9, rax
0x19dd4:    jmp     0x19d86
0x19d86:
0x19dd6:    xor     edx, edx
0x19dd8:    jmp     0x19d86
0x19d86:
0x19dda:    nop     word ptr [rax + rax]
0x19de0:    push    r15
0x19de2:    push    r14
0x19de4:    push    r13
0x19de6:    push    r12
0x19de8:    push    rbp
0x19de9:    push    rbx
0x19dea:    sub     rsp, 0x28
0x19dee:    mov     eax, dword ptr [rip + 0x11cf0]
0x19df4:    mov     ebx, dword ptr [rip + 0x11d02]
0x19dfa:    mov     dword ptr [rsp + 8], esi
0x19dfe:    mov     dword ptr [rsp + 0x18], edx
0x19e02:    mov     dword ptr [rsp + 0x1c], eax
0x19e06:    test    eax, eax
0x19e08:    je      0x19f9d
0x19e0e:    mov     rbp, rdi
0x19e11:    cmp     eax, ebx
0x19e13:    je      0x19fb0
0x19e19:    call    0x49b0
0x19e1e:    mov     ecx, dword ptr [rsp + 8]
0x19e22:    mov     dword ptr [rsp + 0xc], eax
0x19e26:    test    ecx, ecx
0x19e28:    js      0x19fbb
0x19e2e:    mov     rax, qword ptr [rip + 0x11ccb]
0x19e35:    movsxd  r14, dword ptr [rsp + 0xc]
0x19e3a:    mov     qword ptr [rsp + 0x10], rax
0x19e3f:    mov     edx, dword ptr [rsp + 8]
0x19e43:    test    edx, edx
0x19e45:    js      0x19f16
0x19e4b:    cmp     dword ptr [rsp + 0x1c], ebx
0x19e4f:    je      0x19f9d
0x19e55:    mov     rcx, qword ptr [rsp + 0x10]
0x19e5a:    movsxd  rax, ebx
0x19e5d:    mov     rax, qword ptr [rcx + rax*8]
0x19e61:    mov     r13, qword ptr [rax]
0x19e64:    mov     rdi, r13
0x19e67:    call    0x49b0
0x19e6c:    mov     r10, rax

```

```

0x19e6f:    cmp     dword ptr [rsp + 0xc], eax
0x19e73:    jg      0x19f07
0x19e79:    cmp     dword ptr [rsp + 0x18], 1
0x19e7e:    movzx   r12d, byte ptr [rbp]
0x19e83:    je      0x19ee0
0x19e85:    mov     r15d, dword ptr [rsp + 0x18]
0x19e8a:    sub     r10d, dword ptr [rsp + 0xc]
0x19e8f:    jmp     0x19ea5
0x19ea5:
0x19e91:    nop     dword ptr [rax]
0x19e98:    add     r15d, 1
0x19e9c:    add     r13, 1
0x19ea0:    cmp     r15d, r10d
0x19ea3:    jg      0x19f07
0x19ea5:    cmp     byte ptr [r13], r12b
0x19ea9:    jne     0x19e98
0x19eab:    mov     rdx, r14
0x19eae:    mov     rsi, r13
0x19eb1:    mov     rdi, rbp
0x19eb4:    mov     dword ptr [rsp + 4], r10d
0x19eb9:    call    0x4890
0x19ebe:    mov     r10d, dword ptr [rsp + 4]
0x19ec3:    test    eax, eax
0x19ec5:    jne     0x19e98
0x19ec7:    mov     dword ptr [rip + 0x11c2f], ebx
0x19ecd:    add     rsp, 0x28
0x19ed1:    mov     eax, r15d
0x19ed4:    pop     rbx
0x19ed5:    pop     rbp
0x19ed6:    pop     r12
0x19ed8:    pop     r13
0x19eda:    pop     r14
0x19edc:    pop     r15
0x19ede:    ret
0x19edf:    nop
0x19ee0:    cmp     byte ptr [r13], r12b
0x19ee4:    jne     0x19efb
0x19ee6:    mov     rdx, r14
0x19ee9:    mov     rsi, r13
0x19eec:    mov     rdi, rbp
0x19eef:    call    0x4890
0x19ef4:    mov     r15d, eax
0x19ef7:    test    eax, eax
0x19ef9:    je      0x19ec7
0x19efb:    mov     eax, dword ptr [rsp + 8]
0x19eff:    test    eax, eax
0x19f01:    js      0x19f90
0x19f07:    mov     edx, dword ptr [rsp + 8]

```

```

0x19f0b:      add     ebx, 1
0x19f0e:      test    edx, edx
0x19f10:      jns     0x19e4b
0x19f16:      mov     rcx, qword ptr [rsp + 0x10]
0x19f1b:      movsxd   rax, ebx
0x19f1e:      mov     rax, qword ptr [rcx + rax*8]
0x19f22:      mov     r13, qword ptr [rax]
0x19f25:      mov     rdi, r13
0x19f28:      call    0x49b0
0x19f2d:      mov     r8, rax
0x19f30:      cmp     dword ptr [rsp + 0xc], eax
0x19f34:      jg      0x19f90
0x19f36:      cmp     dword ptr [rsp + 0x18], 1
0x19f3b:      movzx   r12d, byte ptr [rbp]
0x19f40:      je      0x19ee0
0x19f42:      mov     r15d, r8d
0x19f45:      sub     r15d, dword ptr [rsp + 0xc]
0x19f4a:      movsxd   rcx, r15d
0x19f4d:      add     r13, rcx
0x19f50:      jmp     0x19f66
0x19f66:
0x19f52:      nop     word ptr [rax + rax]
0x19f58:      sub     r15d, 1
0x19f5c:      sub     r13, 1
0x19f60:      cmp     r15d, -1
0x19f64:      je      0x19f90
0x19f66:      cmp     byte ptr [r13], r12b
0x19f6a:      jne     0x19f58
0x19f6c:      mov     rdx, r14
0x19f6f:      mov     rsi, r13
0x19f72:      mov     rdi, rbp
0x19f75:      call    0x4890
0x19f7a:      test    eax, eax
0x19f7c:      je      0x19ec7
0x19f82:      sub     r15d, 1
0x19f86:      sub     r13, 1
0x19f8a:      cmp     r15d, -1
0x19f8e:      jne     0x19f66
0x19f90:      sub     ebx, 1
0x19f93:      test    dword ptr [rsp + 8], ebx
0x19f97:      jns     0x19e3f
0x19f9d:      mov     r15d, 0xffffffff
0x19fa3:      jmp     0x19ecd
0x19ecd:
0x19fa8:      nop     dword ptr [rax + rax]
0x19fb0:      test    esi, esi
0x19fb2:      jns     0x19f9d
0x19fb4:      mov     eax, dword ptr [rsp + 0x1c]

```

```

0x19fb8:    lea     ebx, [rax - 1]
0x19fbb:    mov     rdi, rbp
0x19fbe:    call    0x49b0
0x19fc3:    mov     dword ptr [rsp + 0xc], eax
0x19fc7:    test    dword ptr [rsp + 8], ebx
0x19fcb:    jns     0x19e2e
0x19fd1:    jmp     0x19f9d
0x19f9d:
0x19fd3:    nop     word ptr cs:[rax + rax]
0x19fde:    nop
0x19fe0:    push    r14
0x19fe2:    push    r13
0x19fe4:    movsxd  r13, esi
0x19fe7:    sub     edx, r13d
0x19fea:    push    r12
0x19fec:    push    rbp
0x19fed:    mov     rbp, rdi
0x19ff0:    push    rbx
0x19ff1:    movsxd  rbx, edx
0x19ff4:    cmp     ecx, 2
0x19ff7:    je      0x1a070
0x19ff9:    mov     eax, 0x1d
0x19ffe:    lea     r14, [rip + 0x843a]
0x1a005:    cmp     ecx, 3
0x1a008:    je      0x1a028
0x1a00a:    xor     eax, eax
0x1a00c:    lea     r14, [rip + 0x83f5]
0x1a013:    lea     rdx, [rip + 0x83fe]
0x1a01a:    cmp     ecx, 1
0x1a01d:    sete    al
0x1a020:    cmovbe  r14, rdx
0x1a024:    lea     eax, [rax + rax*2 + 0xf]
0x1a028:    lea     edi, [rax + rbx + 3]
0x1a02c:    movsxd  rdi, edi
0x1a02f:    call    0x1d670
0x1a034:    lea     rsi, [rbp + r13]
0x1a039:    mov     rdx, rbx
0x1a03c:    mov     r12, rax
0x1a03f:    mov     rdi, rax
0x1a042:    call    0x4880
0x1a047:    mov     byte ptr [r12 + rbx], 0x3a
0x1a04c:    lea     rdi, [r12 + rbx + 2]
0x1a051:    mov     rsi, r14
0x1a054:    mov     byte ptr [r12 + rbx + 1], 0x20
0x1a05a:    call    0x48b0
0x1a05f:    pop     rbx
0x1a060:    mov     rax, r12
0x1a063:    pop     rbp

```



```

0x1a064:      pop     r12
0x1a066:      pop     r13
0x1a068:      pop     r14
0x1a06a:      ret
0x1a06b:      nop     dword ptr [rax + rax]
0x1a070:      mov     eax, 0x13
0x1a075:      lea     r14, [rip + 0x83af]
0x1a07c:      jmp     0x1a028
0x1a028
0x1a07e:      nop
0x1a080:      push    r12
0x1a082:      push    rbp
0x1a083:      push    rbx
0x1a084:      test    rdi, rdi
0x1a087:      je      0x1a0b0
0x1a089:      mov     rbp, rdi
0x1a08c:      call    0x49b0
0x1a091:      lea     rdi, [rax + 1]
0x1a095:      call    0x1d670
0x1a09a:      pop     rbx
0x1a09b:      mov     rsi, rbp
0x1a09e:      pop     rbp
0x1a09f:      mov     rdi, rax
0x1a0a2:      pop     r12
0x1a0a4:      jmp     0x48b0
0x48b0
0x1a0a9:      nop     dword ptr [rax]
0x1a0b0:      lea     rdi, [rip + 0x64a9]
0x1a0b7:      call    0x47e0
0x1a0bc:      mov     r12, rax
0x1a0bf:      test    rax, rax
0x1a0c2:      je      0x1a110
0x1a0c4:      mov     rdi, rax
0x1a0c7:      call    0x49b0
0x1a0cc:      lea     edi, [rax + 0xa]
0x1a0cf:      movsxd  rbp, eax
0x1a0d2:      movsxd  rdi, edi
0x1a0d5:      lea     rbx, [rbp + 1]
0x1a0d9:      call    0x1d670
0x1a0de:      mov     rsi, r12
0x1a0e1:      mov     rdi, rax
0x1a0e4:      call    0x48b0
0x1a0e9:      movabs  rdx, 0x79726f747369682e
0x1a0f3:      mov     byte ptr [rax + rbp], 0x2f
0x1a0f7:      mov     r8, rax
0x1a0fa:      lea     rax, [rax + rbx]
0x1a0fe:      mov     qword ptr [rax], rdx
0x1a101:      mov     byte ptr [rax + 8], 0

```

```

0x1a105:      mov     rax, r8
0x1a108:      pop     rbx
0x1a109:      pop     rbp
0x1a10a:      pop     r12
0x1a10c:      ret
0x1a10d:      nop     dword ptr [rax]
0x1a110:      mov     ebx, 2
0x1a115:      mov     ebp, 1
0x1a11a:      mov     edi, 0xb
0x1a11f:      lea     r12, [rip + 0x75b8]
0x1a126:      jmp     0x1a0d9
0x1a0d9
0x1a128:      nop     dword ptr [rax + rax]
0x1a130:      push    r15
0x1a132:      xor     eax, eax
0x1a134:      push    r14
0x1a136:      mov     r14d, esi
0x1a139:      push    r13
0x1a13b:      push    r12
0x1a13d:      push    rbp
0x1a13e:      push    rbx
0x1a13f:      mov     ebx, edx
0x1a141:      sub     rsp, 0x18
0x1a145:      call    0x1a080
0x1a14a:      cmp     ebx, 1
0x1a14d:      mov     edx, 0x1b6
0x1a152:      sbb     esi, esi
0x1a154:      mov     r13, rax
0x1a157:      mov     rdi, rax
0x1a15a:      xor     eax, eax
0x1a15c:      and     esi, 0x1c0
0x1a162:      add     esi, 0x241
0x1a168:      call    0x4d80
0x1a16d:      cmp     eax, -1
0x1a170:      je      0x1a270
0x1a176:      mov     ebp, eax
0x1a178:      mov     eax, dword ptr [rip + 0x11966]
0x1a17e:      cmp     eax, r14d
0x1a181:      mov     edx, eax
0x1a183:      cmovle  r14d, eax
0x1a187:      sub     edx, r14d
0x1a18a:      cmp     eax, edx
0x1a18c:      jle     0x1a290
0x1a192:      sub     eax, 1
0x1a195:      mov     rcx, qword ptr [rip + 0x11964]
0x1a19c:      movsxd  rsi, edx
0x1a19f:      xor     r12d, r12d
0x1a1a2:      sub     eax, edx

```

```

0x1a1a4:      add     rax, rsi
0x1a1a7:      lea     rbx, [rcx + rsi*8]
0x1a1ab:      lea     r15, [rcx + rax*8 + 8]
0x1a1b0:      mov     rax, qword ptr [rbx]
0x1a1b3:      add     rbx, 8
0x1a1b7:      mov     rdi, qword ptr [rax]
0x1a1ba:      call    0x49b0
0x1a1bf:      lea     r12d, [r12 + rax + 1]
0x1a1c4:      cmp     r15, rbx
0x1a1c7:      jne     0x1a1b0
0x1a1c9:      movsxd  rax, r12d
0x1a1cc:      mov     qword ptr [rsp + 8], rax
0x1a1d1:      mov     rdi, qword ptr [rsp + 8]
0x1a1d6:      call    0x1d670
0x1a1db:      mov     r15, rax
0x1a1de:      mov     eax, dword ptr [rip + 0x11900]
0x1a1e4:      mov     ebx, eax
0x1a1e6:      sub     ebx, r14d
0x1a1e9:      cmp     eax, ebx
0x1a1eb:      jle     0x1a238
0x1a1ed:      mov     r12, qword ptr [rip + 0x1190c]
0x1a1f4:      movsxd  rbx, ebx
0x1a1f7:      xor     r14d, r14d
0x1a1fa:      nop     word ptr [rax + rax]
0x1a200:      mov     rax, qword ptr [r12 + rbx*8]
0x1a204:      movsxd  rdi, r14d
0x1a207:      add     rdi, r15
0x1a20a:      mov     rsi, qword ptr [rax]
0x1a20d:      call    0x48b0
0x1a212:      mov     rax, qword ptr [r12 + rbx*8]
0x1a216:      add     rbx, 1
0x1a21a:      mov     rdi, qword ptr [rax]
0x1a21d:      call    0x49b0
0x1a222:      add     eax, r14d
0x1a225:      lea     r14d, [rax + 1]
0x1a229:      cdqe
0x1a22b:      mov     byte ptr [r15 + rax], 0xa
0x1a230:      cmp     dword ptr [rip + 0x118ae], ebx
0x1a236:      jg      0x1a200
0x1a238:      mov     rdx, qword ptr [rsp + 8]
0x1a23d:      mov     rsi, r15
0x1a240:      mov     edi, ebp
0x1a242:      call    0x4790
0x1a247:      mov     rdi, r15
0x1a24a:      call    0x18d10
0x1a24f:      mov     edi, ebp
0x1a251:      call    0x4ab0
0x1a256:      mov     rdi, r13

```

```

0x1a259:    call    0x18d10
0x1a25e:    add     rsp, 0x18
0x1a262:    xor     eax, eax
0x1a264:    pop     rbx
0x1a265:    pop     rbp
0x1a266:    pop     r12
0x1a268:    pop     r13
0x1a26a:    pop     r14
0x1a26c:    pop     r15
0x1a26e:    ret
0x1a26f:    nop
0x1a270:    mov     rdi, r13
0x1a273:    call    0x18d10
0x1a278:    call    0x4860
0x1a27d:    mov     eax, dword ptr [rax]
0x1a27f:    add     rsp, 0x18
0x1a283:    pop     rbx
0x1a284:    pop     rbp
0x1a285:    pop     r12
0x1a287:    pop     r13
0x1a289:    pop     r14
0x1a28b:    pop     r15
0x1a28d:    ret
0x1a28e:    nop
0x1a290:    mov     qword ptr [rsp + 8], 0
0x1a299:    jmp     0x1a1d1
0x1a1d1:
0x1a29e:    nop
0x1a2a0:    endbr64
0x1a2a4:    sub     rsp, 8
0x1a2a8:    mov     edi, 0x18
0x1a2ad:    call    0x1d670
0x1a2b2:    mov     rdx, qword ptr [rip + 0x11847]
0x1a2b9:    mov     qword ptr [rax], rdx
0x1a2bc:    mov     edx, dword ptr [rip + 0x1183a]
0x1a2c2:    mov     dword ptr [rax + 8], edx
0x1a2c5:    mov     edx, dword ptr [rip + 0x11819]
0x1a2cb:    mov     dword ptr [rax + 0xc], edx
0x1a2ce:    mov     edx, dword ptr [rip + 0x11824]
0x1a2d4:    mov     dword ptr [rax + 0x10], edx
0x1a2d7:    add     rsp, 8
0x1a2db:    ret
0x1a2dc:    nop     dword ptr [rax]
0x1a2e0:    endbr64
0x1a2e4:    mov     rax, qword ptr [rdi]
0x1a2e7:    mov     qword ptr [rip + 0x11812], rax
0x1a2ee:    mov     eax, dword ptr [rdi + 8]
0x1a2f1:    mov     dword ptr [rip + 0x11805], eax

```

```

0x1a2f7:    mov     eax, dword ptr [rdi + 0xc]
0x1a2fa:    mov     dword ptr [rip + 0x117e4], eax
0x1a300:    mov     eax, dword ptr [rdi + 0x10]
0x1a303:    mov     dword ptr [rip + 0x117ef], eax
0x1a309:    ret
0x1a30a:    nop     word ptr [rax + rax]
0x1a310:    endbr64
0x1a314:    mov     eax, dword ptr [rip + 0x117ca]
0x1a31a:    mov     dword ptr [rip + 0x117dc], eax
0x1a320:    ret
0x1a321:    nop     word ptr cs:[rax + rax]
0x1a32c:    nop     dword ptr [rax]
0x1a330:    endbr64
0x1a334:    push    r12
0x1a336:    xor     r12d, r12d
0x1a339:    push    rbx
0x1a33a:    sub     rsp, 8
0x1a33e:    mov     rbx, qword ptr [rip + 0x117bb]
0x1a345:    test    rbx, rbx
0x1a348:    jne     0x1a35b
0x1a34a:    jmp     0x1a367
0x1a367:
0x1a34c:    nop     dword ptr [rax]
0x1a350:    mov     rdi, qword ptr [rax]
0x1a353:    call    0x49b0
0x1a358:    add     r12d, eax
0x1a35b:    mov     rax, qword ptr [rbx]
0x1a35e:    add     rbx, 8
0x1a362:    test    rax, rax
0x1a365:    jne     0x1a350
0x1a367:    add     rsp, 8
0x1a36b:    mov     eax, r12d
0x1a36e:    pop     rbx
0x1a36f:    pop     r12
0x1a371:    ret
0x1a372:    nop     word ptr cs:[rax + rax]
0x1a37d:    nop     dword ptr [rax]
0x1a380:    endbr64
0x1a384:    push    rbp
0x1a385:    mov     rbp, rdi
0x1a388:    push    rbx
0x1a389:    sub     rsp, 8
0x1a38d:    mov     eax, dword ptr [rip + 0x11759]
0x1a393:    test    eax, eax
0x1a395:    je      0x1a3aa
0x1a397:    movsxd  rdx, dword ptr [rip + 0x11746]
0x1a39e:    cmp     edx, dword ptr [rip + 0x11744]
0x1a3a4:    je      0x1a458

```

```

0x1a3aa:    mov     eax, dword ptr [rip + 0x11748]
0x1a3b0:    test    eax, eax
0x1a3b2:    je      0x1a428
0x1a3b4:    mov     edx, dword ptr [rip + 0x1172a]
0x1a3ba:    lea     ecx, [rax - 1]
0x1a3bd:    cmp     ecx, edx
0x1a3bf:    je      0x1a4b0
0x1a3c5:    add     edx, 1
0x1a3c8:    mov     dword ptr [rip + 0x11716], edx
0x1a3ce:    mov     edi, 0x10
0x1a3d3:    call    0x1d670
0x1a3d8:    mov     rdi, rbp
0x1a3db:    mov     rbx, rax
0x1a3de:    call    0x49b0
0x1a3e3:    lea     rdi, [rax + 1]
0x1a3e7:    call    0x1d670
0x1a3ec:    mov     rsi, rbp
0x1a3ef:    mov     rdi, rax
0x1a3f2:    call    0x48b0
0x1a3f7:    movsxd  rdx, dword ptr [rip + 0x116e6]
0x1a3fe:    mov     qword ptr [rbx + 8], 0
0x1a406:    mov     qword ptr [rbx], rax
0x1a409:    mov     rax, qword ptr [rip + 0x116f0]
0x1a410:    mov     qword ptr [rax + rdx*8], 0
0x1a418:    mov     qword ptr [rax + rdx*8 - 8], rbx
0x1a41d:    add     rsp, 8
0x1a421:    pop     rbx
0x1a422:    pop     rbp
0x1a423:    ret
0x1a424:    nop     dword ptr [rax]
0x1a428:    mov     dword ptr [rip + 0x116c6], 0x32
0x1a432:    mov     edi, 0x190
0x1a437:    call    0x1d670
0x1a43c:    mov     dword ptr [rip + 0x1169e], 1
0x1a446:    mov     qword ptr [rip + 0x116b3], rax
0x1a44d:    jmp     0x1a3ce
0x1a3ce
0x1a452:    nop     word ptr [rax + rax]
0x1a458:    test    edx, edx
0x1a45a:    je      0x1a41d
0x1a45c:    mov     rax, qword ptr [rip + 0x1169d]
0x1a463:    mov     rax, qword ptr [rax]
0x1a466:    test    rax, rax
0x1a469:    je      0x1a489
0x1a46b:    mov     rdi, qword ptr [rax]
0x1a46e:    call    0x18d10
0x1a473:    mov     rax, qword ptr [rip + 0x11686]
0x1a47a:    mov     rdi, qword ptr [rax]

```

```

0x1a47d:      call    0x18d10
0x1a482:      movsxd  rdx, dword ptr [rip + 0x1165b]
0x1a489:      test   edx, edx
0x1a48b:      jle     0x1a4a1
0x1a48d:      mov     rdi, qword ptr [rip + 0x1166c]
0x1a494:      shl     rdx, 3
0x1a498:      lea     rsi, [rdi + 8]
0x1a49c:      call    0x4d40
0x1a4a1:      add     dword ptr [rip + 0x10340], 1
0x1a4a8:      jmp     0x1a3ce
0x1a3ce
0x1a4ad:      nop     dword ptr [rax]
0x1a4b0:      add     eax, 0x32
0x1a4b3:      mov     rdi, qword ptr [rip + 0x11646]
0x1a4ba:      mov     dword ptr [rip + 0x11638], eax
0x1a4c0:      cdqe
0x1a4c2:      lea     rsi, [rax*8]
0x1a4ca:      call    0x1d690
0x1a4cf:      mov     edx, dword ptr [rip + 0x1160f]
0x1a4d5:      mov     qword ptr [rip + 0x11624], rax
0x1a4dc:      jmp     0x1a3c5
0x1a3c5
0x1a4e1:      nop     word ptr cs:[rax + rax]
0x1a4ec:      nop     dword ptr [rax]
0x1a4f0:      endbr64
0x1a4f4:      push    r14
0x1a4f6:      xor     r14d, r14d
0x1a4f9:      push    r13
0x1a4fb:      mov     r13, rdx
0x1a4fe:      push    r12
0x1a500:      mov     r12, rsi
0x1a503:      push    rbp
0x1a504:      push    rbx
0x1a505:      movsxd  rbx, edi
0x1a508:      mov     edi, 0x10
0x1a50d:      call    0x1d670
0x1a512:      cmp     dword ptr [rip + 0x115cc], ebx
0x1a518:      jle     0x1a557
0x1a51a:      mov     rbp, rax
0x1a51d:      mov     rax, qword ptr [rip + 0x115dc]
0x1a524:      mov     rdi, r12
0x1a527:      mov     r14, qword ptr [rax + rbx*8]
0x1a52b:      call    0x49b0
0x1a530:      lea     rdi, [rax + 1]
0x1a534:      call    0x1d670
0x1a539:      mov     rsi, r12
0x1a53c:      mov     rdi, rax
0x1a53f:      call    0x48b0

```

```

0x1a544:    mov     qword ptr [rbp + 8], r13
0x1a548:    mov     qword ptr [rbp], rax
0x1a54c:    mov     rax, qword ptr [rip + 0x115ad]
0x1a553:    mov     qword ptr [rax + rbx*8], rbp
0x1a557:    pop     rbx
0x1a558:    mov     rax, r14
0x1a55b:    pop     rbp
0x1a55c:    pop     r12
0x1a55e:    pop     r13
0x1a560:    pop     r14
0x1a562:    ret
0x1a563:    nop     word ptr cs:[rax + rax]
0x1a56e:    nop
0x1a570:    endbr64
0x1a574:    mov     eax, dword ptr [rip + 0x11582]
0x1a57a:    ret
0x1a57b:    nop     dword ptr [rax + rax]
0x1a580:    endbr64
0x1a584:    xor     edx, edx
0x1a586:    xor     eax, eax
0x1a588:    jmp     0x19de0
0x19de0:
0x1a58d:    nop     dword ptr [rax]
0x1a590:    endbr64
0x1a594:    mov     edx, 1
0x1a599:    xor     eax, eax
0x1a59b:    jmp     0x19de0
0x19de0:
0x1a5a0:    endbr64
0x1a5a4:    push    r12
0x1a5a6:    push    rbx
0x1a5a7:    sub     rsp, 8
0x1a5ab:    mov     eax, dword ptr [rip + 0x11533]
0x1a5b1:    cmp     eax, edi
0x1a5b3:    jle     0x1a610
0x1a5b5:    test    eax, eax
0x1a5b7:    je      0x1a610
0x1a5b9:    lea     ebx, [rax - 1]
0x1a5bc:    movsxd  rdx, edi
0x1a5bf:    mov     rcx, qword ptr [rip + 0x1153a]
0x1a5c6:    mov     esi, ebx
0x1a5c8:    shl     rdx, 3
0x1a5cc:    sub     esi, edi
0x1a5ce:    cmp     eax, edi
0x1a5d0:    lea     r9, [rcx + rdx]
0x1a5d4:    mov     eax, 8
0x1a5d9:    lea     r8, [rsi*8 + 8]
0x1a5e1:    mov     r12, qword ptr [r9]

```



```

0x1a5e4:      mov     rdi, r9
0x1a5e7:      lea     rsi, [rcx + rdx + 8]
0x1a5ec:      cmovle  r8, rax
0x1a5f0:      mov     rdx, r8
0x1a5f3:      call    0x4d40
0x1a5f8:      mov     dword ptr [rip + 0x114e6], ebx
0x1a5fe:      add     rsp, 8
0x1a602:      mov     rax, r12
0x1a605:      pop     rbx
0x1a606:      pop     r12
0x1a608:      ret
0x1a609:      nop     dword ptr [rax]
0x1a610:      xor     r12d, r12d
0x1a613:      add     rsp, 8
0x1a617:      mov     rax, r12
0x1a61a:      pop     rbx
0x1a61b:      pop     r12
0x1a61d:      ret
0x1a61e:      nop
0x1a620:      endbr64
0x1a624:      push    r12
0x1a626:      mov     eax, dword ptr [rip + 0x114b8]
0x1a62c:      push    rbp
0x1a62d:      xor     ebp, ebp
0x1a62f:      test    edi, edi
0x1a631:      cmovns  ebp, edi
0x1a634:      push    rbx
0x1a635:      cmp     eax, ebp
0x1a637:      jle     0x1a6ce
0x1a63d:      sub     eax, ebp
0x1a63f:      mov     r12d, edi
0x1a642:      test    eax, eax
0x1a644:      jle     0x1a6e8
0x1a64a:      xor     ebx, ebx
0x1a64c:      nop     dword ptr [rax]
0x1a650:      mov     rax, qword ptr [rip + 0x114a9]
0x1a657:      mov     rax, qword ptr [rax + rbx*8]
0x1a65b:      mov     rdi, qword ptr [rax]
0x1a65e:      call    0x18d10
0x1a663:      mov     rax, qword ptr [rip + 0x11496]
0x1a66a:      mov     rdi, qword ptr [rax + rbx*8]
0x1a66e:      call    0x18d10
0x1a673:      mov     eax, dword ptr [rip + 0x1146b]
0x1a679:      lea     edx, [rbx + 1]
0x1a67c:      add     rbx, 1
0x1a680:      sub     eax, ebp
0x1a682:      cmp     eax, ebx
0x1a684:      jg      0x1a650

```

```

0x1a686:    mov     dword ptr [rip + 0x1015c], edx
0x1a68c:    mov     rcx, qword ptr [rip + 0x1146d]
0x1a693:    test    r12d, r12d
0x1a696:    jle     0x1a6f0
0x1a698:    cdqe
0x1a69a:    lea     rsi, [rcx + rax*8]
0x1a69e:    xor     eax, eax
0x1a6a0:    mov     rdx, qword ptr [rsi + rax*8]
0x1a6a4:    mov     qword ptr [rcx + rax*8], rdx
0x1a6a8:    add     rax, 1
0x1a6ac:    cmp     ebp, eax
0x1a6ae:    jg      0x1a6a0
0x1a6b0:    test    ebp, ebp
0x1a6b2:    mov     eax, 1
0x1a6b7:    cmovg   eax, ebp
0x1a6ba:    movsxd  rdx, eax
0x1a6bd:    lea     rcx, [rcx + rdx*8]
0x1a6c1:    mov     qword ptr [rcx], 0
0x1a6c8:    mov     dword ptr [rip + 0x11416], eax
0x1a6ce:    pop     rbx
0x1a6cf:    mov     dword ptr [rip + 0x11413], ebp
0x1a6d5:    mov     dword ptr [rip + 0x1140d], 1
0x1a6df:    pop     rbp
0x1a6e0:    pop     r12
0x1a6e2:    ret
0x1a6e3:    nop     dword ptr [rax + rax]
0x1a6e8:    xor     edx, edx
0x1a6ea:    jmp     0x1a686
0x1a686:
0x1a6ec:    nop     dword ptr [rax]
0x1a6f0:    xor     eax, eax
0x1a6f2:    jmp     0x1a6c1
0x1a6c1:
0x1a6f4:    nop     word ptr cs:[rax + rax]
0x1a6ff:    nop
0x1a700:    endbr64
0x1a704:    mov     edx, dword ptr [rip + 0x113e2]
0x1a70a:    mov     eax, dword ptr [rip + 0x113d8]
0x1a710:    test    edx, edx
0x1a712:    je      0x1a720
0x1a714:    mov     dword ptr [rip + 0x113ce], 0
0x1a71e:    neg     eax
0x1a720:    ret
0x1a721:    nop     word ptr cs:[rax + rax]
0x1a72c:    nop     dword ptr [rax]
0x1a730:    endbr64
0x1a734:    push    r15
0x1a736:    push    r14

```

```

0x1a738:    mov     r14d, esi
0x1a73b:    push    r13
0x1a73d:    push    r12
0x1a73f:    mov     r12d, edx
0x1a742:    push    rbp
0x1a743:    push    rbx
0x1a744:    sub     rsp, 0xa8
0x1a74b:    mov     rax, qword ptr fs:[0x28]
0x1a754:    mov     qword ptr [rsp + 0x98], rax
0x1a75c:    xor     eax, eax
0x1a75e:    call    0x1a080
0x1a763:    xor     esi, esi
0x1a765:    mov     edx, 0x1b6
0x1a76a:    mov     rbp, rax
0x1a76d:    mov     rdi, rax
0x1a770:    xor     eax, eax
0x1a772:    call    0x4d80
0x1a777:    test    eax, eax
0x1a779:    js      0x1a80f
0x1a77f:    mov     rsi, rsp
0x1a782:    mov     edi, eax
0x1a784:    mov     r15d, eax
0x1a787:    call    0x18ef0
0x1a78c:    cmp     eax, -1
0x1a78f:    je      0x1a807
0x1a791:    mov     eax, dword ptr [rsp + 0x30]
0x1a795:    lea     edi, [rax + 1]
0x1a798:    movsxd  rdi, edi
0x1a79b:    call    0x1d670
0x1a7a0:    mov     rdx, qword ptr [rsp + 0x30]
0x1a7a5:    mov     edi, r15d
0x1a7a8:    mov     rsi, rax
0x1a7ab:    mov     r13, rax
0x1a7ae:    call    0x47a0
0x1a7b3:    cmp     qword ptr [rsp + 0x30], rax
0x1a7b8:    je      0x1a819
0x1a7ba:    mov     edi, r15d
0x1a7bd:    call    0x4ab0
0x1a7c2:    mov     rdi, rbp
0x1a7c5:    call    0x18d10
0x1a7ca:    test    r13, r13
0x1a7cd:    je      0x1a7d7
0x1a7cf:    mov     rdi, r13
0x1a7d2:    call    0x18d10
0x1a7d7:    call    0x4860
0x1a7dc:    mov     eax, dword ptr [rax]
0x1a7de:    mov     rdx, qword ptr [rsp + 0x98]
0x1a7e6:    sub     rdx, qword ptr fs:[0x28]

```

```

0x1a7ef:      jne      0x1a8f0
0x1a7f5:      add      rsp, 0xa8
0x1a7fc:      pop      rbx
0x1a7fd:      pop      rbp
0x1a7fe:      pop      r12
0x1a800:      pop      r13
0x1a802:      pop      r14
0x1a804:      pop      r15
0x1a806:      ret
0x1a807:      mov      edi, r15d
0x1a80a:      call     0x4ab0
0x1a80f:      mov      rdi, rbp
0x1a812:      call     0x18d10
0x1a817:      jmp      0x1a7d7
0x1a7d7
0x1a819:      mov      edi, r15d
0x1a81c:      call     0x4ab0
0x1a821:      mov      rcx, qword ptr [rsp + 0x30]
0x1a826:      test     r12d, r12d
0x1a829:      jns      0x1a82e
0x1a82b:      mov      r12d, ecx
0x1a82e:      xor      ebx, ebx
0x1a830:      xor      esi, esi
0x1a832:      movsxd   rdx, esi
0x1a835:      cmp      ebx, r14d
0x1a838:      jge      0x1a885
0x1a83a:      cmp      rdx, rcx
0x1a83d:      jge      0x1a885
0x1a83f:      lea      eax, [rsi + 1]
0x1a842:      cdqe
0x1a844:      cmp      rdx, rcx
0x1a847:      jl       0x1a85c
0x1a849:      jmp      0x1a87b
0x1a87b
0x1a84b:      nop      dword ptr [rax + rax]
0x1a850:      lea      rdx, [rax + 1]
0x1a854:      cmp      rax, rcx
0x1a857:      je       0x1a878
0x1a859:      mov      rax, rdx
0x1a85c:      cmp      byte ptr [r13 + rax - 1], 0xa
0x1a862:      jne      0x1a850
0x1a864:      add      ebx, 1
0x1a867:      movsxd   rsi, eax
0x1a86a:      cmp      r14d, ebx
0x1a86d:      je       0x1a832
0x1a86f:      lea      rdx, [rax + 1]
0x1a873:      cmp      rax, rcx
0x1a876:      jne      0x1a859

```

```

0x1a878:      movsxd  rdx, esi
0x1a87b:      cmp     rcx, rdx
0x1a87e:      jle     0x1a885
0x1a880:      cmp     r14d, ebx
0x1a883:      jg      0x1a83f
0x1a885:      lea     r14, [r13 + rdx]
0x1a88a:      lea     r15d, [rsi + 1]
0x1a88e:      cmp     rcx, rdx
0x1a891:      jg      0x1a8aa
0x1a893:      jmp     0x1a8d4
0x1a8d4:
0x1a895:      nop     dword ptr [rax]
0x1a898:      movsxd  rax, r15d
0x1a89b:      add     r14, 1
0x1a89f:      add     r15d, 1
0x1a8a3:      cmp     rax, qword ptr [rsp + 0x30]
0x1a8a8:      jge     0x1a8d4
0x1a8aa:      cmp     byte ptr [r14], 0xa
0x1a8ae:      jne     0x1a898
0x1a8b0:      lea     rdi, [r13 + rsi]
0x1a8b5:      mov     byte ptr [r14], 0
0x1a8b9:      cmp     byte ptr [rdi], 0
0x1a8bc:      jne     0x1a8cb
0x1a8be:      add     ebx, 1
0x1a8c1:      cmp     r12d, ebx
0x1a8c4:      jle     0x1a8d4
0x1a8c6:      movsxd  rsi, r15d
0x1a8c9:      jmp     0x1a898
0x1a898:
0x1a8cb:      xor     eax, eax
0x1a8cd:      call    0x1a380
0x1a8d2:      jmp     0x1a8be
0x1a8be:
0x1a8d4:      mov     rdi, rbp
0x1a8d7:      call    0x18d10
0x1a8dc:      test    r13, r13
0x1a8df:      je      0x1a8e9
0x1a8e1:      mov     rdi, r13
0x1a8e4:      call    0x18d10
0x1a8e9:      xor     eax, eax
0x1a8eb:      jmp     0x1a7de
0x1a7de:
0x1a8f0:      call    0x49d0
0x1a8f5:      nop     word ptr cs:[rax + rax]
0x1a900:      endbr64
0x1a904:      mov     edx, 0xffffffff
0x1a909:      xor     esi, esi
0x1a90b:      xor     eax, eax

```

```

0x1a90d:      jmp      0x1a730
0x1a730
0x1a912:      nop      word ptr cs:[rax + rax]
0x1a91d:      nop      dword ptr [rax]
0x1a920:      endbr64
0x1a924:      push     r14
0x1a926:      push     r13
0x1a928:      push     r12
0x1a92a:      mov      r12d, esi
0x1a92d:      push     rbp
0x1a92e:      push     rbx
0x1a92f:      sub      rsp, 0xa0
0x1a936:      mov      rax, qword ptr fs:[0x28]
0x1a93f:      mov      qword ptr [rsp + 0x98], rax
0x1a947:      xor      eax, eax
0x1a949:      call     0x1a080
0x1a94e:      xor      esi, esi
0x1a950:      mov      edx, 0x1b6
0x1a955:      mov      rbp, rax
0x1a958:      mov      rdi, rax
0x1a95b:      xor      eax, eax
0x1a95d:      call     0x4d80
0x1a962:      cmp      eax, -1
0x1a965:      je       0x1aa58
0x1a96b:      mov      rsi, rsp
0x1a96e:      mov      edi, eax
0x1a970:      mov      r14d, eax
0x1a973:      call     0x18ef0
0x1a978:      cmp      eax, -1
0x1a97b:      je       0x1aa58
0x1a981:      mov      eax, dword ptr [rsp + 0x30]
0x1a985:      lea      edi, [rax + 1]
0x1a988:      movsxd   rdi, edi
0x1a98b:      call     0x1d670
0x1a990:      mov      rdx, qword ptr [rsp + 0x30]
0x1a995:      mov      edi, r14d
0x1a998:      mov      rsi, rax
0x1a99b:      mov      r13, rax
0x1a99e:      call     0x47a0
0x1a9a3:      mov      edi, r14d
0x1a9a6:      mov      rbx, rax
0x1a9a9:      call     0x4ab0
0x1a9ae:      test     ebx, ebx
0x1a9b0:      jle      0x1aa88
0x1a9b6:      lea      eax, [rbx - 1]
0x1a9b9:      cmp      ebx, 1
0x1a9bc:      je       0x1a9f3
0x1a9be:      test     r12d, r12d

```

```

0x1a9c1:      je      0x1a9f3
0x1a9c3:      lea     edx, [rbx - 2]
0x1a9c6:      movsxd  rdx, edx
0x1a9c9:      nop     dword ptr [rax]
0x1a9d0:      cmp     byte ptr [r13 + rdx + 1], 0xa
0x1a9d6:      jne     0x1a9dc
0x1a9d8:      sub     r12d, 1
0x1a9dc:      test    r12d, r12d
0x1a9df:      mov     eax, edx
0x1a9e1:      setne   sil
0x1a9e5:      test    edx, edx
0x1a9e7:      setne   cl
0x1a9ea:      sub     rdx, 1
0x1a9ee:      test    sil, cl
0x1a9f1:      jne     0x1a9d0
0x1a9f3:      test    eax, eax
0x1a9f5:      je      0x1aa88
0x1a9fb:      cdqe
0x1a9fd:      jmp     0x1aa08
0x1aa08:
0x1a9ff:      nop
0x1aa00:      sub     rax, 1
0x1aa04:      test    eax, eax
0x1aa06:      je      0x1aa50
0x1aa08:      cmp     byte ptr [r13 + rax], 0xa
0x1aa0e:      mov     ebx, eax
0x1aa10:      jne     0x1aa00
0x1aa12:      mov     edx, 0x1b6
0x1aa17:      mov     esi, 0x201
0x1aa1c:      mov     rdi, rbp
0x1aa1f:      xor     eax, eax
0x1aa21:      call    0x4d80
0x1aa26:      mov     r12d, eax
0x1aa29:      cmp     eax, -1
0x1aa2c:      je      0x1aa50
0x1aa2e:      mov     rdx, qword ptr [rsp + 0x30]
0x1aa33:      lea     esi, [rbx + 1]
0x1aa36:      mov     edi, eax
0x1aa38:      movsxd  rsi, esi
0x1aa3b:      sub     rdx, rsi
0x1aa3e:      add     rsi, r13
0x1aa41:      call    0x4790
0x1aa46:      mov     edi, r12d
0x1aa49:      call    0x4ab0
0x1aa4e:      nop
0x1aa50:      mov     rdi, r13
0x1aa53:      call    0x18d10
0x1aa58:      mov     rdi, rbp

```

```

0x1aa5b:      call    0x18d10
0x1aa60:      mov     rax, qword ptr [rsp + 0x98]
0x1aa68:      sub     rax, qword ptr fs:[0x28]
0x1aa71:      jne     0x1aa8f
0x1aa73:      add     rsp, 0xa0
0x1aa7a:      xor     eax, eax
0x1aa7c:      pop     rbx
0x1aa7d:      pop     rbp
0x1aa7e:      pop     r12
0x1aa80:      pop     r13
0x1aa82:      pop     r14
0x1aa84:      ret
0x1aa85:      nop     dword ptr [rax]
0x1aa88:      test    r13, r13
0x1aa8b:      je      0x1aa58
0x1aa8d:      jmp     0x1aa50
0x1aa50
0x1aa8f:      call    0x49d0
0x1aa94:      nop     word ptr cs:[rax + rax]
0x1aa9f:      nop
0x1aaa0:      endbr64
0x1aaa4:      mov     r8d, edi
0x1aaa7:      xor     edx, edx
0x1aaa9:      mov     rdi, rsi
0x1aaac:      xor     eax, eax
0x1aaae:      mov     esi, r8d
0x1aab1:      jmp     0x1a130
0x1a130
0x1aab6:      nop     word ptr cs:[rax + rax]
0x1aac0:      endbr64
0x1aac4:      mov     esi, dword ptr [rip + 0x1101a]
0x1aaca:      mov     edx, 1
0x1aacf:      xor     eax, eax
0x1aad1:      jmp     0x1a130
0x1a130
0x1aad6:      nop     word ptr cs:[rax + rax]
0x1aae0:      endbr64
0x1aae4:      movsxd  rdx, dword ptr [rip + 0x11011]
0x1aaeb:      cmp     edx, dword ptr [rip + 0x10ff3]
0x1aaf1:      je      0x1ab08
0x1aaf3:      mov     rax, qword ptr [rip + 0x11006]
0x1aafa:      test    rax, rax
0x1aafd:      je      0x1ab0a
0x1aaff:      mov     rax, qword ptr [rax + rdx*8]
0x1ab03:      ret
0x1ab04:      nop     dword ptr [rax]
0x1ab08:      xor     eax, eax
0x1ab0a:      ret

```



```

0x1ab0b:      nop      dword ptr [rax + rax]
0x1ab10:      endbr64
0x1ab14:      mov      eax, dword ptr [rip + 0x10fe2]
0x1ab1a:      xor      r8d, r8d
0x1ab1d:      test     eax, eax
0x1ab1f:      je       0x1ab37
0x1ab21:      sub      eax, 1
0x1ab24:      mov      rdx, qword ptr [rip + 0x10fd5]
0x1ab2b:      mov      dword ptr [rip + 0x10fcb], eax
0x1ab31:      cdqe
0x1ab33:      mov      r8, qword ptr [rdx + rax*8]
0x1ab37:      mov      rax, r8
0x1ab3a:      ret
0x1ab3b:      nop      dword ptr [rax + rax]
0x1ab40:      endbr64
0x1ab44:      mov      eax, dword ptr [rip + 0x10fb2]
0x1ab4a:      cmp      eax, dword ptr [rip + 0x10f94]
0x1ab50:      je       0x1ab70
0x1ab52:      add      eax, 1
0x1ab55:      mov      rdx, qword ptr [rip + 0x10fa4]
0x1ab5c:      mov      dword ptr [rip + 0x10f9a], eax
0x1ab62:      cdqe
0x1ab64:      mov      rax, qword ptr [rdx + rax*8]
0x1ab68:      ret
0x1ab69:      nop      dword ptr [rax]
0x1ab70:      xor      eax, eax
0x1ab72:      ret
0x1ab73:      nop      word ptr cs:[rax + rax]
0x1ab7e:      nop
0x1ab80:      endbr64
0x1ab84:      mov      rax, qword ptr [rip + 0x10f75]
0x1ab8b:      ret
0x1ab8c:      nop      dword ptr [rax]
0x1ab90:      endbr64
0x1ab94:      sub      edi, dword ptr [rip + 0xfc4e]
0x1ab9a:      cmp      dword ptr [rip + 0x10f44], edi
0x1aba0:      jle      0x1abc0
0x1aba2:      test     edi, edi
0x1aba4:      js       0x1abc0
0x1aba6:      mov      rax, qword ptr [rip + 0x10f53]
0x1abad:      test     rax, rax
0x1abb0:      je       0x1abc2
0x1abb2:      movsxd   rdi, edi
0x1abb5:      mov      rax, qword ptr [rax + rdi*8]
0x1abb9:      ret
0x1abba:      nop      word ptr [rax + rax]
0x1abc0:      xor      eax, eax
0x1abc2:      ret

```

```

0x1abc3:      nop      word ptr cs:[rax + rax]
0x1abce:      nop
0x1abd0:      endbr64
0x1abd4:      cmp      edx, dword ptr [rip + 0x10f0a]
0x1abda:      push     rbx
0x1abdb:      mov      ebx, dword ptr [rip + 0x10f1b]
0x1abe1:      jg       0x1abf7
0x1abe3:      test     edx, edx
0x1abe5:      js       0x1abf7
0x1abe7:      cmp      qword ptr [rip + 0x10f11], 0
0x1abef:      je       0x1abf7
0x1abf1:      mov      dword ptr [rip + 0x10f05], edx
0x1abf7:      xor      edx, edx
0x1abf9:      xor      eax, eax
0x1abfb:      call     0x19de0
0x1ac00:      mov      edx, ebx
0x1ac02:      shr      edx, 0x1f
0x1ac05:      cmp      eax, -1
0x1ac08:      je       0x1ac10
0x1ac0a:      mov      eax, dword ptr [rip + 0x10eec]
0x1ac10:      cmp      ebx, dword ptr [rip + 0x10ece]
0x1ac16:      jg       0x1ac2c
0x1ac18:      test     dl, dl
0x1ac1a:      jne      0x1ac2c
0x1ac1c:      cmp      qword ptr [rip + 0x10edc], 0
0x1ac24:      je       0x1ac2c
0x1ac26:      mov      dword ptr [rip + 0x10ed0], ebx
0x1ac2c:      pop      rbx
0x1ac2d:      ret
0x1ac2e:      nop
0x1ac30:      endbr64
0x1ac34:      cmp      dword ptr [rip + 0x10eaa], edi
0x1ac3a:      jl       0x1ac60
0x1ac3c:      test     edi, edi
0x1ac3e:      js       0x1ac60
0x1ac40:      cmp      qword ptr [rip + 0x10eb8], 0
0x1ac48:      je       0x1ac60
0x1ac4a:      mov      dword ptr [rip + 0x10eac], edi
0x1ac50:      mov      eax, 1
0x1ac55:      ret
0x1ac56:      nop      word ptr cs:[rax + rax]
0x1ac60:      xor      eax, eax
0x1ac62:      ret
0x1ac63:      nop      word ptr cs:[rax + rax]
0x1ac6e:      nop
0x1ac70:      endbr64
0x1ac74:      movsxd   rax, dword ptr [rsi]
0x1ac77:      movzx    r8d, byte ptr [rip + 0xfb6e]

```

```

0x1ac7f:      cmp     byte ptr [rdi + rax], r8b
0x1ac83:      jne     0x1afd0
0x1ac89:      push    r15
0x1ac8b:      lea     ecx, [rax + 1]
0x1ac8e:      mov     r9d, edx
0x1ac91:      mov     rdx, rax
0x1ac94:      push    r14
0x1ac96:      movsxd  r10, ecx
0x1ac99:      push    r13
0x1ac9b:      mov     r13, rsi
0x1ac9e:      lea     rsi, [rdi + r10]
0x1aca2:      push    r12
0x1aca4:      push    rbp
0x1aca5:      push    rbx
0x1aca6:      sub     rsp, 0x18
0x1acaa:      movzx   eax, byte ptr [rsi]
0x1acad:      cmp     r8b, al
0x1acb0:      je      0x1af20
0x1acb6:      mov     r8d, 1
0x1acbc:      cmp     al, 0x2d
0x1acbe:      je      0x1aef8
0x1acc4:      lea     edx, [rax - 0x30]
0x1acc7:      cmp     dl, 9
0x1acca:      jbe     0x1ae68
0x1acd0:      mov     ebp, 1
0x1acd5:      xor     ebx, ebx
0x1acd7:      cmp     al, 0x3f
0x1acd9:      jne     0x1acef
0x1acdb:      add     ecx, 1
0x1acde:      xor     ebp, ebp
0x1ace0:      mov     ebx, 1
0x1ace5:      movsxd  r10, ecx
0x1ace8:      lea     rsi, [rdi + r10]
0x1acec:      movzx   eax, byte ptr [rsi]
0x1acef:      test    al, al
0x1acf1:      je      0x1b042
0x1acf7:      lea     rdx, [rdi + r10 + 1]
0x1acfc:      mov     r12d, ecx
0x1acff:      mov     r8, rsi
0x1ad02:      movabs  r10, 0x400000100000200
0x1ad0c:      jmp     0x1ad48
0x1ad48:
0x1ad0e:      nop
0x1ad10:      cmp     al, 0x3a
0x1ad12:      ja      0x1ad1e
0x1ad14:      bt      r10, rax
0x1ad18:      jb      0x1afd8
0x1ad1e:      movsx   edi, al

```

```

0x1ad21:      cmp     edi, r9d
0x1ad24:      je      0x1b010
0x1ad2a:      cmp     al, 0xa
0x1ad2c:      je      0x1ae10
0x1ad32:      mov     r8, rdx
0x1ad35:      movzx   eax, byte ptr [rdx]
0x1ad38:      lea     rdx, [rdx + 1]
0x1ad3c:      add     r12d, 1
0x1ad40:      test    al, al
0x1ad42:      je      0x1ae10
0x1ad48:      test    ebx, ebx
0x1ad4a:      je      0x1ad10
0x1ad4c:      cmp     al, 0xa
0x1ad4e:      je      0x1ae10
0x1ad54:      cmp     al, 0x3f
0x1ad56:      jne     0x1ad32
0x1ad58:      mov     r15d, r12d
0x1ad5b:      mov     qword ptr [rsp + 8], r8
0x1ad60:      sub     r15d, ecx
0x1ad63:      mov     qword ptr [rsp], rsi
0x1ad67:      lea     edi, [r15 + 1]
0x1ad6b:      movsxd  r15, r15d
0x1ad6e:      movsxd  rdi, edi
0x1ad71:      call    0x1d670
0x1ad76:      mov     rsi, qword ptr [rsp]
0x1ad7a:      mov     rdx, r15
0x1ad7d:      mov     rdi, rax
0x1ad80:      mov     r14, rax
0x1ad83:      call    0x4880
0x1ad88:      mov     r8, qword ptr [rsp + 8]
0x1ad8d:      xor     eax, eax
0x1ad8f:      mov     byte ptr [r14 + r15], 0
0x1ad94:      cmp     byte ptr [r8], 0x3f
0x1ad98:      sete    al
0x1ad9b:      add     r12d, eax
0x1ad9e:      nop
0x1ada0:      mov     dword ptr [r13], r12d
0x1ada4:      mov     r12d, ebx
0x1ada7:      and     r12d, 1
0x1adab:      jmp     0x1add3
0x1add3:
0x1adad:      nop     dword ptr [rax]
0x1adb0:      movsxd  rax, dword ptr [rip + 0x10d45]
0x1adb7:      je      0x1af78
0x1adb d:      test    r12b, r12b
0x1adc0:      jne     0x1af78
0x1adc6:      test    eax, eax
0x1adc8:      je      0x1ade8

```

```

0x1adca:    sub     eax, 1
0x1adcd:    mov     dword ptr [rip + 0x10d29], eax
0x1add3:    xor     eax, eax
0x1add5:    mov     edx, ebp
0x1add7:    mov     esi, 0xffffffff
0x1addc:    mov     rdi, r14
0x1addf:    call    0x19de0
0x1ade4:    test    eax, eax
0x1ade6:    jns     0x1adb0
0x1ade8:    mov     eax, dword ptr [rip + 0x10cf6]
0x1adee:    mov     rdi, r14
0x1adf1:    mov     dword ptr [rip + 0x10d05], eax
0x1adf7:    call    0x18d10
0x1adfc:    add     rsp, 0x18
0x1ae00:    xor     eax, eax
0x1ae02:    pop     rbx
0x1ae03:    pop     rbp
0x1ae04:    pop     r12
0x1ae06:    pop     r13
0x1ae08:    pop     r14
0x1ae0a:    pop     r15
0x1ae0c:    ret
0x1ae0d:    nop     dword ptr [rax]
0x1ae10:    mov     r15d, r12d
0x1ae13:    sub     r15d, ecx
0x1ae16:    lea     edi, [r15 + 1]
0x1ae1a:    movsxd  r15, r15d
0x1ae1d:    movsxd  rdi, edi
0x1ae20:    mov     qword ptr [rsp + 8], r8
0x1ae25:    mov     qword ptr [rsp], rsi
0x1ae29:    call    0x1d670
0x1ae2e:    mov     rsi, qword ptr [rsp]
0x1ae32:    mov     rdx, r15
0x1ae35:    mov     rdi, rax
0x1ae38:    mov     r14, rax
0x1ae3b:    call    0x4880
0x1ae40:    test    ebx, ebx
0x1ae42:    mov     byte ptr [r14 + r15], 0
0x1ae47:    mov     r8, qword ptr [rsp + 8]
0x1ae4c:    je      0x1ada0
0x1ae52:    xor     eax, eax
0x1ae54:    cmp     byte ptr [r8], 0x3f
0x1ae58:    sete    al
0x1ae5b:    add     r12d, eax
0x1ae5e:    jmp     0x1ada0
0x1ada0:
0x1ae63:    nop     dword ptr [rax + rax]
0x1ae68:    lea     edx, [rcx + 1]

```

```

0x1ae6b:    xor     ecx, ecx
0x1ae6d:    movsxd  rdx, edx
0x1ae70:    sub     eax, 0x30
0x1ae73:    lea     ecx, [rcx + rcx*4]
0x1ae76:    mov     r9d, edx
0x1ae79:    movsx   eax, al
0x1ae7c:    lea     ecx, [rax + rcx*2]
0x1ae7f:    movzx   eax, byte ptr [rdi + rdx]
0x1ae83:    add     rdx, 1
0x1ae87:    lea     esi, [rax - 0x30]
0x1ae8a:    cmp     sil, 9
0x1ae8e:    jbe     0x1ae70
0x1ae90:    mov     dword ptr [r13], r9d
0x1ae94:    mov     eax, dword ptr [rip + 0x10c4a]
0x1ae9a:    mov     edx, dword ptr [rip + 0xf948]
0x1aea0:    cmp     r8d, -1
0x1aea4:    jne     0x1aead
0x1aea6:    lea     esi, [rax + rdx]
0x1aea9:    sub     esi, ecx
0x1aeab:    mov     ecx, esi
0x1aead:    sub     ecx, edx
0x1aeaf:    cmp     ecx, eax
0x1aeb1:    jge     0x1adfc
0x1aeb7:    test    ecx, ecx
0x1aeb9:    js      0x1adfc
0x1aebf:    mov     rax, qword ptr [rip + 0x10c3a]
0x1aec6:    test    rax, rax
0x1aec9:    je      0x1adfc
0x1aecf:    movsxd  rcx, ecx
0x1aed2:    mov     rax, qword ptr [rax + rcx*8]
0x1aed6:    test    rax, rax
0x1aed9:    je      0x1adfc
0x1aedef:    mov     rax, qword ptr [rax]
0x1aee2:    add     rsp, 0x18
0x1aee6:    pop     rbx
0x1aee7:    pop     rbp
0x1aee8:    pop     r12
0x1aeea:    pop     r13
0x1aeec:    pop     r14
0x1aeef:    pop     r15
0x1aef0:    ret
0x1aef1:    nop     dword ptr [rax]
0x1aef8:    lea     ecx, [rdx + 2]
0x1aefb:    mov     r8d, 0xffffffff
0x1af01:    movsxd  r10, ecx
0x1af04:    lea     rsi, [rdi + r10]
0x1af08:    movzx   eax, byte ptr [rsi]
0x1af0b:    lea     edx, [rax - 0x30]

```

```

0x1af0e:    cmp     dl, 9
0x1af11:    ja      0x1acd0
0x1af17:    jmp     0x1ae68
0x1ae68:
0x1af1c:    nop     dword ptr [rax]
0x1af20:    mov     ecx, dword ptr [rip + 0x10bbe]
0x1af26:    mov     eax, dword ptr [rip + 0xf8bc]
0x1af2c:    add     edx, 2
0x1af2f:    mov     dword ptr [r13], edx
0x1af33:    lea     eax, [rcx + rax - 1]
0x1af37:    sub     eax, dword ptr [rip + 0xf8ab]
0x1af3d:    cmp     eax, dword ptr [rip + 0x10ba1]
0x1af43:    jge     0x1adfc
0x1af49:    test    eax, eax
0x1af4b:    js      0x1adfc
0x1af51:    mov     rdx, qword ptr [rip + 0x10ba8]
0x1af58:    test    rdx, rdx
0x1af5b:    je      0x1adfc
0x1af61:    cdqe
0x1af63:    mov     rax, qword ptr [rdx + rax*8]
0x1af67:    test    rax, rax
0x1af6a:    jne     0x1aedef
0x1af70:    jmp     0x1adfc
0x1adfc:
0x1af75:    nop     dword ptr [rax]
0x1af78:    mov     edx, dword ptr [rip + 0x10b66]
0x1af7e:    cmp     edx, eax
0x1af80:    je      0x1b03b
0x1af86:    mov     rbp, qword ptr [rip + 0x10b73]
0x1af8d:    test    rbp, rbp
0x1af90:    je      0x1af97
0x1af92:    mov     rbp, qword ptr [rbp + rax*8]
0x1af97:    mov     dword ptr [rip + 0x10b5f], edx
0x1af9d:    test    ebx, ebx
0x1af9f:    je      0x1b02e
0x1afa5:    mov     rdi, qword ptr [rip + 0x10b44]
0x1afac:    test    rdi, rdi
0x1afaf:    je      0x1afb6
0x1afb1:    call    0x18d10
0x1afb6:    mov     qword ptr [rip + 0x10b33], r14
0x1afbd:    mov     rax, qword ptr [rbp]
0x1afc1:    add     rsp, 0x18
0x1afc5:    pop     rbx
0x1afc6:    pop     rbp
0x1afc7:    pop     r12
0x1afc9:    pop     r13
0x1afcb:    pop     r14
0x1afcd:    pop     r15

```

```

0x1afcf:      ret
0x1afd0:      xor     eax, eax
0x1afd2:      ret
0x1afd3:      nop     dword ptr [rax + rax]
0x1afd8:      mov     eax, r12d
0x1afdb:      mov     qword ptr [rsp], rsi
0x1afdf:      sub     eax, ecx
0x1afe1:      movsxd   r15, eax
0x1afe4:      add     eax, 1
0x1afe7:      movsxd   rdi, eax
0x1afea:      call    0x1d670
0x1afef:      mov     r14, rax
0x1aff2:      mov     rsi, qword ptr [rsp]
0x1aff6:      mov     rdx, r15
0x1aff9:      mov     rdi, r14
0x1affc:      call    0x4880
0x1b001:      mov     byte ptr [r14 + r15], 0
0x1b006:      jmp     0x1ada0
0x1ada0
0x1b00b:      nop     dword ptr [rax + rax]
0x1b010:      mov     r15d, r12d
0x1b013:      mov     qword ptr [rsp], rsi
0x1b017:      sub     r15d, ecx
0x1b01a:      lea     edi, [r15 + 1]
0x1b01e:      movsxd   r15, r15d
0x1b021:      movsxd   rdi, edi
0x1b024:      call    0x1d670
0x1b029:      mov     r14, rax
0x1b02c:      jmp     0x1aff2
0x1aff2
0x1b02e:      mov     rdi, r14
0x1b031:      call    0x18d10
0x1b036:      jmp     0x1afbd
0x1afbd
0x1b03b:      xor     ebp, ebp
0x1b03d:      jmp     0x1af97
0x1af97
0x1b042:      mov     r12d, ecx
0x1b045:      mov     r8, rsi
0x1b048:      xor     r15d, r15d
0x1b04b:      mov     edi, 1
0x1b050:      jmp     0x1ae20
0x1ae20
0x1b055:      nop     word ptr cs:[rax + rax]
0x1b060:      endbr64
0x1b064:      push    r15
0x1b066:      push    r14
0x1b068:      push    r13

```



```

0x1b06a:    push    r12
0x1b06c:    push    rbp
0x1b06d:    push    rbx
0x1b06e:    sub     rsp, 0x28
0x1b072:    movzx   r14d, byte ptr [rdi]
0x1b076:    test    r14b, r14b
0x1b079:    je      0x1b30d
0x1b07f:    mov     qword ptr [rsp], 0
0x1b087:    mov     r12, rdi
0x1b08a:    mov     r13, rdi
0x1b08d:    xor     ebx, ebx
0x1b08f:    mov     dword ptr [rsp + 8], 0
0x1b097:    xor     ebp, ebp
0x1b099:    mov     dword ptr [rsp + 0xc], 0
0x1b0a1:    jmp     0x1b0c0
0x1b0c0:
0x1b0a3:    nop     dword ptr [rax + rax]
0x1b0a8:    add     ebx, 1
0x1b0ab:    movsxd  rbp, ebx
0x1b0ae:    lea     r13, [r12 + rbp]
0x1b0b2:    movzx   r14d, byte ptr [r13]
0x1b0b7:    test    r14b, r14b
0x1b0ba:    je      0x1b315
0x1b0c0:    cmp     r14b, 0x20
0x1b0c4:    je      0x1b0a8
0x1b0c6:    cmp     r14b, 9
0x1b0ca:    je      0x1b0a8
0x1b0cc:    cmp     byte ptr [rip + 0x10a0d], r14b
0x1b0d3:    je      0x1b315
0x1b0d9:    movsx   r15d, r14b
0x1b0dd:    lea     rdi, [rip + 0x7379]
0x1b0e4:    mov     esi, r15d
0x1b0e7:    call    0x4a00
0x1b0ec:    test    rax, rax
0x1b0ef:    je      0x1b190
0x1b0f5:    add     ebx, 1
0x1b0f8:    mov     r14d, 1
0x1b0fe:    mov     r9d, 2
0x1b104:    movsxd  rbp, ebx
0x1b107:    lea     rcx, [r12 + rbp]
0x1b10b:    mov     eax, dword ptr [rsp + 8]
0x1b10f:    add     eax, 2
0x1b112:    cmp     eax, dword ptr [rsp + 0xc]
0x1b116:    jge     0x1b2a0
0x1b11c:    movsxd  r8, dword ptr [rsp + 8]
0x1b121:    mov     rdx, qword ptr [rsp]
0x1b125:    mov     rdi, r9
0x1b128:    mov     qword ptr [rsp + 0x18], rcx

```

```

0x1b12d:    shl     r8, 3
0x1b131:    lea     r15, [rdx + r8]
0x1b135:    mov     qword ptr [rsp + 0x10], r8
0x1b13a:    call    0x1d670
0x1b13f:    mov     rdx, r14
0x1b142:    mov     rsi, r13
0x1b145:    mov     qword ptr [r15], rax
0x1b148:    mov     rdi, rax
0x1b14b:    call    0x4880
0x1b150:    mov     rax, qword ptr [r15]
0x1b153:    mov     r8, qword ptr [rsp + 0x10]
0x1b158:    mov     rcx, qword ptr [rsp + 0x18]
0x1b15d:    add     dword ptr [rsp + 8], 1
0x1b162:    mov     byte ptr [rax + r14], 0
0x1b167:    mov     rax, qword ptr [rsp]
0x1b16b:    mov     qword ptr [rax + r8 + 8], 0
0x1b174:    movzx   r14d, byte ptr [rcx]
0x1b178:    test    r14b, r14b
0x1b17b:    je      0x1b315
0x1b181:    mov     r13, rcx
0x1b184:    jmp     0x1b0c0
0x1b0c0:
0x1b189:    nop     dword ptr [rax]
0x1b190:    mov     esi, r15d
0x1b193:    lea     rdi, [rip + 0x72c7]
0x1b19a:    call    0x4a00
0x1b19f:    test    rax, rax
0x1b1a2:    je      0x1b369
0x1b1a8:    movsx   edx, byte ptr [r12 + rbp + 1]
0x1b1ae:    cmp     dl, r14b
0x1b1b1:    je      0x1b2d8
0x1b1b7:    cmp     edx, 0x26
0x1b1ba:    je      0x1b340
0x1b1c0:    cmp     edx, 0x3e
0x1b1c3:    jne     0x1b3ac
0x1b1c9:    cmp     r14b, 0x26
0x1b1cd:    je      0x1b34e
0x1b1d3:    cmp     r14b, 0x24
0x1b1d7:    jne     0x1b0f5
0x1b1dd:    mov     edx, ebx
0x1b1df:    xor     r15d, r15d
0x1b1e2:    movsxd  r8, edx
0x1b1e5:    lea     rcx, [r12 + r8]
0x1b1e9:    mov     rbp, r8
0x1b1ec:    movzx   eax, byte ptr [rcx]
0x1b1ef:    test    al, al
0x1b1f1:    je      0x1b404
0x1b1f7:    mov     qword ptr [rsp + 0x10], r13

```

```

0x1b1fc:    lea     r14, [rip + 0x7265]
0x1b203:    mov     r13, r8
0x1b206:    mov     dword ptr [rsp + 0x18], ebx
0x1b20a:    mov     rbx, r12
0x1b20d:    mov     r12d, edx
0x1b210:    jmp     0x1b265
0x1b265:
0x1b212:    nop     word ptr [rax + rax]
0x1b218:    movsx   ebp, al
0x1b21b:    test    r15d, r15d
0x1b21e:    jne     0x1b290
0x1b220:    mov     esi, ebp
0x1b222:    lea     rdi, [rip + 0x7243]
0x1b229:    call    0x4a00
0x1b22e:    test    rax, rax
0x1b231:    jne     0x1b3c0
0x1b237:    mov     esi, ebp
0x1b239:    mov     rdi, r14
0x1b23c:    call    0x4a00
0x1b241:    lea     esi, [r12 + 1]
0x1b246:    test    rax, rax
0x1b249:    cmovne  r15d, ebp
0x1b24d:    movsxd  rbp, esi
0x1b250:    mov     r12d, esi
0x1b253:    lea     rcx, [rbx + rbp]
0x1b257:    mov     r13, rbp
0x1b25a:    movzx   eax, byte ptr [rcx]
0x1b25d:    test    al, al
0x1b25f:    je      0x1b389
0x1b265:    cmp     al, 0x5c
0x1b267:    jne     0x1b218
0x1b269:    cmp     byte ptr [rbx + r13 + 1], 0xa
0x1b26f:    lea     esi, [r12 + 1]
0x1b274:    je      0x1b330
0x1b27a:    xor     eax, eax
0x1b27c:    cmp     r15d, 0x27
0x1b280:    setne   al
0x1b283:    add     esi, eax
0x1b285:    jmp     0x1b24d
0x1b24d:
0x1b287:    nop     word ptr [rax + rax]
0x1b290:    xor     eax, eax
0x1b292:    cmp     r15d, ebp
0x1b295:    lea     esi, [r12 + 1]
0x1b29a:    cmovne  r15d, eax
0x1b29e:    jmp     0x1b24d
0x1b24d:
0x1b2a0:    add     dword ptr [rsp + 0xc], 0xa

```

```

0x1b2a5:      movsxd  rsi, dword ptr [rsp + 0xc]
0x1b2aa:      mov     rdi, qword ptr [rsp]
0x1b2ae:      mov     qword ptr [rsp + 0x18], rcx
0x1b2b3:      shl     rsi, 3
0x1b2b7:      mov     qword ptr [rsp + 0x10], r9
0x1b2bc:      call    0x1d690
0x1b2c1:      mov     rcx, qword ptr [rsp + 0x18]
0x1b2c6:      mov     r9, qword ptr [rsp + 0x10]
0x1b2cb:      mov     qword ptr [rsp], rax
0x1b2cf:      jmp     0x1b11c
0x1b11c:
0x1b2d4:      nop     dword ptr [rax]
0x1b2d8:      cmp     edx, 0x24
0x1b2db:      je      0x1b1d3
0x1b2e1:      mov     eax, ebx
0x1b2e3:      cmp     edx, 0x3c
0x1b2e6:      je      0x1b3f0
0x1b2ec:      add     eax, 2
0x1b2ef:      mov     r14d, eax
0x1b2f2:      movsxd  rbp, eax
0x1b2f5:      sub     r14d, ebx
0x1b2f8:      lea     rcx, [r12 + rbp]
0x1b2fc:      mov     ebx, eax
0x1b2fe:      lea     r9d, [r14 + 1]
0x1b302:      movsxd  r14, r14d
0x1b305:      movsxd  r9, r9d
0x1b308:      jmp     0x1b10b
0x1b10b:
0x1b30d:      mov     qword ptr [rsp], 0
0x1b315:      mov     rax, qword ptr [rsp]
0x1b319:      add     rsp, 0x28
0x1b31d:      pop     rbx
0x1b31e:      pop     rbp
0x1b31f:      pop     r12
0x1b321:      pop     r13
0x1b323:      pop     r14
0x1b325:      pop     r15
0x1b327:      ret
0x1b328:      nop     dword ptr [rax + rax]
0x1b330:      lea     esi, [r12 + 2]
0x1b335:      jmp     0x1b24d
0x1b24d:
0x1b33a:      nop     word ptr [rax + rax]
0x1b340:      mov     eax, r14d
0x1b343:      and     eax, 0xffffffff
0x1b346:      cmp     al, 0x3c
0x1b348:      jne     0x1b1d3
0x1b34e:      add     ebx, 2

```

```

0x1b351:      mov     r14d, 2
0x1b357:      mov     r9d, 3
0x1b35d:      movsxd  rbp, ebx
0x1b360:      lea     rcx, [r12 + rbp]
0x1b364:      jmp     0x1b10b
0x1b10b
0x1b369:      mov     esi, r15d
0x1b36c:      lea     rdi, [rip + 0x70f5]
0x1b373:      call    0x4a00
0x1b378:      test    rax, rax
0x1b37b:      je      0x1b1dd
0x1b381:      lea     edx, [rbx + 1]
0x1b384:      jmp     0x1b1e2
0x1b1e2
0x1b389:      mov     r12, rbx
0x1b38c:      mov     ebx, dword ptr [rsp + 0x18]
0x1b390:      mov     r14d, esi
0x1b393:      mov     r13, qword ptr [rsp + 0x10]
0x1b398:      sub     r14d, ebx
0x1b39b:      mov     ebx, esi
0x1b39d:      lea     r9d, [r14 + 1]
0x1b3a1:      movsxd  r14, r14d
0x1b3a4:      movsxd  r9, r9d
0x1b3a7:      jmp     0x1b10b
0x1b10b
0x1b3ac:      cmp     r14b, 0x24
0x1b3b0:      jne     0x1b0f5
0x1b3b6:      cmp     edx, 0x28
0x1b3b9:      je      0x1b34e
0x1b3bb:      jmp     0x1b1d3
0x1b1d3
0x1b3c0:      mov     edx, r12d
0x1b3c3:      mov     r12, rbx
0x1b3c6:      mov     ebx, dword ptr [rsp + 0x18]
0x1b3ca:      mov     r8, r13
0x1b3cd:      mov     r14d, edx
0x1b3d0:      mov     r13, qword ptr [rsp + 0x10]
0x1b3d5:      mov     rbp, r8
0x1b3d8:      lea     rcx, [r12 + r8]
0x1b3dc:      sub     r14d, ebx
0x1b3df:      mov     ebx, edx
0x1b3e1:      lea     r9d, [r14 + 1]
0x1b3e5:      movsxd  r14, r14d
0x1b3e8:      movsxd  r9, r9d
0x1b3eb:      jmp     0x1b10b
0x1b10b
0x1b3f0:      cmp     byte ptr [r12 + rbp + 2], 0x2d
0x1b3f6:      jne     0x1b2ec

```

```

0x1b3fc:    lea     eax, [rbx + 1]
0x1b3ff:    jmp     0x1b2ec
0x1b2ec:
0x1b404:    mov     r14d, edx
0x1b407:    sub     r14d, ebx
0x1b40a:    mov     ebx, edx
0x1b40c:    lea     r9d, [r14 + 1]
0x1b410:    movsxd  r14, r14d
0x1b413:    movsxd  r9, r9d
0x1b416:    jmp     0x1b10b
0x1b10b:
0x1b41b:    nop     dword ptr [rax + rax]
0x1b420:    endbr64
0x1b424:    push    r15
0x1b426:    xor     eax, eax
0x1b428:    push    r14
0x1b42a:    push    r13
0x1b42c:    push    r12
0x1b42e:    mov     r12d, esi
0x1b431:    push    rbp
0x1b432:    push    rbx
0x1b433:    mov     ebx, edi
0x1b435:    mov     rdi, rdx
0x1b438:    sub     rsp, 8
0x1b43c:    call    0x1b060
0x1b441:    test    rax, rax
0x1b444:    je      0x1b598
0x1b44a:    cmp     qword ptr [rax], 0
0x1b44e:    mov     rbp, rax
0x1b451:    je      0x1b590
0x1b457:    mov     eax, 1
0x1b45c:    nop     dword ptr [rax]
0x1b460:    mov     r14d, eax
0x1b463:    add     rax, 1
0x1b467:    cmp     qword ptr [rbp + rax*8 - 8], 0
0x1b46d:    jne     0x1b460
0x1b46f:    lea     eax, [r14 + r12 - 1]
0x1b474:    test    r12d, r12d
0x1b477:    cmovs   r12d, eax
0x1b47b:    lea     eax, [r14 + rbx - 1]
0x1b480:    test    ebx, ebx
0x1b482:    cmovs   ebx, eax
0x1b485:    lea     eax, [r12 + 1]
0x1b48a:    cmp     r12d, 0x24
0x1b48e:    jne     0x1b497
0x1b490:    lea     r12d, [r14 - 1]
0x1b494:    mov     eax, r14d
0x1b497:    lea     edx, [r14 - 1]

```

```

0x1b49b:    cmp     ebx, 0x24
0x1b49e:    cmovbe ebx, edx
0x1b4a1:    mov     edx, eax
0x1b4a3:    cmp     ebx, eax
0x1b4a5:    cmovge  edx, ebx
0x1b4a8:    xor     r13d, r13d
0x1b4ab:    cmp     edx, r14d
0x1b4ae:    jg      0x1b547
0x1b4b4:    mov     ecx, ebx
0x1b4b6:    or      ecx, eax
0x1b4b8:    js      0x1b547
0x1b4be:    cmp     ebx, eax
0x1b4c0:    jge     0x1b59d
0x1b4c6:    movsxd  rbx, ebx
0x1b4c9:    xor     r15d, r15d
0x1b4cc:    mov     r13, rbx
0x1b4cf:    nop
0x1b4d0:    mov     rdi, qword ptr [rbp + r13*8]
0x1b4d5:    add     r13, 1
0x1b4d9:    call    0x49b0
0x1b4de:    lea     r15d, [r15 + rax + 1]
0x1b4e3:    lea     eax, [r13 - 1]
0x1b4e7:    cmp     r12d, eax
0x1b4ea:    jg      0x1b4d0
0x1b4ec:    lea     edi, [r15 + 1]
0x1b4f0:    xor     r15d, r15d
0x1b4f3:    movsxd  rdi, edi
0x1b4f6:    call    0x1d670
0x1b4fb:    mov     r13, rax
0x1b4fe:    nop
0x1b500:    mov     rsi, qword ptr [rbp + rbx*8]
0x1b505:    movsxd  rdi, r15d
0x1b508:    add     rdi, r13
0x1b50b:    call    0x48b0
0x1b510:    mov     rdi, qword ptr [rbp + rbx*8]
0x1b515:    call    0x49b0
0x1b51a:    add     r15d, eax
0x1b51d:    cmp     r12d, ebx
0x1b520:    jle     0x1b53b
0x1b522:    lea     eax, [r15 + 1]
0x1b526:    movsxd  r15, r15d
0x1b529:    movsxd  rdx, eax
0x1b52c:    mov     byte ptr [r13 + r15], 0x20
0x1b532:    mov     r15d, eax
0x1b535:    mov     byte ptr [r13 + rdx], 0
0x1b53b:    add     rbx, 1
0x1b53f:    lea     eax, [rbx - 1]
0x1b542:    cmp     r12d, eax

```

```

0x1b545:      jg      0x1b500
0x1b547:      test   r14d, r14d
0x1b54a:      je      0x1b571
0x1b54c:      lea     eax, [r14 - 1]
0x1b550:      mov     rbx, rbp
0x1b553:      lea     r12, [rbp + rax*8 + 8]
0x1b558:      nop     dword ptr [rax + rax]
0x1b560:      mov     rdi, qword ptr [rbx]
0x1b563:      add     rbx, 8
0x1b567:      call    0x18d10
0x1b56c:      cmp     r12, rbx
0x1b56f:      jne     0x1b560
0x1b571:      mov     rdi, rbp
0x1b574:      call    0x18d10
0x1b579:      add     rsp, 8
0x1b57d:      mov     rax, r13
0x1b580:      pop     rbx
0x1b581:      pop     rbp
0x1b582:      pop     r12
0x1b584:      pop     r13
0x1b586:      pop     r14
0x1b588:      pop     r15
0x1b58a:      ret
0x1b58b:      nop     dword ptr [rax + rax]
0x1b590:      xor     r14d, r14d
0x1b593:      jmp     0x1b46f
0x1b46f
0x1b598:      xor     r13d, r13d
0x1b59b:      jmp     0x1b579
0x1b579
0x1b59d:      mov     edi, 1
0x1b5a2:      call    0x1d670
0x1b5a7:      mov     r13, rax
0x1b5aa:      jmp     0x1b547
0x1b547
0x1b5ac:      nop     dword ptr [rax]
0x1b5b0:      endbr64
0x1b5b4:      push    r15
0x1b5b6:      push    r14
0x1b5b8:      push    r13
0x1b5ba:      push    r12
0x1b5bc:      push    rbp
0x1b5bd:      push    rbx
0x1b5be:      mov     rbx, rdi
0x1b5c1:      sub     rsp, 0x88
0x1b5c8:      mov     qword ptr [rsp + 0x20], rdi
0x1b5cd:      mov     qword ptr [rsp + 0x30], rsi
0x1b5d2:      mov     rax, qword ptr fs:[0x28]

```



```

0x1b5db:    mov     qword ptr [rsp + 0x78], rax
0x1b5e0:    xor     eax, eax
0x1b5e2:    cmp     byte ptr [rip + 0xf204], 0
0x1b5e9:    je      0x1b656
0x1b5eb:    mov     edi, 0x100
0x1b5f0:    call    0x1d670
0x1b5f5:    mov     rbx, qword ptr [rsp + 0x20]
0x1b5fa:    mov     byte ptr [rax], 0
0x1b5fd:    mov     r15, rax
0x1b600:    mov     rdi, rbx
0x1b603:    call    0x49b0
0x1b608:    mov     dword ptr [rsp + 0xc], eax
0x1b60c:    mov     r12, rax
0x1b60f:    movzx   eax, byte ptr [rbx]
0x1b612:    cmp     al, byte ptr [rip + 0xf1d4]
0x1b618:    je      0x1c1e0
0x1b61e:    movzx   ebp, byte ptr [rip + 0xf1c8]
0x1b625:    mov     r13, qword ptr [rip + 0xf1c4]
0x1b62c:    lea     rbx, [rbx + 1]
0x1b630:    test    al, al
0x1b632:    jne     0x1b6c7
0x1b638:    nop     dword ptr [rax + rax]
0x1b640:    test    bpl, bpl
0x1b643:    je      0x1b6d8
0x1b649:    mov     rdi, r15
0x1b64c:    call    0x18d10
0x1b651:    mov     rbx, qword ptr [rsp + 0x20]
0x1b656:    mov     rdi, rbx
0x1b659:    call    0x49b0
0x1b65e:    lea     rdi, [rax + 1]
0x1b662:    call    0x1d670
0x1b667:    mov     rsi, rbx
0x1b66a:    mov     rdi, rax
0x1b66d:    call    0x48b0
0x1b672:    mov     rcx, qword ptr [rsp + 0x30]
0x1b677:    mov     qword ptr [rcx], rax
0x1b67a:    xor     eax, eax
0x1b67c:    mov     rdx, qword ptr [rsp + 0x78]
0x1b681:    sub     rdx, qword ptr fs:[0x28]
0x1b68a:    jne     0x1c487
0x1b690:    add     rsp, 0x88
0x1b697:    pop     rbx
0x1b698:    pop     rbp
0x1b699:    pop     r12
0x1b69b:    pop     r13
0x1b69d:    pop     r14
0x1b69f:    pop     r15
0x1b6a1:    ret

```

```

0x1b6a2:      nop      word ptr [rax + rax]
0x1b6a8:      mov      rdi, r13
0x1b6ab:      call     0x4a00
0x1b6b0:      test     rax, rax
0x1b6b3:      je        0x1b6d8
0x1b6b5:      nop      dword ptr [rax]
0x1b6b8:      movzx     eax, byte ptr [rbx]
0x1b6bb:      add      rbx, 1
0x1b6bf:      test     al, al
0x1b6c1:      je        0x1b640
0x1b6c7:      movsx     esi, byte ptr [rbx]
0x1b6ca:      cmp      al, bpl
0x1b6cd:      jne       0x1b6b8
0x1b6cf:      test     esi, esi
0x1b6d1:      je        0x1b6b8
0x1b6d3:      jmp       0x1b6a8
0x1b6a8:
0x1b6d5:      nop      dword ptr [rax]
0x1b6d8:      mov      rbp, qword ptr [rsp + 0x20]
0x1b6dd:      test     r12d, r12d
0x1b6e0:      jle       0x1c40a
0x1b6e6:      mov      dword ptr [rsp + 4], 0x100
0x1b6ee:      xor      r13d, r13d
0x1b6f1:      xor      ebx, ebx
0x1b6f3:      mov      dword ptr [rsp + 0x28], 0
0x1b6fb:      mov      dword ptr [rsp + 0x3c], 0
0x1b703:      mov      dword ptr [rsp + 8], 0
0x1b70b:      jmp       0x1b758
0x1b758:
0x1b70d:      nop      dword ptr [rax]
0x1b710:      cmp      r12b, byte ptr [rip + 0xf0d6]
0x1b717:      je        0x1b7d0
0x1b71d:      cmp      r12b, 0xfd
0x1b721:      je        0x1b7d0
0x1b727:      cmp      r12b, 0x5c
0x1b72b:      je        0x1bde0
0x1b731:      cmp      ebx, dword ptr [rsp + 4]
0x1b735:      jge       0x1bfe0
0x1b73b:      lea      eax, [rbx + 1]
0x1b73e:      mov      byte ptr [r15 + rbx], r12b
0x1b742:      movsxd   rdx, eax
0x1b745:      movsxd   rbx, eax
0x1b748:      mov      byte ptr [r15 + rdx], 0
0x1b74d:      add      r13d, 1
0x1b751:      cmp      r13d, dword ptr [rsp + 0xc]
0x1b756:      jge       0x1b79d
0x1b758:      movsxd   r14, r13d
0x1b75b:      mov      r10d, dword ptr [rsp + 8]

```

```

0x1b760:    lea     rdx, [rbp + r14]
0x1b765:    movzx   r12d, byte ptr [rdx]
0x1b769:    test    r10d, r10d
0x1b76c:    je      0x1b710
0x1b76e:    cmp     ebx, dword ptr [rsp + 4]
0x1b772:    jge     0x1be10
0x1b778:    lea     eax, [rbx + 1]
0x1b77b:    mov     byte ptr [r15 + rbx], r12b
0x1b77f:    add     r13d, 1
0x1b783:    movsxd  rdx, eax
0x1b786:    mov     dword ptr [rsp + 8], 0
0x1b78e:    movsxd  rbx, eax
0x1b791:    mov     byte ptr [r15 + rdx], 0
0x1b796:    cmp     r13d, dword ptr [rsp + 0xc]
0x1b79b:    jl      0x1b758
0x1b79d:    mov     rax, qword ptr [rsp + 0x30]
0x1b7a2:    mov     qword ptr [rax], r15
0x1b7a5:    cmp     qword ptr [rsp + 0x20], rbp
0x1b7aa:    je      0x1b7b4
0x1b7ac:    mov     rdi, rbp
0x1b7af:    call    0x18d10
0x1b7b4:    mov     ecx, dword ptr [rsp + 0x28]
0x1b7b8:    test    ecx, ecx
0x1b7ba:    jne     0x1c35b
0x1b7c0:    mov     edx, dword ptr [rsp + 0x3c]
0x1b7c4:    xor     eax, eax
0x1b7c6:    test    edx, edx
0x1b7c8:    setne   al
0x1b7cb:    jmp     0x1b67c
0x1b67c:
0x1b7d0:    lea     rcx, [rbp + r14 + 1]
0x1b7d5:    mov     qword ptr [rsp + 0x10], rdx
0x1b7da:    movsx   esi, byte ptr [rcx]
0x1b7dd:    test    esi, esi
0x1b7df:    je      0x1b830
0x1b7e1:    mov     rdi, qword ptr [rip + 0xf008]
0x1b7e8:    mov     qword ptr [rsp + 0x18], rcx
0x1b7ed:    call    0x4a00
0x1b7f2:    mov     rcx, qword ptr [rsp + 0x18]
0x1b7f7:    test    rax, rax
0x1b7fa:    je      0x1b830
0x1b7fc:    cmp     ebx, dword ptr [rsp + 4]
0x1b800:    jl      0x1b73b
0x1b806:    add     dword ptr [rsp + 4], 0x40
0x1b80b:    movsxd  rsi, dword ptr [rsp + 4]
0x1b810:    mov     rdi, r15
0x1b813:    call    0x1d690
0x1b818:    mov     rdx, qword ptr [rsp + 0x10]

```

```

0x1b81d:      mov     r15, rax
0x1b820:      movzx   r12d, byte ptr [rdx]
0x1b824:      jmp     0x1b73b
0x1b73b:
0x1b829:      nop     dword ptr [rax]
0x1b830:      mov     edi, 0x80
0x1b835:      mov     qword ptr [rsp + 0x18], rcx
0x1b83a:      call    0x1d670
0x1b83f:      mov     rcx, qword ptr [rsp + 0x18]
0x1b844:      mov     dword ptr [rsp + 0x6c], r13d
0x1b849:      mov     qword ptr [rsp + 0x10], rax
0x1b84e:      movzx   r12d, byte ptr [rcx]
0x1b852:      test    r12b, r12b
0x1b855:      je      0x1be30
0x1b85b:      movsx   esi, r12b
0x1b85f:      lea     rdi, [rip + 0x6c11]
0x1b866:      call    0x4a00
0x1b86b:      test    rax, rax
0x1b86e:      je      0x1bfc0
0x1b874:      lea     eax, [r13 + 1]
0x1b878:      lea     rsi, [rsp + 0x70]
0x1b87d:      xor     edx, edx
0x1b87f:      mov     byte ptr [rsp + 0x77], 0
0x1b884:      mov     dword ptr [rsp + 0x6c], eax
0x1b888:      movzx   eax, byte ptr [rip + 0xef5e]
0x1b88f:      lea     rdi, [rsp + 0x75]
0x1b894:      mov     dword ptr [rsp + 0x70], 0
0x1b89c:      mov     byte ptr [rsp + 0x76], al
0x1b8a0:      mov     byte ptr [rsp + 0x75], al
0x1b8a4:      xor     eax, eax
0x1b8a6:      call    0x1ac70
0x1b8ab:      mov     r12d, dword ptr [rsp + 0x6c]
0x1b8b0:      mov     r14, rax
0x1b8b3:      test    r14, r14
0x1b8b6:      je      0x1c435
0x1b8bc:      movsxd  rax, r12d
0x1b8bf:      xor     esi, esi
0x1b8c1:      mov     edx, r12d
0x1b8c4:      movzx   eax, byte ptr [rbp + rax]
0x1b8c9:      cmp     al, 0x3a
0x1b8cb:      jne     0x1b8df
0x1b8cd:      lea     edx, [r12 + 1]
0x1b8d2:      mov     esi, 1
0x1b8d7:      movsxd  rax, edx
0x1b8da:      movzx   eax, byte ptr [rbp + rax]
0x1b8df:      cmp     al, 0x25
0x1b8e1:      je      0x1c010
0x1b8e7:      cmp     al, 0x2d

```

```

0x1b8e9:      je      0x1c100
0x1b8ef:      jg      0x1bd48
0x1b8f5:      cmp     al, 0x24
0x1b8f7:      je      0x1bf80
0x1b8fd:      cmp     al, 0x2a
0x1b8ff:      jne     0x1bd54
0x1b905:      add     edx, 1
0x1b908:      mov     esi, 0x24
0x1b90d:      mov     edi, 1
0x1b912:      xor     eax, eax
0x1b914:      mov     dword ptr [rsp + 0x6c], edx
0x1b918:      mov     rdx, r14
0x1b91b:      call    0x1b420
0x1b920:      mov     r13, rax
0x1b923:      test    rax, rax
0x1b926:      je      0x1c080
0x1b92c:      lea     rax, [rip + 0x101ed]
0x1b933:      cmp     r13, rax
0x1b936:      je      0x1bdae
0x1b93c:      mov     rdi, r13
0x1b93f:      call    0x49b0
0x1b944:      lea     rdi, [rax + 1]
0x1b948:      call    0x1d670
0x1b94d:      mov     rsi, r13
0x1b950:      mov     rdi, rax
0x1b953:      call    0x48b0
0x1b958:      mov     rdi, r13
0x1b95b:      mov     r14, rax
0x1b95e:      call    0x18d10
0x1b963:      movsxd  rcx, dword ptr [rsp + 0x6c]
0x1b968:      cmp     byte ptr [rbp + rcx], 0x3a
0x1b96d:      mov     dword ptr [rsp + 0x44], ecx
0x1b971:      jne     0x1c3d4
0x1b977:      mov     dword ptr [rsp + 0x18], 0
0x1b97f:      lea     r12, [rip + 0x6afe]
0x1b986:      mov     dword ptr [rsp + 0x38], 0
0x1b98e:      mov     dword ptr [rsp + 0x40], ebx
0x1b992:      mov     qword ptr [rsp + 0x28], r15
0x1b997:      mov     r15, r14
0x1b99a:      mov     r14d, ecx
0x1b99d:      nop     dword ptr [rax]
0x1b9a0:      movzx   eax, byte ptr [rbp + rcx + 1]
0x1b9a5:      cmp     al, 0x67
0x1b9a7:      jne     0x1b9bf
0x1b9a9:      add     r14d, 1
0x1b9ad:      mov     dword ptr [rsp + 0x18], 1
0x1b9b5:      mov     dword ptr [rsp + 0x6c], r14d
0x1b9ba:      movzx   eax, byte ptr [rbp + rcx + 2]

```

```

0x1b9bf:      cmp     al, 0x26
0x1b9c1:      je      0x1be70
0x1b9c7:      sub     eax, 0x65
0x1b9ca:      cmp     al, 0xf
0x1b9cc:      ja      0x1b9e0
0x1b9ce:      movzx   eax, al
0x1b9d1:      movsxd  rax, dword ptr [r12 + rax*4]
0x1b9d5:      add     rax, r12
0x1b9d8:      jmp     rax
rax
->->->                                indirect JMP
0x77e0: mov     dword ptr [rsp + 0xc], edi
0x77e4: and     r12d, 1
0x77e8: je      0x75ad
0x77ee: call    0x4f00
0x77f3: mov     r8, rax
0x77f6: movsxd  rax, dword ptr [rsp + 0xc]
0x77fb: mov     rdx, qword ptr [r8]
0x77fe: mov     rdi, rax
0x7801: test    byte ptr [rdx + rax*2 + 1], 0x40
0x7806: je      0x7818
0x7808: call    0xcd30
0x780d: jmp     0x75ad
0x75ad
0x7812: nop     word ptr [rax + rax]
0x7818: xor     r12d, r12d
0x781b: jmp     0x75ad
0x75ad
0x7820: xor     eax, eax
0x7822: call    0xcde0
0x7827: jmp     0x75ad
0x75ad
0x782c: nop     dword ptr [rax]
0x7830: endbr64
0x7834: xor     eax, eax
0x7836: cmp     dword ptr [rip + 0x2315f], 6
0x783d: jle     0x7869
0x783f: mov     edx, dword ptr [rip + 0x22413]
0x7845: mov     eax, dword ptr [rip + 0x2314d]
0x784b: test    edx, edx
0x784d: je      0x7860
0x784f: cmp     eax, 0xb
0x7852: setg    al
0x7855: movzx   eax, al
0x7858: ret
0x7859: nop     dword ptr [rax]
0x7860: cmp     eax, 5
0x7863: setg    al

```

```

0x7866: movzx    eax, al
0x7869: ret
0x786a: nop        word ptr [rax + rax]
0x7870: endbr64
0x7874: xor        eax, eax
0x7876: cmp        dword ptr [rip + 0x23067], 1
0x787d: sete      al
0x7880: ret
0x7881: nop        word ptr cs:[rax + rax]
0x788c: nop        dword ptr [rax]
0x7890: endbr64
0x7894: push       r13
0x7896: xor        eax, eax
0x7898: push       r12
0x789a: push       rbp
0x789b: push       rbx
0x789c: mov        ebx, edi
0x789e: sub        rsp, 8
0x78a2: mov        ebp, dword ptr [rip + 0x230f4]
0x78a8: mov        r12d, dword ptr [rip + 0x230e9]
0x78af: call       0xb520
0x78b4: mov        ecx, dword ptr [rip + 0x230e2]
0x78ba: test       ebx, ebx
0x78bc: jne        0x78c6
0x78be: cmp        ecx, ebp
0x78c0: je         0x7b00
0x78c6: mov        edx, dword ptr [rip + 0x230bc]
0x78cc: test       edx, edx
0x78ce: jne        0x7a60
0x78d4: mov        rdi, qword ptr [rip + 0x22fd5]
0x78db: cmp        ecx, 1
0x78de: jle        0x79e0
0x78e4: cmp        ecx, 2
0x78e7: je         0x7b70
0x78ed: cmp        ecx, 6
0x78f0: jle        0x790d
0x78f2: mov        eax, dword ptr [rip + 0x22360]
0x78f8: movsxd     r8, dword ptr [rip + 0x23099]
0x78ff: test       eax, eax
0x7901: je         0x7a90
0x7907: cmp        r8d, 0xb
0x790b: jg         0x7950
0x790d: mov        r8d, 0x50
0x7913: mov        ecx, 2
0x7918: mov        edx, 0x10000
0x791d: mov        esi, 0x10000
0x7922: call       0xf590
0x7927: mov        r8d, 0x50

```

```

0x792d: mov     ecx, 2
0x7932: mov     rdi, qword ptr [rip + 0x22f6f]
0x7939: mov     edx, 0x10000
0x793e: mov     esi, 0x10000
0x7943: call    0xf590
0x7948: jmp     0x7ad2
0x7ad2:
0x794d: nop     dword ptr [rax]
0x7950: mov     ebp, r8d
0x7953: and     r8d, 1
0x7957: sar     ebp, 1
0x7959: lea     r13d, [r8 + rbp]
0x795d: movsxd  rbp, ebp
0x7960: call    0xf470
0x7965: movsxd  r13, r13d
0x7968: mov     rdi, rax
0x796b: call    0xc6a0
0x7970: mov     rdi, qword ptr [rip + 0x22f31]
0x7977: mov     r12d, eax
0x797a: call    0xf470
0x797f: mov     rdi, rax
0x7982: call    0xc6a0
0x7987: cmp     r12d, eax
0x798a: jg      0x7b20
0x7990: mov     eax, dword ptr [rip + 0x23006]
0x7996: mov     r8, r13
0x7999: xor     esi, esi
0x799b: mov     edx, 1
0x79a0: mov     rdi, qword ptr [rip + 0x22f09]
0x79a7: lea     ecx, [rax - 3]
0x79aa: movsxd  rcx, ecx
0x79ad: call    0xf590
0x79b2: mov     r8, rbp
0x79b5: mov     edx, 1
0x79ba: mov     rsi, r13
0x79bd: mov     eax, dword ptr [rip + 0x22fd9]
0x79c3: mov     rdi, qword ptr [rip + 0x22ede]
0x79ca: lea     ecx, [rax - 3]
0x79cd: movsxd  rcx, ecx
0x79d0: call    0xf590
0x79d5: jmp     0x7ad2
0x7ad2:
0x79da: nop     word ptr [rax + rax]
0x79e0: mov     r8d, 0x50
0x79e6: mov     ecx, 2
0x79eb: mov     edx, 0x10000
0x79f0: mov     esi, 0x10000
0x79f5: call    0xf590

```



```

0x79fa: mov     r8d, 0x50
0x7a00: mov     ecx, 2
0x7a05: mov     rdi, qword ptr [rip + 0x22e9c]
0x7a0c: mov     edx, 0x10000
0x7a11: mov     esi, 0x10000
0x7a16: call    0xf590
0x7a1b: xor     esi, esi
0x7a1d: xor     edi, edi
0x7a1f: call    0x16d30
0x7a24: mov     eax, dword ptr [rip + 0x22f72]
0x7a2a: xor     edi, edi
0x7a2c: lea     esi, [rax - 1]
0x7a2f: movsxd  rsi, esi
0x7a32: call    0xde50
0x7a37: mov     eax, dword ptr [rip + 0x22f5f]
0x7a3d: mov     edi, dword ptr [rip + 0x22f55]
0x7a43: cmp     eax, 1
0x7a46: lea     esi, [rax - 2]
0x7a49: mov     eax, 0
0x7a4e: cmov    esi, eax
0x7a51: add     rsp, 8
0x7a55: pop     rbx
0x7a56: pop     rbp
0x7a57: pop     r12
0x7a59: pop     r13
0x7a5b: jmp     0xc890
0xc890
0x7a60: imul    ecx, dword ptr [rip + 0x22f31]
0x7a67: mov     rdi, qword ptr [rip + 0x22e5a]
0x7a6e: lea     esi, [rcx + rcx + 4]
0x7a72: movsxd  rsi, esi
0x7a75: call    0x1d690
0x7a7a: mov     ecx, dword ptr [rip + 0x22f1c]
0x7a80: mov     qword ptr [rip + 0x22e41], rax
0x7a87: jmp     0x78d4
0x78d4
0x7a8c: nop     dword ptr [rax]
0x7a90: cmp     r8d, 5
0x7a94: jle     0x790d
0x7a9a: sub     ecx, 3
0x7a9d: mov     edx, 1
0x7aa2: xor     esi, esi
0x7aa4: movsxd  rcx, ecx
0x7aa7: call    0xf590
0x7aac: mov     eax, dword ptr [rip + 0x22eea]
0x7ab2: mov     edx, 1
0x7ab7: xor     esi, esi
0x7ab9: movsxd  r8, dword ptr [rip + 0x22ed8]

```

```

0x7ac0: mov     rdi, qword ptr [rip + 0x22de1]
0x7ac7: lea     ecx, [rax - 3]
0x7aca: movsxd  rcx, ecx
0x7acd: call    0xf590
0x7ad2: movsxd  rdi, dword ptr [rip + 0x22ebf]
0x7ad9: xor     esi, esi
0x7adb: call    0x16d30
0x7ae0: mov     eax, dword ptr [rip + 0x22eb6]
0x7ae6: lea     esi, [rax - 1]
0x7ae9: movsxd  rsi, esi
0x7aec: movsxd  rdi, dword ptr [rip + 0x22ea5]
0x7af3: jmp     0x7a32
0x7a32:
0x7af8: nop     dword ptr [rax + rax]
0x7b00: cmp     dword ptr [rip + 0x22e91], r12d
0x7b07: jne     0x78c6
0x7b0d: add     rsp, 8
0x7b11: pop     rbx
0x7b12: pop     rbp
0x7b13: pop     r12
0x7b15: pop     r13
0x7b17: ret
0x7b18: nop     dword ptr [rax + rax]
0x7b20: mov     eax, dword ptr [rip + 0x22e76]
0x7b26: mov     rdi, qword ptr [rip + 0x22d83]
0x7b2d: mov     r8, rbp
0x7b30: mov     rsi, r13
0x7b33: mov     edx, 1
0x7b38: lea     ecx, [rax - 3]
0x7b3b: movsxd  rcx, ecx
0x7b3e: call    0xf590
0x7b43: mov     eax, dword ptr [rip + 0x22e53]
0x7b49: mov     r8, r13
0x7b4c: xor     esi, esi
0x7b4e: mov     rdi, qword ptr [rip + 0x22d53]
0x7b55: mov     edx, 1
0x7b5a: lea     ecx, [rax - 3]
0x7b5d: movsxd  rcx, ecx
0x7b60: call    0xf590
0x7b65: jmp     0x7ad2
0x7ad2:
0x7b6a: nop     word ptr [rax + rax]
0x7b70: mov     r8d, 0x50
0x7b76: mov     ecx, 2
0x7b7b: mov     edx, 0x10000
0x7b80: mov     esi, 0x10000
0x7b85: call    0xf590
0x7b8a: mov     r8d, 0x50

```

```

0x7b90: mov     ecx, 2
0x7b95: mov     rdi, qword ptr [rip + 0x22d0c]
0x7b9c: mov     edx, 0x10000
0x7ba1: mov     esi, 0x10000
0x7ba6: call    0xf590
0x7bab: xor     esi, esi
0x7bad: xor     edi, edi
0x7baf: call    0x16d30
0x7bb4: mov     eax, dword ptr [rip + 0x22de2]
0x7bba: lea     esi, [rax - 1]
0x7bbd: movsxd  rsi, esi
0x7bc0: jmp     0x7aec
0x7aec:
0x7bc5: nop     word ptr cs:[rax + rax]
0x7bd0: endbr64
0x7bd4: push    rbx
0x7bd5: xor     eax, eax
0x7bd7: mov     ebx, edi
0x7bd9: xor     edi, edi
0x7bdb: call    0x7890
0x7be0: cmp     ebx, 0x12
0x7be3: je      0x7cc8
0x7be9: mov     esi, dword ptr [rip + 0x22ced]
0x7bef: test    esi, esi
0x7bf1: je      0x7bf8
0x7bf3: pop     rbx
0x7bf4: ret
0x7bf5: nop     dword ptr [rax]
0x7bf8: mov     rdi, qword ptr [rip + 0x22cb1]
0x7bff: call    0xf480
0x7c04: mov     rdi, qword ptr [rip + 0x22c9d]
0x7c0b: call    0xf480
0x7c10: mov     rdi, qword ptr [rip + 0x22c99]
0x7c17: call    0xf530
0x7c1c: mov     rdi, qword ptr [rip + 0x22c85]
0x7c23: call    0xf530
0x7c28: mov     ecx, dword ptr [rip + 0x22cb6]
0x7c2e: test    ecx, ecx
0x7c30: jne     0x7ce0
0x7c36: cmp     dword ptr [rip + 0x22d5f], 6
0x7c3d: jle     0x7c60
0x7c3f: mov     edx, dword ptr [rip + 0x22013]
0x7c45: mov     eax, dword ptr [rip + 0x22d4d]
0x7c4b: test    edx, edx
0x7c4d: je      0x7d00
0x7c53: cmp     eax, 0xb
0x7c56: jg      0x7c6e
0x7c58: nop     dword ptr [rax + rax]

```

```

0x7c60: xor     eax, eax
0x7c62: call    0xa6c0
0x7c67: xor     eax, eax
0x7c69: call    0xa300
0x7c6e: xor     eax, eax
0x7c70: call    0x16d50
0x7c75: mov     rdi, qword ptr [rip + 0x22c34]
0x7c7c: call    0x11010
0x7c81: mov     eax, dword ptr [rip + 0x21fd1]
0x7c87: test    eax, eax
0x7c89: jne     0x7cf0
0x7c8b: xor     eax, eax
0x7c8d: call    0xdeaa0
0x7c92: xor     eax, eax
0x7c94: call    0xd870
0x7c99: xor     eax, eax
0x7c9b: call    0xd740
0x7ca0: xor     eax, eax
0x7ca2: call    0xa820
0x7ca7: cmp     ebx, 0x12
0x7caa: jne     0x7bf3
0x7cb0: mov     rdi, qword ptr [rip + 0x22bf9]
0x7cb7: call    0x11100
0x7cbc: pop     rbx
0x7cbd: mov     rdi, rax
0x7cc0: jmp     0xc4b0
0xc4b0
0x7cc5: nop     dword ptr [rax]
0x7cc8: mov     edi, 1
0x7ccd: call    0x9b00
0x7cd2: xor     eax, eax
0x7cd4: call    0xa6c0
0x7cd9: jmp     0x7be9
0x7be9
0x7cde: nop
0x7ce0: mov     rdi, qword ptr [rip + 0x22be1]
0x7ce7: call    0xbb40
0x7cec: jmp     0x7c8b
0x7c8b
0x7cee: nop
0x7cf0: mov     rdi, qword ptr [rip + 0x22bb1]
0x7cf7: call    0x11010
0x7cfc: jmp     0x7c8b
0x7c8b
0x7cfe: nop
0x7d00: cmp     eax, 5
0x7d03: jle     0x7c60
0x7d09: jmp     0x7c6e

```

```

0x7c6e
0x7d0e: nop
0x7d10: endbr64
0x7d14: push    r12
0x7d16: xor      eax, eax
0x7d18: mov      r12, rdi
0x7d1b: push     rbp
0x7d1c: push     rbx
0x7d1d: call     0xb510
0x7d22: mov      rdi, rax
0x7d25: mov      rbp, rax
0x7d28: call     0x49b0
0x7d2d: test     rax, rax
0x7d30: je       0x7d39
0x7d32: cmp      byte ptr [rbp + rax - 1], 7
0x7d37: jne      0x7d70
0x7d39: xor      eax, eax
0x7d3b: call     0xa5a0
0x7d40: test     r12, r12
0x7d43: je       0x7ded
0x7d49: xor      edx, edx
0x7d4b: xor      esi, esi
0x7d4d: mov      rdi, r12
0x7d50: call     0xe480
0x7d55: xor      eax, eax
0x7d57: call     0xd740
0x7d5c: pop      rbx
0x7d5d: xor      eax, eax
0x7d5f: pop      rbp
0x7d60: pop      r12
0x7d62: jmp      0xa820
0xa820
0x7d67: nop      word ptr [rax + rax]
0x7d70: mov      rdi, rbp
0x7d73: call     0x9fe0
0x7d78: mov      rdi, rax
0x7d7b: mov      rbx, rax
0x7d7e: call     0x49b0
0x7d83: lea      rdi, [rax + 0x80]
0x7d8a: call     0x1d670
0x7d8f: mov      r8, rbx
0x7d92: mov      esi, 1
0x7d97: lea      rcx, [rip + 0x18762]
0x7d9e: mov      rbp, rax
0x7da1: mov      rdi, rax
0x7da4: mov      rdx, -1
0x7dab: xor      eax, eax
0x7dad: call     0x4f10

```

```

0x7db2: mov     edx, 1
0x7db7: mov     esi, 2
0x7dbc: mov     rdi, rbp
0x7dbf: call    0xe480
0x7dc4: mov     rdi, rbp
0x7dc7: call    0x18d10
0x7dcc: xor     eax, eax
0x7dce: call    0xa5a0
0x7dd3: xor     eax, eax
0x7dd5: call    0xa820
0x7dda: mov     edi, 1
0x7ddf: call    0x4ea0
0x7de4: test    r12, r12
0x7de7: jne     0x7d49
0x7ded: xor     eax, eax
0x7def: call    0xe4e0
0x7df4: jmp     0x7d55
0x7d55:
0x7df9: nop     dword ptr [rax]
0x7e00: endbr64
0x7e04: push    r13
0x7e06: mov     edx, 1
0x7e0b: push    r12
0x7e0d: mov     r12, rsi
0x7e10: push    rbp
0x7e11: mov     rbp, rdi
0x7e14: sub     rsp, 0x10
0x7e18: mov     rax, qword ptr fs:[0x28]
0x7e21: mov     qword ptr [rsp + 8], rax
0x7e26: xor     eax, eax
0x7e28: mov     r13, rsp
0x7e2b: mov     rsi, r13
0x7e2e: call    0x19200
0x7e33: test    rax, rax
0x7e36: je      0x7e48
0x7e38: mov     rdi, qword ptr [rsp]
0x7e3c: mov     rsi, r12
0x7e3f: call    0x4b30
0x7e44: test    eax, eax
0x7e46: je      0x7e5f
0x7e48: mov     rdi, r12
0x7e4b: call    0x1d7a0
0x7e50: mov     rsi, r13
0x7e53: mov     rdi, rbp
0x7e56: mov     qword ptr [rsp], rax
0x7e5a: call    0x19150
0x7e5f: mov     rax, qword ptr [rsp + 8]
0x7e64: sub     rax, qword ptr fs:[0x28]

```

```

0x7e6d: jne      0x7e79
0x7e6f: add      rsp, 0x10
0x7e73: pop      rbp
0x7e74: pop      r12
0x7e76: pop      r13
0x7e78: ret
0x7e79: call     0x49d0
0x7e7e: nop
0x7e80: endbr64
0x7e84: sub      rsp, 0x18
0x7e88: mov      edx, esi
0x7e8a: mov      rax, qword ptr fs:[0x28]
0x7e93: mov      qword ptr [rsp + 8], rax
0x7e98: xor      eax, eax
0x7e9a: mov      rsi, rsp
0x7e9d: call     0x19200
0x7ea2: test     rax, rax
0x7ea5: cmovne   rax, qword ptr [rsp]
0x7eaa: mov      rdx, qword ptr [rsp + 8]
0x7eaf: sub      rdx, qword ptr fs:[0x28]
0x7eb8: jne      0x7ebf
0x7eba: add      rsp, 0x18
0x7ebe: ret
0x7ebf: call     0x49d0
0x7ec4: nop      word ptr cs:[rax + rax]
0x7ecf: nop
0x7ed0: endbr64
0x7ed4: test     sil, 2
0x7ed8: je       0x7ee0
0x7eda: jmp      0x7570
0x7570:
0x7edf: nop
0x7ee0: xor      eax, eax
0x7ee2: ret
0x7ee3: nop      word ptr cs:[rax + rax]
0x7eee: nop
0x7ef0: endbr64
0x7ef4: push     r15
0x7ef6: push     r14
0x7ef8: push     r13
0x7efa: push     r12
0x7efc: push     rbp
0x7efd: push     rbx
0x7efe: sub      rsp, 8
0x7f02: test     rdi, rdi
0x7f05: je       0x7fc8
0x7f0b: mov      rbp, rdi
0x7f0e: call     0x49b0

```

```

0x7f13: lea      r13, [rax + 1]
0x7f17: mov      rdi, r13
0x7f1a: call     0x1d670
0x7f1f: movzx    ebx, byte ptr [rbp]
0x7f23: mov      r15, rax
0x7f26: test     bl, bl
0x7f28: je       0x7fd0
0x7f2e: add      rbp, 1
0x7f32: xor      r12d, r12d
0x7f35: jmp      0x7f6d
0x7f6d:
0x7f37: nop      word ptr [rax + rax]
0x7f40: call     0x4f00
0x7f45: movsx    rdx, bl
0x7f49: add      r12d, 1
0x7f4d: mov      rax, qword ptr [rax]
0x7f50: test     byte ptr [rax + rdx*2 + 1], 0x40
0x7f55: mov      eax, 0x3f
0x7f5a: cmov     ebx, eax
0x7f5d: add      rbp, 1
0x7f61: mov      byte ptr [r15 + r14], bl
0x7f65: movzx    ebx, byte ptr [rbp - 1]
0x7f69: test     bl, bl
0x7f6b: je       0x7fa5
0x7f6d: movsxd   r14, r12d
0x7f70: cmp      bl, 9
0x7f73: jne      0x7f40
0x7f75: add      r13, 8
0x7f79: mov      rdi, r15
0x7f7c: add      rbp, 1
0x7f80: add      r12d, 8
0x7f84: mov      rsi, r13
0x7f87: call     0x1d690
0x7f8c: mov      r15, rax
0x7f8f: movabs   rax, 0x2020202020202020
0x7f99: mov      qword ptr [r15 + r14], rax
0x7f9d: movzx    ebx, byte ptr [rbp - 1]
0x7fa1: test     bl, bl
0x7fa3: jne      0x7f6d
0x7fa5: movsxd   r12, r12d
0x7fa8: add      r12, r15
0x7fab: mov      byte ptr [r12], 0
0x7fb0: add      rsp, 8
0x7fb4: mov      rax, r15
0x7fb7: pop      rbx
0x7fb8: pop      rbp
0x7fb9: pop      r12
0x7fbb: pop      r13

```



```

0x7fbd: pop      r14
0x7fbf: pop      r15
0x7fc1: ret
0x7fc2: nop      word ptr [rax + rax]
0x7fc8: xor      r15d, r15d
0x7fcb: jmp      0x7fb0
0x7fb0:
0x7fcd: nop      dword ptr [rax]
0x7fd0: mov      r12, rax
0x7fd3: jmp      0x7fab
0x7fab:
0x7fd5: nop      word ptr cs:[rax + rax]
0x7fe0: endbr64
0x7fe4: push     rbx
0x7fe5: mov      rbx, rdi
0x7fe8: call     0x49b0
0x7fed: lea      rdi, [rax + rax*2 + 9]
0x7ff2: call     0x1d670
0x7ff7: mov      byte ptr [rax], 0x28
0x7ffa: cmp      byte ptr [rbx + 1], 0
0x7ffe: je       0x8050
0x8000: mov      edx, 1
0x8005: nop      dword ptr [rax]
0x8008: movzx    ecx, byte ptr [rbx]
0x800b: mov      byte ptr [rax + rdx + 1], 0x2c
0x8010: add      rbx, 1
0x8014: mov      byte ptr [rax + rdx], cl
0x8017: mov      rcx, rdx
0x801a: add      rdx, 3
0x801e: mov      byte ptr [rax + rdx - 1], 0x20
0x8023: cmp      byte ptr [rbx + 1], 0
0x8027: jne      0x8008
0x8029: lea      r8, [rcx + 4]
0x802d: lea      rdi, [rcx + 5]
0x8031: add      rcx, 6
0x8035: movzx    esi, byte ptr [rbx]
0x8038: mov      byte ptr [rax + rdx], sil
0x803c: mov      byte ptr [rax + r8], 0x29
0x8041: mov      byte ptr [rax + rdi], 0x20
0x8045: mov      byte ptr [rax + rcx], 0
0x8049: pop      rbx
0x804a: ret
0x804b: nop      dword ptr [rax + rax]
0x8050: mov      ecx, 4
0x8055: mov      edi, 3
0x805a: mov      r8d, 2
0x8060: mov      edx, 1
0x8065: jmp      0x8035

```

```

0x8035
0x8067: nop      word ptr [rax + rax]
0x8070: endbr64
0x8074: push     r15
0x8076: push     r14
0x8078: push     r13
0x807a: push     r12
0x807c: mov      r12, rdi
0x807f: push     rbp
0x8080: mov      rbp, rsi
0x8083: push     rbx
0x8084: mov      ebx, edx
0x8086: sub      rsp, 0x28
0x808a: mov      rax, qword ptr fs:[0x28]
0x8093: mov      qword ptr [rsp + 0x18], rax
0x8098: mov      eax, edx
0x809a: and      eax, 0x10
0x809d: mov      dword ptr [rsp + 0xc], eax
0x80a1: jne      0x82f0
0x80a7: xor      r13d, r13d
0x80aa: xor      eax, eax
0x80ac: call     0xcff0
0x80b1: test     r12, r12
0x80b4: je       0x8103
0x80b6: mov      rdi, r12
0x80b9: call     0x7ef0
0x80be: mov      r12, rax
0x80c1: test     bl, 8
0x80c4: jne      0x8330
0x80ca: mov      rdi, r12
0x80cd: call     0xd610
0x80d2: test     bl, 0x20
0x80d5: jne      0x8310
0x80db: mov      rdi, r12
0x80de: call     0x18d10
0x80e3: test     rbp, rbp
0x80e6: je       0x8103
0x80e8: mov      rdi, rbp
0x80eb: call     0x7fe0
0x80f0: mov      r12, rax
0x80f3: mov      rdi, rax
0x80f6: call     0xd610
0x80fb: mov      rdi, r12
0x80fe: call     0x18d10
0x8103: xor      eax, eax
0x8105: call     0xd870
0x810a: xor      eax, eax
0x810c: call     0xd740

```

```

0x8111: xor     eax, eax
0x8113: call    0xa820
0x8118: test    bl, 4
0x811b: jne     0x8300
0x8121: mov     r14d, ebx
0x8124: lea     r12, [rsp + 0x14]
0x8129: and     r14d, 2
0x812d: mov     rdi, r12
0x8130: call    0xb180
0x8135: test    rax, rax
0x8138: je      0x81b2
0x813a: mov     rdx, qword ptr [rax]
0x813d: mov     rax, qword ptr [rax + 0x10]
0x8141: movzx   r15d, byte ptr [rdx]
0x8145: test    rax, rax
0x8148: je      0x8250
0x814e: cmp     byte ptr [rax + 0x1b], 0
0x8152: je      0x8250
0x8158: mov     rcx, qword ptr [rax]
0x815b: lea     rax, [rip + 0x20ebe]
0x8162: sub     rcx, rax
0x8165: movabs  rax, 0xea0ea0ea0ea0ea1
0x816f: imul    rcx
0x8172: sar     rcx, 0x3f
0x8176: sar     rdx, 1
0x8179: sub     rdx, rcx
0x817c: mov     r15d, edx
0x817f: not     r15d
0x8182: cmp     edx, 7
0x8185: jne     0x81d0
0x8187: xor     edi, edi
0x8189: xor     eax, eax
0x818b: call    0x7bd0
0x8190: xor     eax, eax
0x8192: call    0xdea0
0x8197: xor     eax, eax
0x8199: call    0xd740
0x819e: xor     eax, eax
0x81a0: call    0xa820
0x81a5: mov     rdi, r12
0x81a8: call    0xb180
0x81ad: test    rax, rax
0x81b0: jne     0x813a
0x81b2: xor     eax, eax
0x81b4: call    0xa5a0
0x81b9: xor     eax, eax
0x81bb: call    0xdea0
0x81c0: xor     eax, eax

```

```

0x81c2: call    0xd740
0x81c7: jmp     0x812d
0x812d:
0x81cc: nop     dword ptr [rax]
0x81d0: cmp     r15d, 7
0x81d4: je      0x82b0
0x81da: cmp     r15d, -0x3d
0x81de: jne     0x8256
0x81e0: test    rbp, rbp
0x81e3: je      0x8360
0x81e9: xor     edi, edi
0x81eb: movsx   r15d, byte ptr [rbp]
0x81f0: call    0xdba0
0x81f5: mov     esi, dword ptr [rsp + 0xc]
0x81f9: test    esi, esi
0x81fb: je      0x821f
0x81fd: test    r13, r13
0x8200: je      0x821f
0x8202: mov     rdi, r13
0x8205: call    0xc970
0x820a: xor     eax, eax
0x820c: call    0xd870
0x8211: xor     eax, eax
0x8213: call    0xd740
0x8218: xor     eax, eax
0x821a: call    0xa820
0x821f: cmp     r15d, 7
0x8223: je      0x82e1
0x8229: mov     eax, r15d
0x822c: mov     rdx, qword ptr [rsp + 0x18]
0x8231: sub     rdx, qword ptr fs:[0x28]
0x823a: jne     0x83d9
0x8240: add     rsp, 0x28
0x8244: pop     rbx
0x8245: pop     rbp
0x8246: pop     r12
0x8248: pop     r13
0x824a: pop     r14
0x824c: pop     r15
0x824e: ret
0x824f: nop
0x8250: cmp     r15d, 7
0x8254: je      0x82b0
0x8256: mov     eax, dword ptr [rsp + 0x14]
0x825a: lea     edx, [rax - 1]
0x825d: mov     dword ptr [rsp + 0x14], edx
0x8261: test    eax, eax
0x8263: je      0x83a8

```

```

0x8269: test    r14d, r14d
0x826c: je      0x8280
0x826e: mov     esi, ebx
0x8270: mov     edi, r15d
0x8273: call    0x7570
0x8278: test    eax, eax
0x827a: jne     0x8256
0x827c: nop     dword ptr [rax]
0x8280: test    rbp, rbp
0x8283: je      0x83b8
0x8289: mov     esi, r15d
0x828c: mov     rdi, rbp
0x828f: call    0x4a00
0x8294: test    rax, rax
0x8297: jne     0x83b8
0x829d: call    0xa5a0
0x82a2: jmp     0x8190
0x8190
0x82a7: nop     word ptr [rax + rax]
0x82b0: xor     edi, edi
0x82b2: call    0xdba0
0x82b7: mov     ecx, dword ptr [rsp + 0xc]
0x82bb: test    ecx, ecx
0x82bd: je      0x82e1
0x82bf: test    r13, r13
0x82c2: je      0x82e1
0x82c4: mov     rdi, r13
0x82c7: call    0xc970
0x82cc: xor     eax, eax
0x82ce: call    0xd870
0x82d3: xor     eax, eax
0x82d5: call    0xd740
0x82da: xor     eax, eax
0x82dc: call    0xa820
0x82e1: xor     eax, eax
0x82e3: jmp     0x822c
0x822c
0x82e8: nop     dword ptr [rax + rax]
0x82f0: xor     eax, eax
0x82f2: call    0xc8c0
0x82f7: mov     r13, rax
0x82fa: jmp     0x80aa
0x80aa
0x82ff: nop
0x8300: xor     eax, eax
0x8302: call    0xa5a0
0x8307: jmp     0x8121
0x8121

```

```

0x830c: nop      dword ptr [rax]
0x8310: xor      eax, eax
0x8312: call     0xccf0
0x8317: mov      rdi, r12
0x831a: call     0x18d10
0x831f: test     rbp, rbp
0x8322: jne      0x80e8
0x8328: jmp      0x8103
0x8103
0x832d: nop      dword ptr [rax]
0x8330: lea      rdi, [rip + 0x181da]
0x8337: call     0xd610
0x833c: mov      edi, 1
0x8341: call     0xdba0
0x8346: mov      rdi, r12
0x8349: call     0xd610
0x834e: test     bl, 0x20
0x8351: je       0x80db
0x8357: jmp      0x8310
0x8310
0x8359: nop      dword ptr [rax]
0x8360: xor      edi, edi
0x8362: call     0xdba0
0x8367: mov      edx, dword ptr [rsp + 0xc]
0x836b: mov      eax, 0xffffffffc3
0x8370: test     edx, edx
0x8372: je       0x822c
0x8378: test     r13, r13
0x837b: je       0x822c
0x8381: mov      rdi, r13
0x8384: call     0xc970
0x8389: xor      eax, eax
0x838b: call     0xd870
0x8390: xor      eax, eax
0x8392: call     0xd740
0x8397: xor      eax, eax
0x8399: call     0xa820
0x839e: jmp      0x8229
0x8229
0x83a3: nop      dword ptr [rax + rax]
0x83a8: call     0xd870
0x83ad: jmp      0x8190
0x8190
0x83b2: nop      word ptr [rax + rax]
0x83b8: xor      edi, edi
0x83ba: call     0xdba0
0x83bf: mov      eax, dword ptr [rsp + 0xc]
0x83c3: test     eax, eax

```

```

0x83c5: je      0x8229
0x83cb: test    r13, r13
0x83ce: jne      0x8202
0x83d4: jmp      0x8229
0x8229:
0x83d9: call     0x49d0
0x83de: nop
0x83e0: endbr64
0x83e4: push     r15
0x83e6: push     r14
0x83e8: mov      r14, rdi
0x83eb: push     r13
0x83ed: mov      r13, rsi
0x83f0: push     r12
0x83f2: mov      r12, rdx
0x83f5: push     rbp
0x83f6: mov      rbp, rcx
0x83f9: push     rbx
0x83fa: sub      rsp, 0x38
0x83fe: mov      rax, qword ptr fs:[0x28]
0x8407: mov      qword ptr [rsp + 0x28], rax
0x840c: xor      eax, eax
0x840e: call     0xcff0
0x8413: test     r14, r14
0x8416: je       0x8420
0x8418: mov      rdi, r14
0x841b: call     0xd540
0x8420: test     r12, r12
0x8423: je       0x8450
0x8425: mov      rdi, r12
0x8428: call     0xd610
0x842d: test     rbp, rbp
0x8430: je       0x8450
0x8432: mov      rsi, r12
0x8435: mov      rdi, rbp
0x8438: xor      eax, eax
0x843a: mov      r14d, 1
0x8440: call     0x7e00
0x8445: jmp      0x8453
0x8453:
0x8447: nop      word ptr [rax + rax]
0x8450: xor      r14d, r14d
0x8453: xor      eax, eax
0x8455: lea      r12, [rsp + 0x1c]
0x845a: lea      rbx, [rip + 0x17c03]
0x8461: call     0xd870
0x8466: xor      eax, eax
0x8468: call     0xd740

```

```

0x846d: xor      eax, eax
0x846f: call     0xa820
0x8474: lea      rax, [rsp + 0x20]
0x8479: mov      qword ptr [rsp], rax
0x847d: mov      rdi, r12
0x8480: call     0xb180
0x8485: test     rax, rax
0x8488: je       0x853f
0x848e: mov      rdx, qword ptr [rax]
0x8491: mov      rax, qword ptr [rax + 0x10]
0x8495: movzx    r15d, byte ptr [rdx]
0x8499: test     rax, rax
0x849c: je       0x84d7
0x849e: cmp      byte ptr [rax + 0x1b], 0
0x84a2: je       0x84d7
0x84a4: mov      rcx, qword ptr [rax]
0x84a7: lea      rax, [rip + 0x20b72]
0x84ae: sub      rcx, rax
0x84b1: movabs   rax, 0xea0ea0ea0ea0ea1
0x84bb: imul     rcx
0x84be: sar      rcx, 0x3f
0x84c2: sar      rdx, 1
0x84c5: sub      rdx, rcx
0x84c8: mov      r15d, edx
0x84cb: not      r15d
0x84ce: cmp      edx, 0x3c
0x84d1: je       0x8560
0x84d7: lea      eax, [r15 + 0x22]
0x84db: cmp      eax, 0x29
0x84de: ja       0x8505
0x84e0: movsxd   rax, dword ptr [rbx + rax*4]
0x84e4: add      rax, rbx
0x84e7: jmp      rax
rax
->->->                                indirect JMP

0x8c28: mov      dword ptr [rip + 0x21cc6], 0
0x8c32: nop      word ptr [rax + rax]
0x8c38: mov      rdi, qword ptr [r12]
0x8c3c: call     0x18d10
0x8c41: mov      qword ptr [r12], 0
0x8c49: xor      eax, eax
0x8c4b: mov      rdx, qword ptr [rsp + 0x58]
0x8c50: sub      rdx, qword ptr fs:[0x28]
0x8c59: jne      0x929f
0x8c5f: add      rsp, 0x68
0x8c63: pop      rbx
0x8c64: pop      rbp
0x8c65: pop      r12

```



```

0x8c67: pop    r13
0x8c69: pop    r14
0x8c6b: pop    r15
0x8c6d: ret
0x8c6e: nop
0x8c70: mov    rdi, r15
0x8c73: call   0xf510
0x8c78: cmp    eax, 1
0x8c7b: jne    0x8d61
0x8c81: nop    dword ptr [rax]
0x8c88: mov    rdi, r15
0x8c8b: call   0xf490
0x8c90: mov    r14, rax
0x8c93: mov    rdi, r14
0x8c96: call   0x49b0
0x8c9b: lea    rdi, [rax + 3]
0x8c9f: call   0x1d670
0x8ca4: mov    r8, r14
0x8ca7: mov    qword ptr [rsp + 0x50], rax
0x8cac: mov    rdi, rax
0x8caf: lea    rcx, [rip + 0x185df]
0x8cb6: mov    rdx, -1
0x8cbd: mov    esi, 1
0x8cc2: xor    eax, eax
0x8cc4: call   0x4f10
0x8cc9: mov    byte ptr [rbx], 0
0x8ccc: mov    rdi, qword ptr [rsp + 0x50]
0x8cd1: lea    r14, [rbp + 1]
0x8cd5: test   rdi, rdi
0x8cd8: je     0x8bd9
0x8cde: call   0x49b0
0x8ce3: mov    rdi, qword ptr [r12]
0x8ce7: add    r13, rax
0x8cea: mov    rsi, r13
0x8ced: call   0x1d690
0x8cf2: mov    rsi, qword ptr [rsp + 0x50]
0x8cf7: mov    qword ptr [r12], rax
0x8cfb: mov    rdi, rax
0x8cfe: call   0x4df0
0x8d03: mov    rbx, qword ptr [r12]
0x8d07: mov    rdi, rbx
0x8d0a: call   0x49b0
0x8d0f: mov    rdi, qword ptr [rsp + 0x50]
0x8d14: add    rbx, rax
0x8d17: call   0x18d10
0x8d1c: mov    qword ptr [rsp + 0x50], 0
0x8d25: movzx  eax, byte ptr [r14]
0x8d29: test   al, al

```

```

0x8d2b: jne      0x8be5
0x8d31: nop      dword ptr [rax]
0x8d38: mov      eax, dword ptr [rsp + 0x24]
0x8d3c: add      eax, 1
0x8d3f: mov      byte ptr [rbx], 0
0x8d42: jmp      0x8c4b
0x8c4b
0x8d47: nop      word ptr [rax + rax]
0x8d50: mov      rdi, r15
0x8d53: call     0xf510
0x8d58: cmp      eax, 4
0x8d5b: je       0x8c88
0x8d61: mov      rdi, qword ptr [rsp + 0x50]
0x8d66: test     rdi, rdi
0x8d69: je       0x8d70
0x8d6b: call     0x18d10
0x8d70: mov      qword ptr [r12], 0
0x8d78: mov      eax, 0xffffffff
0x8d7d: jmp      0x8c4b
0x8c4b
0x8d82: nop      word ptr [rax + rax]
0x8d88: mov      rdi, r15
0x8d8b: call     0xf510
0x8d90: cmp      eax, 3
0x8d93: je       0x8c88
0x8d99: jmp      0x8d61
0x8d61
0x8d9b: nop      dword ptr [rax + rax]
0x8da0: mov      rdi, qword ptr [r15 + 0x18]
0x8da4: call     0x49b0
0x8da9: lea      rdi, [rax + 3]
0x8dad: call     0x1d670
0x8db2: mov      r8, qword ptr [r15 + 0x18]
0x8db6: mov      qword ptr [rsp + 0x50], rax
0x8dbb: mov      rdi, rax
0x8dbe: jmp      0x8caf
0x8caf
0x8dc3: nop      dword ptr [rax + rax]
0x8dc8: mov      rdi, r15
0x8dcb: call     0xf4b0
0x8dd0: mov      edi, eax
0x8dd2: mov      r14d, eax
0x8dd5: call     0x4970
0x8dda: test     rax, rax
0x8ddd: je       0x8eab
0x8de3: mov      rdi, qword ptr [rax]
0x8de6: call     0x1d7a0
0x8deb: mov      qword ptr [rsp + 0x50], rax

```

```

0x8df0: jmp      0x8cc9
0x8cc9
0x8df5: nop      dword ptr [rax]
0x8df8: mov      edi, 0x10
0x8dfd: call     0x1d670
0x8e02: mov      rdi, r15
0x8e05: mov      qword ptr [rsp + 0x50], rax
0x8e0a: call     0xf4f0
0x8e0f: mov      rdi, qword ptr [rsp + 0x50]
0x8e14: mov      esi, 1
0x8e19: lea      rcx, [rip + 0x1771c]
0x8e20: and      eax, 0xffff
0x8e25: mov      rdx, -1
0x8e2c: mov      r8d, eax
0x8e2f: xor      eax, eax
0x8e31: call     0x4f10
0x8e36: jmp      0x8cc9
0x8cc9
0x8e3b: nop      dword ptr [rax + rax]
0x8e40: xor      eax, eax
0x8e42: cmp      dl, 0x69
0x8e45: mov      edx, dword ptr [rip + 0x21aad]
0x8e4b: setne    al
0x8e4e: add      eax, 1
0x8e51: mov      dword ptr [rsp + 0x24], eax
0x8e55: test     edx, edx
0x8e57: je       0x9198
0x8e5d: mov      eax, dword ptr [r15 + 0x70]
0x8e61: test     eax, eax
0x8e63: je       0x9198
0x8e69: mov      edi, 0x15
0x8e6e: call     0x1d670
0x8e73: movdqa   xmm0, xmmword ptr [rip + 0x17d35]
0x8e7b: mov      qword ptr [rsp + 0x50], rax
0x8e80: mov      byte ptr [rax + 0x10], 0
0x8e84: movups   xmmword ptr [rax], xmm0
0x8e87: jmp      0x8cc9
0x8cc9
0x8e8c: nop      dword ptr [rax]
0x8e90: mov      rdi, r15
0x8e93: call     0xf4d0
0x8e98: mov      edi, eax
0x8e9a: mov      r14d, eax
0x8e9d: call     0x4a10
0x8ea2: test     rax, rax
0x8ea5: jne      0x8de3
0x8eab: mov      edi, 0x10
0x8eb0: call     0x1d670

```

```

0x8eb5: mov     r8d, r14d
0x8eb8: mov     esi, 1
0x8ebd: lea     rcx, [rip + 0x17678]
0x8ec4: mov     qword ptr [rsp + 0x50], rax
0x8ec9: mov     rdi, rax
0x8ecc: mov     rdx, -1
0x8ed3: xor     eax, eax
0x8ed5: call    0x4f10
0x8eda: jmp     0x8cc9
0x8cc9
0x8edf: nop
0x8ee0: mov     rdi, r15
0x8ee3: call    0xf510
0x8ee8: cmp     eax, 2
0x8eeb: je      0x8c88
0x8ef1: jmp     0x8d61
0x8d61
0x8ef6: nop     word ptr cs:[rax + rax]
0x8f00: mov     rdi, r15
0x8f03: call    0xf510
0x8f08: test    eax, eax
0x8f0a: je      0x8c88
0x8f10: jmp     0x8d61
0x8d61
0x8f15: nop     dword ptr [rax]
0x8f18: mov     rdi, qword ptr [r15 + 0x18]
0x8f1c: mov     esi, 0x2f
0x8f21: lea     r14, [rip + 0x17604]
0x8f28: call    0x4a20
0x8f2d: mov     edi, 8
0x8f32: cmp     byte ptr [rax + 1], 0
0x8f36: je      0x8c9f
0x8f3c: lea     r14, [rax + 1]
0x8f40: jmp     0x8c93
0x8c93
0x8f45: mov     esi, dword ptr [rip + 0x219ad]
0x8f4b: test    esi, esi
0x8f4d: jne     0x8c28
0x8f53: cmp     byte ptr [r14 + 2], 0x7b
0x8f58: jne     0x8c38
0x8f5e: add     r14, 3
0x8f62: mov     esi, 0x7d
0x8f67: mov     rdi, r14
0x8f6a: call    0x4a00
0x8f6f: mov     rbp, rax
0x8f72: test    rax, rax
0x8f75: je      0x8c38
0x8f7b: mov     byte ptr [rax], 0

```

```

0x8f7e: mov     edx, 2
0x8f83: lea     rsi, [rip + 0x175a8]
0x8f8a: mov     rdi, r14
0x8f8d: call    0x8070
0x8f92: mov     byte ptr [rbp], 0x7d
0x8f96: cmp     al, 0x79
0x8f98: je      0x8cc9
0x8f9e: jmp     0x8d61
0x8d61:
0x8fa3: nop     dword ptr [rax + rax]
0x8fa8: mov     ecx, dword ptr [rip + 0x2194a]
0x8fae: test    ecx, ecx
0x8fb0: jne     0x8c28
0x8fb6: cmp     byte ptr [r14 + 2], 0x7b
0x8fbb: jne     0x8c38
0x8fc1: add     r14, 3
0x8fc5: mov     esi, 0x2c
0x8fca: mov     rdi, r14
0x8fcd: call    0x4a00
0x8fd2: mov     rbp, rax
0x8fd5: test    rax, rax
0x8fd8: je      0x8c38
0x8fde: mov     byte ptr [rax], 0
0x8fe1: mov     rax, qword ptr [rsp + 0x18]
0x8fe6: lea     r15, [rsp + 0x48]
0x8feb: mov     rcx, qword ptr [rsp + 0x10]
0x8ff0: mov     rdx, qword ptr [rsp + 8]
0x8ff5: mov     rsi, r15
0x8ff8: mov     dword ptr [rip + 0x218f6], 1
0x9002: mov     r8, qword ptr [rax + 8]
0x9006: mov     qword ptr [rax + 8], r14
0x900a: mov     rdi, rax
0x900d: mov     r14, rax
0x9010: xor     eax, eax
0x9012: mov     qword ptr [rsp + 0x28], r8
0x9017: call    0x8b50
0x901c: mov     r8, qword ptr [rsp + 0x28]
0x9021: mov     dword ptr [rip + 0x218cd], 0
0x902b: mov     qword ptr [r14 + 8], r8
0x902f: test    eax, eax
0x9031: jle     0x9290
0x9037: mov     rax, qword ptr [rsp + 0x18]
0x903c: mov     rdi, qword ptr [rax]
0x903f: call    0x49b0
0x9044: mov     r14, qword ptr [rsp + 0x48]
0x9049: mov     qword ptr [rsp + 0x28], rax
0x904e: mov     rdi, r14
0x9051: call    0x49b0

```

```

0x9056: mov     rdx, qword ptr [rsp + 0x28]
0x905b: lea     rdi, [rdx + rax + 0x11]
0x9060: call    0x1d670
0x9065: mov     r9, r14
0x9068: mov     esi, 1
0x906d: lea     rcx, [rip + 0x174c1]
0x9074: mov     r10, rax
0x9077: mov     rax, qword ptr [rsp + 0x18]
0x907c: mov     rdx, -1
0x9083: mov     rdi, r10
0x9086: mov     qword ptr [rsp + 0x28], r10
0x908b: mov     r8, qword ptr [rax]
0x908e: xor     eax, eax
0x9090: call    0x4f10
0x9095: mov     rdi, r14
0x9098: lea     r14, [rbp + 1]
0x909c: call    0x18d10
0x90a1: mov     byte ptr [rbp], 0x2c
0x90a5: mov     esi, 0x7d
0x90aa: mov     rdi, r14
0x90ad: mov     qword ptr [rsp + 0x48], 0
0x90b6: call    0x4a00
0x90bb: mov     rbp, rax
0x90be: test    rax, rax
0x90c1: je      0x8c38
0x90c7: mov     byte ptr [rax], 0
0x90ca: mov     rdi, qword ptr [rsp + 0x28]
0x90cf: call    0x49b0
0x90d4: mov     r10, qword ptr [rsp + 0x28]
0x90d9: cmp     rax, 0x32
0x90dd: jbe     0x90e4
0x90df: mov     byte ptr [r10 + 0x32], 0
0x90e4: mov     rax, qword ptr [rsp + 0x18]
0x90e9: mov     rcx, qword ptr [rsp + 0x10]
0x90ee: mov     rsi, r15
0x90f1: mov     qword ptr [rsp + 0x30], r10
0x90f6: mov     dword ptr [rip + 0x217f8], 1
0x9100: mov     rdx, qword ptr [rsp + 8]
0x9105: mov     r8, qword ptr [rax + 8]
0x9109: mov     qword ptr [rax + 8], r14
0x910d: mov     r15, rax
0x9110: mov     rdi, rax
0x9113: xor     eax, eax
0x9115: mov     qword ptr [rsp + 0x28], r8
0x911a: call    0x8b50
0x911f: mov     r8, qword ptr [rsp + 0x28]
0x9124: mov     r10, qword ptr [rsp + 0x30]
0x9129: mov     dword ptr [rip + 0x217c5], 0

```

```

0x9133: test    eax, eax
0x9135: mov     qword ptr [r15 + 8], r8
0x9139: jle     0x9266
0x913f: mov     rax, qword ptr [rsp + 0x18]
0x9144: mov     r15, qword ptr [rsp + 0x48]
0x9149: mov     rdi, r10
0x914c: lea     rsi, [rsp + 0x50]
0x9151: mov     qword ptr [rsp + 0x28], r10
0x9156: mov     rcx, qword ptr [rax + 0x28]
0x915a: mov     rdx, r15
0x915d: call    0x83e0
0x9162: mov     rdi, qword ptr [rsp + 0x28]
0x9167: mov     r14, rax
0x916a: call    0x18d10
0x916f: mov     rdi, r15
0x9172: call    0x18d10
0x9177: mov     byte ptr [rbp], 0x7d
0x917b: mov     qword ptr [rsp + 0x48], 0
0x9184: test    r14, r14
0x9187: jne     0x8cc9
0x918d: jmp     0x8d61
0x8d61
0x9192: nop     word ptr [rax + rax]
0x9198: mov     qword ptr [rsp + 0x50], 0
0x91a1: mov     rdi, r15
0x91a4: xor     r14d, r14d
0x91a7: call    0x100c0
0x91ac: mov     qword ptr [rsp + 0x28], rbx
0x91b1: mov     qword ptr [rsp + 0x30], rbp
0x91b6: mov     qword ptr [rsp + 0x38], r12
0x91bb: jmp     0x9232
0x9232
0x91bd: nop     dword ptr [rax]
0x91c0: cdqe
0x91c2: lea     rcx, [rax + rax*4]
0x91c6: mov     rax, qword ptr [r15 + 0x38]
0x91ca: shl     rcx, 4
0x91ce: mov     rdi, qword ptr [rax + rcx]
0x91d2: mov     r12, rcx
0x91d5: call    0x49b0
0x91da: mov     rdi, qword ptr [rsp + 0x50]
0x91df: lea     rbp, [rax + r14]
0x91e3: lea     rsi, [rbp + 4]
0x91e7: lea     rbx, [rbp + 3]
0x91eb: call    0x1d690
0x91f0: mov     qword ptr [rsp + 0x50], rax
0x91f5: mov     byte ptr [rax + r14], 0x22
0x91fa: mov     rax, qword ptr [r15 + 0x38]

```

```

0x91fe: mov     rsi, qword ptr [rax + r12]
0x9202: mov     rax, qword ptr [rsp + 0x50]
0x9207: lea     rdi, [rax + r14 + 1]
0x920c: mov     r14, rbx
0x920f: call    0x48b0
0x9214: mov     rax, qword ptr [rsp + 0x50]
0x9219: mov     byte ptr [rax + rbp + 1], 0x22
0x921e: mov     rax, qword ptr [rsp + 0x50]
0x9223: mov     byte ptr [rax + rbp + 2], 0x20
0x9228: mov     rax, qword ptr [rsp + 0x50]
0x922d: mov     byte ptr [rax + rbp + 3], 0
0x9232: mov     rdi, r15
0x9235: call    0x100e0
0x923a: cmp     eax, -1
0x923d: jne     0x91c0
0x923f: mov     rbx, qword ptr [rsp + 0x28]
0x9244: mov     rbp, qword ptr [rsp + 0x30]
0x9249: mov     r12, qword ptr [rsp + 0x38]
0x924e: test    r14, r14
0x9251: jne     0x8cc9
0x9257: jmp     0x8d61
0x8d61
0x925c: mov     eax, 1
0x9261: jmp     0x8d3f
0x8d3f
0x9266: mov     byte ptr [rbp], 0x7d
0x926a: mov     rdi, r10
0x926d: mov     dword ptr [rsp + 8], eax
0x9271: call    0x18d10
0x9276: mov     eax, dword ptr [rsp + 8]
0x927a: test    eax, eax
0x927c: je      0x8c38
0x9282: jmp     0x8d61
0x8d61
0x9287: nop     word ptr [rax + rax]
0x9290: mov     byte ptr [rbp], 0x2c
0x9294: je      0x8c38
0x929a: jmp     0x8d61
0x8d61
0x929f: call    0x49d0
0x92a4: nop     word ptr cs:[rax + rax]
0x92af: nop
0x92b0: endbr64
0x92b4: push    r13
0x92b6: mov     r13, rdx
0x92b9: push    r12
0x92bb: mov     r12, rsi
0x92be: push    rbp

```



```

0x92bf: mov     rbp, rdi
0x92c2: call    0x47e0
0x92c7: test    rax, rax
0x92ca: je      0x92d8
0x92cc: pop     rbp
0x92cd: pop     r12
0x92cf: pop     r13
0x92d1: ret
0x92d2: nop     word ptr [rax + rax]
0x92d8: test    r12, r12
0x92db: je      0x9300
0x92dd: mov     rdi, r12
0x92e0: call    0x47e0
0x92e5: mov     rsi, rax
0x92e8: test    rax, rax
0x92eb: je      0x9300
0x92ed: mov     rdi, rbp
0x92f0: pop     rbp
0x92f1: pop     r12
0x92f3: pop     r13
0x92f5: jmp     0x18380
0x18380
0x92fa: nop     word ptr [rax + rax]
0x9300: mov     rsi, r13
0x9303: mov     rdi, rbp
0x9306: pop     rbp
0x9307: pop     r12
0x9309: pop     r13
0x930b: jmp     0x18380
0x18380
0x9310: endbr64
0x9314: push    r15
0x9316: push    r14
0x9318: push    r13
0x931a: push    r12
0x931c: push    rbp
0x931d: push    rbx
0x931e: mov     ebx, edi
0x9320: sub     rsp, 0xb8
0x9327: mov     qword ptr [rsp + 0x18], rsi
0x932c: mov     rax, qword ptr fs:[0x28]
0x9335: mov     qword ptr [rsp + 0xa8], rax
0x933d: xor     eax, eax
0x933f: mov     dword ptr [rsi], 0
0x9345: cmp     edi, 0x7ff
0x934b: jg      0x95d8
0x9351: lea     rax, [rsp + 0x20]
0x9356: lea     r12, [rsp + 0x50]

```

```

0x935b: mov     qword ptr [rsp + 8], rax
0x9360: lea     r13, [rip + 0x2089e]
0x9367: jmp     0x94df
0x94df:
0x936c: nop     dword ptr [rax]
0x9370: mov     rdi, r12
0x9373: call    0xbd80
0x9378: test    rax, rax
0x937b: je      0x9508
0x9381: cmp     byte ptr [rax], 0
0x9384: je      0x94d0
0x938a: mov     edx, 0x50
0x938f: mov     rsi, rax
0x9392: mov     rdi, r12
0x9395: call    0x4cf0
0x939a: mov     dword ptr [rsp + 4], 0
0x93a2: mov     edi, 1
0x93a7: mov     esi, 0x30
0x93ac: call    0x1d740
0x93b1: mov     rdi, qword ptr [rsp + 0x20]
0x93b6: mov     rbp, rax
0x93b9: test    rdi, rdi
0x93bc: je      0x9560
0x93c2: call    0x1d7a0
0x93c7: mov     edi, 8
0x93cc: mov     qword ptr [rbp], rax
0x93d0: call    0x19100
0x93d5: mov     rdi, qword ptr [rsp + 0x30]
0x93da: mov     qword ptr [rbp + 0x28], rax
0x93de: test    rdi, rdi
0x93e1: je      0x93ec
0x93e3: call    0x1d7a0
0x93e8: mov     qword ptr [rbp + 0x10], rax
0x93ec: mov     rdi, qword ptr [rsp + 0x28]
0x93f1: test    rdi, rdi
0x93f4: je      0x9474
0x93f6: call    0x1d7a0
0x93fb: mov     qword ptr [rbp + 8], rax
0x93ff: mov     r14, qword ptr [rsp + 0x38]
0x9404: mov     eax, 1
0x9409: test    r14, r14
0x940c: je      0x9421
0x940e: call    0x4ef0
0x9413: movsx   rdx, byte ptr [r14]
0x9417: mov     rax, qword ptr [rax]
0x941a: cmp     dword ptr [rax + rdx*4], 0x79
0x941e: sete    al
0x9421: mov     byte ptr [rbp + 0x18], al

```

```

0x9424: mov     rdx, qword ptr [rsp + 0x40]
0x9429: test    rdx, rdx
0x942c: je      0x944c
0x942e: mov     qword ptr [rsp + 0x10], rdx
0x9433: call    0x4ef0
0x9438: mov     rdx, qword ptr [rsp + 0x10]
0x943d: mov     rax, qword ptr [rax]
0x9440: movsx   rdx, byte ptr [rdx]
0x9444: cmp     dword ptr [rax + rdx*4], 0x79
0x9448: sete    byte ptr [rbp + 0x19]
0x944c: mov     rdx, qword ptr [rsp + 0x48]
0x9451: test    rdx, rdx
0x9454: je      0x9474
0x9456: mov     qword ptr [rsp + 0x10], rdx
0x945b: call    0x4ef0
0x9460: mov     rdx, qword ptr [rsp + 0x10]
0x9465: mov     rax, qword ptr [rax]
0x9468: movsx   rdx, byte ptr [rdx]
0x946c: cmp     dword ptr [rax + rdx*4], 0x79
0x9470: sete    byte ptr [rbp + 0x1a]
0x9474: call    0x4f00
0x9479: mov     r15, qword ptr [rbp]
0x947d: mov     rax, qword ptr [rax]
0x9480: movsx   rdx, byte ptr [r15]
0x9484: test    byte ptr [rax + rdx*2 + 1], 2
0x9489: jne     0x9518
0x948f: mov     rdi, r12
0x9492: call    0x1d7a0
0x9497: cmp     byte ptr [rbp + 0x1b], 0
0x949b: mov     qword ptr [rbp + 0x20], rax
0x949f: jne     0x94ac
0x94a1: cmp     qword ptr [rbp + 8], 0
0x94a6: je      0x9590
0x94ac: mov     eax, dword ptr [rsp + 4]
0x94b0: test    eax, eax
0x94b2: je      0x94c5
0x94b4: mov     rdi, r12
0x94b7: call    0x9f00
0x94bc: test    rax, rax
0x94bf: je      0x95a0
0x94c5: mov     rsi, rbp
0x94c8: mov     rdi, r12
0x94cb: call    0xaf70
0x94d0: add     ebx, 1
0x94d3: cmp     ebx, 0x800
0x94d9: je      0x9580
0x94df: mov     rsi, qword ptr [rsp + 8]
0x94e4: xor     ecx, ecx

```

```

0x94e6: mov     edx, 6
0x94eb: mov     rdi, r12
0x94ee: call    0x17320
0x94f3: movzx   eax, byte ptr [rsp + 0x50]
0x94f8: test    al, al
0x94fa: je      0x95d8
0x9500: cmp     al, 0x5e
0x9502: jne     0x9370
0x9508: mov     dword ptr [rsp + 4], 1
0x9510: jmp     0x93a2
0x93a2:
0x9515: nop     dword ptr [rax]
0x9518: lea     r14, [rip + 0x1fb01]
0x951f: jmp     0x9535
0x9535:
0x9521: nop     dword ptr [rax]
0x9528: add     r14, 0x23
0x952c: cmp     r13, r14
0x952f: je      0x948f
0x9535: mov     rsi, r14
0x9538: mov     rdi, r15
0x953b: call    0x4b30
0x9540: test    eax, eax
0x9542: jne     0x9528
0x9544: mov     rdi, r15
0x9547: call    0x18d10
0x954c: mov     qword ptr [rbp], r14
0x9550: mov     byte ptr [rbp + 0x1b], 1
0x9554: jmp     0x948f
0x948f:
0x9559: nop     dword ptr [rax]
0x9560: mov     rdi, rax
0x9563: add     ebx, 1
0x9566: call    0x18d10
0x956b: cmp     ebx, 0x800
0x9571: jne     0x94df
0x9577: nop     word ptr [rax + rax]
0x9580: mov     eax, 0x800
0x9585: jmp     0x95da
0x95da:
0x9587: nop     word ptr [rax + rax]
0x9590: cmp     qword ptr [rbp + 0x10], 0
0x9595: jne     0x94ac
0x959b: jmp     0x94d0
0x94d0:
0x95a0: mov     rcx, qword ptr [rip + 0x21341]
0x95a7: mov     rdi, qword ptr [rip + 0x212d2]
0x95ae: mov     r8, r12

```

```

0x95b1: xor     eax, eax
0x95b3: lea     rdx, [rip + 0x1720e]
0x95ba: mov     esi, 1
0x95bf: call    0x4e60
0x95c4: mov     rax, qword ptr [rsp + 0x18]
0x95c9: add     dword ptr [rax], 1
0x95cc: jmp     0x94d0
0x94d0:
0x95d1: nop     dword ptr [rax]
0x95d8: mov     eax, ebx
0x95da: mov     rdx, qword ptr [rsp + 0xa8]
0x95e2: sub     rdx, qword ptr fs:[0x28]
0x95eb: jne     0x95ff
0x95ed: add     rsp, 0xb8
0x95f4: pop     rbx
0x95f5: pop     rbp
0x95f6: pop     r12
0x95f8: pop     r13
0x95fa: pop     r14
0x95fc: pop     r15
0x95fe: ret
0x95ff: call    0x49d0
0x9604: nop     word ptr cs:[rax + rax]
0x960f: nop
0x9610: endbr64
0x9614: sub     rsp, 8
0x9618: xor     edi, edi
0x961a: call    0x9b00
0x961f: xor     eax, eax
0x9621: call    0xa6c0
0x9626: mov     rdi, qword ptr [rip + 0x2129b]
0x962d: add     rsp, 8
0x9631: jmp     0xbb40
0xbb40:
0x9636: nop     word ptr cs:[rax + rax]
0x9640: endbr64
0x9644: push    rbp
0x9645: lea     rbp, [rip + 0x205b9]
0x964c: push    rbx
0x964d: mov     rdi, rbp
0x9650: sub     rsp, 0x48
0x9654: mov     rax, qword ptr fs:[0x28]
0x965d: mov     qword ptr [rsp + 0x38], rax
0x9662: xor     eax, eax
0x9664: call    0x49b0
0x9669: mov     rdi, qword ptr [rip + 0x21240]
0x9670: mov     rbx, rax
0x9673: call    0x11100

```

```

0x9678: mov     edx, 0x33
0x967d: mov     rsi, rsp
0x9680: sub     edx, ebx
0x9682: mov     rdi, rax
0x9685: call    0x182b0
0x968a: mov     rsi, rbp
0x968d: mov     rdi, rax
0x9690: call    0x4df0
0x9695: mov     rdi, rax
0x9698: call    0xd540
0x969d: mov     rax, qword ptr [rsp + 0x38]
0x96a2: sub     rax, qword ptr fs:[0x28]
0x96ab: jne     0x96b4
0x96ad: add     rsp, 0x48
0x96b1: pop     rbx
0x96b2: pop     rbp
0x96b3: ret
0x96b4: call    0x49d0
0x96b9: nop     dword ptr [rax]
0x96c0: endbr64
0x96c4: sub     rsp, 8
0x96c8: mov     r8d, 1
0x96ce: mov     rdx, qword ptr [rip + 0x211db]
0x96d5: mov     rdi, qword ptr [rip + 0x211cc]
0x96dc: mov     rcx, -1
0x96e3: mov     esi, 0x22
0x96e8: call    0x13e30
0x96ed: mov     r8d, 1
0x96f3: mov     rdx, qword ptr [rip + 0x211ae]
0x96fa: mov     rdi, qword ptr [rip + 0x211af]
0x9701: mov     rcx, -1
0x9708: mov     esi, 0x22
0x970d: add     rsp, 8
0x9711: jmp     0x13e30
0x13e30:
0x9716: nop     word ptr cs:[rax + rax]
0x9720: endbr64
0x9724: sub     rsp, 8
0x9728: mov     rdx, qword ptr [rip + 0x211b9]
0x972f: mov     edi, 1
0x9734: xor     eax, eax
0x9736: lea     rsi, [rip + 0x170b3]
0x973d: call    0x4d20
0x9742: lea     rdi, [rip + 0x170cf]
0x9749: call    0x48d0
0x974e: lea     rdi, [rip + 0x170eb]
0x9755: call    0x48d0
0x975a: lea     rdi, [rip + 0x16dde]

```

```

0x9761: call    0x48d0
0x9766: lea     rdi, [rip + 0x170fb]
0x976d: call    0x48d0
0x9772: lea     rdi, [rip + 0x17117]
0x9779: call    0x48d0
0x977e: lea     rdi, [rip + 0x1713b]
0x9785: add     rsp, 8
0x9789: jmp     0x48d0
0x48d0
0x978e: nop
0x9790: cmp     dword ptr [rip + 0x21ba9], 1
0x9797: push    rbx
0x9798: mov     ebx, edi
0x979a: je      0x9830
0x97a0: cmp     ebx, 1
0x97a3: je      0x984a
0x97a9: mov     rdi, qword ptr [rip + 0x20598]
0x97b0: test    rdi, rdi
0x97b3: je      0x97c6
0x97b5: lea     rdx, [rip + 0x2b4]
0x97bc: mov     esi, 1
0x97c1: call    0x4820
0x97c6: movzx   edx, byte ptr [rip + 0x21b83]
0x97cd: mov     dword ptr [rip + 0x21b75], 0
0x97d7: mov     dword ptr [rip + 0x21b67], 0
0x97e1: mov     eax, edx
0x97e3: and     eax, 0x7f
0x97e6: and     edx, 0x40
0x97e9: mov     byte ptr [rip + 0x21b61], al
0x97ef: jne     0x9810
0x97f1: mov     dword ptr [rip + 0x21b45], 1
0x97fb: and     eax, 0x7f
0x97fe: shl     ebx, 7
0x9801: or      eax, ebx
0x9803: mov     byte ptr [rip + 0x21b47], al
0x9809: pop     rbx
0x980a: ret
0x980b: nop     dword ptr [rax + rax]
0x9810: mov     dword ptr [rip + 0x21b2a], 0
0x981a: mov     edi, 1
0x981f: call    0x9870
0x9824: movzx   eax, byte ptr [rip + 0x21b25]
0x982b: jmp     0x97f1
0x97f1
0x982d: nop     dword ptr [rax]
0x9830: movzx   eax, byte ptr [rip + 0x21b19]
0x9837: shr     al, 7
0x983a: movzx   eax, al

```

```

0x983d: cmp     eax, edi
0x983f: je      0x9809
0x9841: cmp     ebx, 1
0x9844: jne     0x97a9
0x984a: mov     rdi, qword ptr [rip + 0x20517]
0x9851: test    rdi, rdi
0x9854: je      0x9824
0x9856: lea     rdx, [rip + 0x213]
0x985d: mov     esi, 1
0x9862: call    0x4820
0x9867: movzx   eax, byte ptr [rip + 0x21ae2]
0x986e: jmp     0x97f1
0x97f1:
0x9870: cmp     dword ptr [rip + 0x21acd], 1
0x9877: push    rbx
0x9878: mov     ebx, edi
0x987a: je      0x98e0
0x987c: cmp     ebx, 1
0x987f: je      0x98f6
0x9881: mov     rdi, qword ptr [rip + 0x204c0]
0x9888: test    rdi, rdi
0x988b: je      0x989e
0x988d: lea     rdx, [rip + 0x1dc]
0x9894: mov     esi, 1
0x9899: call    0x4820
0x989e: movzx   eax, byte ptr [rip + 0x21aab]
0x98a5: mov     dword ptr [rip + 0x21a9d], 0
0x98af: mov     dword ptr [rip + 0x21a8f], 0
0x98b9: and     eax, 0xffffffffbf
0x98bc: mov     byte ptr [rip + 0x21a8e], al
0x98c2: test    al, al
0x98c4: js      0x9920
0x98c6: mov     dword ptr [rip + 0x21a74], 1
0x98d0: and     eax, 0xffffffffbf
0x98d3: shl     ebx, 6
0x98d6: or      eax, ebx
0x98d8: mov     byte ptr [rip + 0x21a72], al
0x98de: pop     rbx
0x98df: ret
0x98e0: movzx   eax, byte ptr [rip + 0x21a69]
0x98e7: sar     eax, 6
0x98ea: and     eax, 1
0x98ed: cmp     eax, edi
0x98ef: je      0x98de
0x98f1: cmp     ebx, 1
0x98f4: jne     0x9881
0x98f6: mov     rdi, qword ptr [rip + 0x2048b]
0x98fd: test    rdi, rdi

```



```

0x9900: je      0x9934
0x9902: lea     rdx, [rip + 0x167]
0x9909: mov     esi, 1
0x990e: call    0x4820
0x9913: movzx   eax, byte ptr [rip + 0x21a36]
0x991a: jmp     0x98c6
0x98c6
0x991c: nop      dword ptr [rax]
0x9920: mov     dword ptr [rip + 0x21a16], 0
0x992a: mov     edi, 1
0x992f: call    0x9790
0x9934: movzx   eax, byte ptr [rip + 0x21a15]
0x993b: jmp     0x98c6
0x98c6
0x993d: nop      dword ptr [rax]
0x9940: push    r12
0x9942: mov     edx, 5
0x9947: lea     rsi, [rip + 0x17293]
0x994e: mov     r12d, 1
0x9954: push    rbp
0x9955: mov     rbp, rdi
0x9958: sub     rsp, 8
0x995c: call    0x4890
0x9961: test    eax, eax
0x9963: je      0x99d3
0x9965: mov     edx, 4
0x996a: lea     rsi, [rip + 0x1725f]
0x9971: mov     rdi, rbp
0x9974: call    0x4890
0x9979: test    eax, eax
0x997b: je      0x99d3
0x997d: mov     edx, 9
0x9982: lea     rsi, [rip + 0x1724c]
0x9989: mov     rdi, rbp
0x998c: call    0x4890
0x9991: test    eax, eax
0x9993: je      0x99d3
0x9995: lea     rsi, [rip + 0x17243]
0x999c: mov     rdi, rbp
0x999f: call    0x4b30
0x99a4: test    eax, eax
0x99a6: je      0x99d3
0x99a8: lea     rsi, [rip + 0x17238]
0x99af: mov     rdi, rbp
0x99b2: call    0x4b30
0x99b7: test    eax, eax
0x99b9: je      0x99d3
0x99bb: lea     rsi, [rip + 0x1722b]

```

```

0x99c2: mov     rdi, rbp
0x99c5: xor      r12d, r12d
0x99c8: call     0x4b30
0x99cd: test     eax, eax
0x99cf: sete     r12b
0x99d3: add      rsp, 8
0x99d7: mov      eax, r12d
0x99da: pop      rbp
0x99db: pop      r12
0x99dd: ret
0x99de: nop
0x99e0: push     r13
0x99e2: push     r12
0x99e4: mov      r12, qword ptr [rip + 0x20f8d]
0x99eb: push     rbp
0x99ec: test     r12, r12
0x99ef: je       0x9a58
0x99f1: mov      r13, rdi
0x99f4: jmp      0x9a13
0x9a13:
0x99f6: nop      word ptr cs:[rax + rax]
0x9a00: js       0x9a30
0x9a02: mov      r12, qword ptr [r12 + 8]
0x9a07: mov      qword ptr [rip + 0x20f6a], r12
0x9a0e: test     r12, r12
0x9a11: je       0x9a58
0x9a13: mov      rbp, qword ptr [r12]
0x9a17: mov      rdi, r13
0x9a1a: mov      rsi, rbp
0x9a1d: call     0x4b30
0x9a22: test     eax, eax
0x9a24: jne      0x9a00
0x9a26: mov      rax, r12
0x9a29: pop      rbp
0x9a2a: pop      r12
0x9a2c: pop      r13
0x9a2e: ret
0x9a2f: nop
0x9a30: mov      rdi, r13
0x9a33: call     0x49b0
0x9a38: mov      rsi, rbp
0x9a3b: mov      rdi, r13
0x9a3e: mov      rdx, rax
0x9a41: call     0x4890
0x9a46: pop      rbp
0x9a47: neg      eax
0x9a49: sbb      r12, r12
0x9a4c: mov      rax, r12

```

```

0x9a4f: pop      r12
0x9a51: pop      r13
0x9a53: ret
0x9a54: nop      dword ptr [rax]
0x9a58: mov      r12, -1
0x9a5f: pop      rbp
0x9a60: mov      rax, r12
0x9a63: pop      r12
0x9a65: pop      r13
0x9a67: ret
0x9a68: nop      dword ptr [rax + rax]
0x9a70: endbr64
0x9a74: movsxd   rdx, dword ptr [rip + 0x2148d]
0x9a7b: push     r12
0x9a7d: push     rbp
0x9a7e: mov      ebp, edi
0x9a80: push     rbx
0x9a81: cmp      edx, 0x400
0x9a87: je       0x9ab0
0x9a89: lea      eax, [rdx + 1]
0x9a8c: lea      r12, [rip + 0x2148d]
0x9a93: mov      byte ptr [r12 + rdx], bpl
0x9a97: pop      rbx
0x9a98: mov      dword ptr [rip + 0x2146a], eax
0x9a9e: pop      rbp
0x9a9f: mov      eax, 1
0x9aa4: pop      r12
0x9aa6: ret
0x9aa7: nop      word ptr [rax + rax]
0x9ab0: xor      ebx, ebx
0x9ab2: lea      r12, [rip + 0x21467]
0x9ab9: jmp      0x9acc
0x9acc:
0x9abb: nop      dword ptr [rax + rax]
0x9ac0: mov      edx, dword ptr [rip + 0x21442]
0x9ac6: add      ebx, eax
0x9ac8: cmp      edx, ebx
0x9aca: jle     0x9ae5
0x9acc: sub      edx, ebx
0x9ace: movsxd   rsi, ebx
0x9ad1: mov      edi, 1
0x9ad6: add      rsi, r12
0x9ad9: movsxd   rdx, edx
0x9adc: call     0x18da0
0x9ae1: test     eax, eax
0x9ae3: jns      0x9ac0
0x9ae5: mov      eax, 1
0x9aea: xor      edx, edx

```

```

0x9aec: jmp      0x9a93
0x9a93:
0x9aee: nop
0x9af0: endbr64
0x9af4: mov     dword ptr [rip + 0x21862], edi
0x9afa: ret
0x9afb: nop     dword ptr [rax + rax]
0x9b00: endbr64
0x9b04: push    r12
0x9b06: push    rbp
0x9b07: push    rbx
0x9b08: mov     ebx, edi
0x9b0a: sub     rsp, 0x50
0x9b0e: mov     rax, qword ptr fs:[0x28]
0x9b17: mov     qword ptr [rsp + 0x48], rax
0x9b1c: xor     eax, eax
0x9b1e: cmp     edi, 1
0x9b21: je      0x9b60
0x9b23: lea     rdx, [rip + 0x21896]
0x9b2a: mov     esi, 1
0x9b2f: mov     edi, 1
0x9b34: call    0x4d70
0x9b39: mov     dword ptr [rip + 0x20e19], ebx
0x9b3f: mov     rax, qword ptr [rsp + 0x48]
0x9b44: sub     rax, qword ptr fs:[0x28]
0x9b4d: jne     0x9c7e
0x9b53: add     rsp, 0x50
0x9b57: pop     rbx
0x9b58: pop     rbp
0x9b59: pop     r12
0x9b5b: ret
0x9b5c: nop     dword ptr [rax]
0x9b60: mov     rax, qword ptr [rip + 0x21889]
0x9b67: mov     rdx, qword ptr [rip + 0x21852]
0x9b6e: lea     r12, [rip + 0x2180b]
0x9b75: mov     esi, 1
0x9b7a: movdqa  xmm0, xmmword ptr [rip + 0x2183e]
0x9b82: movdqa  xmm1, xmmword ptr [rip + 0x21846]
0x9b8a: mov     edi, 1
0x9b8f: mov     qword ptr [rip + 0x2181a], rax
0x9b96: mov     eax, dword ptr [rip + 0x2185c]
0x9b9c: movaps  xmmword ptr [rip + 0x217dd], xmm0
0x9ba3: movdqa  xmm2, xmmword ptr [rip + 0x21835]
0x9bab: mov     dword ptr [rip + 0x21807], eax
0x9bb1: movabs  rax, 0xfffffffffffffa3c
0x9bbb: and     rax, rdx
0x9bbe: mov     rdx, r12
0x9bc1: movaps  xmmword ptr [rip + 0x217c8], xmm1

```

```

0x9bc8: mov     qword ptr [rip + 0x217b1], rax
0x9bcf: mov     eax, dword ptr [rip + 0x217b7]
0x9bd5: movaps   xmmword ptr [rip + 0x217c4], xmm2
0x9bdc: and     eax, 0xffffffff5
0x9bdf: mov     byte ptr [rip + 0x217bb], 0
0x9be6: mov     dword ptr [rip + 0x217a1], 7
0x9bf0: or      al, 0x81
0x9bf2: mov     dword ptr [rip + 0x21794], eax
0x9bf8: mov     eax, 0x100
0x9bfd: mov     word ptr [rip + 0x21792], ax
0x9c04: mov     qword ptr [rip + 0x2178a], 0x1a0000
0x9c0f: call    0x4d70
0x9c14: mov     rdi, r12
0x9c17: mov     r12, rsp
0x9c1a: call    0x4d10
0x9c1f: mov     edx, 1
0x9c24: mov     esi, 0x540a
0x9c29: mov     edi, 1
0x9c2e: mov     dword ptr [rip + 0x20cfc], eax
0x9c34: xor     eax, eax
0x9c36: call    0x4a70
0x9c3b: mov     esi, 1
0x9c40: mov     edi, 1
0x9c45: call    0x4db0
0x9c4a: mov     ebp, dword ptr [rip + 0x200d4]
0x9c50: mov     rsi, r12
0x9c53: mov     edi, 1
0x9c58: call    0x4d60
0x9c5d: mov     rdx, r12
0x9c60: mov     esi, 1
0x9c65: mov     edi, 1
0x9c6a: mov     byte ptr [rsp + 0x11], bpl
0x9c6f: mov     byte ptr [rsp + 0x12], 0
0x9c74: call    0x4d70
0x9c79: jmp     0x9b39
0x9b39
0x9c7e: call    0x49d0
0x9c83: nop     word ptr cs:[rax + rax]
0x9c8e: nop
0x9c90: endbr64
0x9c94: mov     eax, dword ptr [rip + 0x20cbe]
0x9c9a: ret
0x9c9b: nop     dword ptr [rax + rax]
0x9ca0: endbr64
0x9ca4: push    r12
0x9ca6: push    rbx
0x9ca7: mov     ebx, edi
0x9ca9: mov     edi, 1

```

```

0x9cae: sub    rsp, 0x58
0x9cb2: mov    rax, qword ptr fs:[0x28]
0x9cbb: mov    qword ptr [rsp + 0x48], rax
0x9cc0: xor    eax, eax
0x9cc2: mov    r12, rsp
0x9cc5: mov    rsi, r12
0x9cc8: call   0x4d60
0x9ccd: mov    rdx, r12
0x9cd0: mov    esi, 1
0x9cd5: mov    edi, 1
0x9cda: mov    byte ptr [rsp + 0x11], bl
0x9cde: mov    byte ptr [rsp + 0x12], 0
0x9ce3: call   0x4d70
0x9ce8: mov    dword ptr [rip + 0x20036], ebx
0x9cee: mov    rax, qword ptr [rsp + 0x48]
0x9cf3: sub    rax, qword ptr fs:[0x28]
0x9cfc: jne    0x9d06
0x9cfe: add    rsp, 0x58
0x9d02: pop    rbx
0x9d03: pop    r12
0x9d05: ret
0x9d06: call   0x49d0
0x9d0b: nop    dword ptr [rax + rax]
0x9d10: endbr64
0x9d14: mov    edx, dword ptr [rip + 0x211ee]
0x9d1a: test   edx, edx
0x9d1c: jle    0x9d66
0x9d1e: push   rbp
0x9d1f: xor    ebp, ebp
0x9d21: push   rbx
0x9d22: lea    rbx, [rip + 0x211f7]
0x9d29: sub    rsp, 8
0x9d2d: jmp    0x9d3c
0x9d3c:
0x9d2f: nop
0x9d30: mov    edx, dword ptr [rip + 0x211d2]
0x9d36: add    ebp, eax
0x9d38: cmp    edx, ebp
0x9d3a: jle    0x9d55
0x9d3c: sub    edx, ebp
0x9d3e: movsxd rsi, ebp
0x9d41: mov    edi, 1
0x9d46: add    rsi, rbx
0x9d49: movsxd rdx, edx
0x9d4c: call   0x18da0
0x9d51: test   eax, eax
0x9d53: jns    0x9d30
0x9d55: mov    dword ptr [rip + 0x211a9], 0

```

```

0x9d5f: add     rsp, 8
0x9d63: pop     rbx
0x9d64: pop     rbp
0x9d65: ret
0x9d66: mov     dword ptr [rip + 0x21198], 0
0x9d70: ret
0x9d71: nop     word ptr cs:[rax + rax]
0x9d7c: nop     dword ptr [rax]
0x9d80: endbr64
0x9d84: push    rbp
0x9d85: lea     rdx, [rip - 0x31c]
0x9d8c: push    rbx
0x9d8d: sub     rsp, 8
0x9d91: mov     esi, dword ptr [rip + 0x20c05]
0x9d97: mov     rdi, qword ptr [rip + 0x2004a]
0x9d9e: call    0x4820
0x9da3: mov     edx, dword ptr [rip + 0x2115f]
0x9da9: test    edx, edx
0x9dab: jle     0x9de5
0x9dad: xor     ebp, ebp
0x9daf: lea     rbx, [rip + 0x2116a]
0x9db6: jmp     0x9dcc
0x9dcc:
0x9db8: nop     dword ptr [rax + rax]
0x9dc0: mov     edx, dword ptr [rip + 0x21142]
0x9dc6: add     ebp, eax
0x9dc8: cmp     edx, ebp
0x9dca: jle     0x9de5
0x9dcc: sub     edx, ebp
0x9dce: movsxd rsi, ebp
0x9dd1: mov     edi, 1
0x9dd6: add     rsi, rbx
0x9dd9: movsxd rdx, edx
0x9ddc: call    0x18da0
0x9de1: test    eax, eax
0x9de3: jns     0x9dc0
0x9de5: mov     dword ptr [rip + 0x21119], 0
0x9def: add     rsp, 8
0x9df3: pop     rbx
0x9df4: pop     rbp
0x9df5: ret
0x9df6: nop     word ptr cs:[rax + rax]
0x9e00: endbr64
0x9e04: push    rbp
0x9e05: lea     rdx, [rip - 0x39c]
0x9e0c: push    rbx
0x9e0d: sub     rsp, 8
0x9e11: mov     eax, dword ptr [rip + 0x20b85]

```

```

0x9e17: mov     rdi, qword ptr [rip + 0x2010a]
0x9e1e: lea     esi, [rax - 1]
0x9e21: call    0x4820
0x9e26: mov     edx, dword ptr [rip + 0x210dc]
0x9e2c: test    edx, edx
0x9e2e: jle     0x9e65
0x9e30: xor     ebp, ebp
0x9e32: lea     rbx, [rip + 0x210e7]
0x9e39: jmp     0x9e4c
0x9e4c:
0x9e3b: nop     dword ptr [rax + rax]
0x9e40: mov     edx, dword ptr [rip + 0x210c2]
0x9e46: add     ebp, eax
0x9e48: cmp     edx, ebp
0x9e4a: jle     0x9e65
0x9e4c: sub     edx, ebp
0x9e4e: movsxd  rsi, ebp
0x9e51: mov     edi, 1
0x9e56: add     rsi, rbx
0x9e59: movsxd  rdx, edx
0x9e5c: call    0x18da0
0x9e61: test    eax, eax
0x9e63: jns     0x9e40
0x9e65: mov     dword ptr [rip + 0x21099], 0
0x9e6f: add     rsp, 8
0x9e73: pop     rbx
0x9e74: pop     rbp
0x9e75: ret
0x9e76: nop     word ptr cs:[rax + rax]
0x9e80: endbr64
0x9e84: push    rbp
0x9e85: lea     rdx, [rip - 0x41c]
0x9e8c: push    rbx
0x9e8d: sub     rsp, 8
0x9e91: mov     eax, dword ptr [rip + 0x20b05]
0x9e97: mov     rdi, qword ptr [rip + 0x200aa]
0x9e9e: lea     esi, [rax - 1]
0x9ea1: call    0x4820
0x9ea6: mov     edx, dword ptr [rip + 0x2105c]
0x9eac: test    edx, edx
0x9eae: jle     0x9ee5
0x9eb0: xor     ebp, ebp
0x9eb2: lea     rbx, [rip + 0x21067]
0x9eb9: jmp     0x9ecc
0x9ecc:
0x9ebb: nop     dword ptr [rax + rax]
0x9ec0: mov     edx, dword ptr [rip + 0x21042]
0x9ec6: add     ebp, eax

```



```

0x9ec8: cmp     edx, ebp
0x9eca: jle     0x9ee5
0x9ecc: sub     edx, ebp
0x9ece: movsxd  rsi, ebp
0x9ed1: mov     edi, 1
0x9ed6: add     rsi, rbx
0x9ed9: movsxd  rdx, edx
0x9edc: call    0x18da0
0x9ee1: test    eax, eax
0x9ee3: jns     0x9ec0
0x9ee5: mov     dword ptr [rip + 0x21019], 0
0x9eef: add     rsp, 8
0x9ef3: pop     rbx
0x9ef4: pop     rbp
0x9ef5: ret
0x9ef6: nop     word ptr cs:[rax + rax]
0x9f00: endbr64
0x9f04: mov     edx, dword ptr [rip + 0x214f2]
0x9f0a: movzx   eax, byte ptr [rdi]
0x9f0d: push    r14
0x9f0f: push    r13
0x9f11: mov     r13, rdi
0x9f14: push    r12
0x9f16: push    rbp
0x9f17: push    rbx
0x9f18: test    edx, edx
0x9f1a: jne     0x9fc8
0x9f20: cmp     al, 0x5e
0x9f22: jne     0x9fb8
0x9f28: lea     rbx, [r13 + 1]
0x9f2c: mov     rbp, r13
0x9f2f: lea     r14, [rip + 0x1fd6a]
0x9f36: jmp     0x9f56
0x9f56:
0x9f38: nop     dword ptr [rax + rax]
0x9f40: add     rbp, 1
0x9f44: mov     byte ptr [rbx - 1], al
0x9f47: mov     rdx, rbx
0x9f4a: add     rbx, 1
0x9f4e: movzx   eax, byte ptr [rbp]
0x9f52: test    al, al
0x9f54: je      0x9fa6
0x9f56: cmp     al, 0x5e
0x9f58: jne     0x9f40
0x9f5a: movzx   r12d, byte ptr [rbp + 1]
0x9f5f: test    r12b, r12b
0x9f62: je      0x9fb8
0x9f64: call    0x47c0

```

```

0x9f69: mov     r8, rax
0x9f6c: movzx    eax, r12b
0x9f70: mov     rdx, qword ptr [r8]
0x9f73: mov     eax, dword ptr [rdx + rax*4]
0x9f76: cmp     eax, 0x47
0x9f79: je      0x9fb8
0x9f7b: cmp     eax, 0x5a
0x9f7e: je      0x9fb8
0x9f80: and     r12d, 0x7f
0x9f84: add     rbp, 2
0x9f88: mov     rdx, rbx
0x9f8b: add     rbx, 1
0x9f8f: sub     r12d, 0x20
0x9f93: movsxd  r12, r12d
0x9f96: movzx    eax, byte ptr [r14 + r12]
0x9f9b: mov     byte ptr [rbx - 2], al
0x9f9e: movzx    eax, byte ptr [rbp]
0x9fa2: test    al, al
0x9fa4: jne     0x9f56
0x9fa6: mov     byte ptr [rdx], 0
0x9fa9: mov     rax, r13
0x9fac: pop     rbx
0x9fad: pop     rbp
0x9fae: pop     r12
0x9fb0: pop     r13
0x9fb2: pop     r14
0x9fb4: ret
0x9fb5: nop     dword ptr [rax]
0x9fb8: pop     rbx
0x9fb9: xor     eax, eax
0x9fbb: pop     rbp
0x9fbc: pop     r12
0x9fbe: pop     r13
0x9fc0: pop     r14
0x9fc2: ret
0x9fc3: nop     dword ptr [rax + rax]
0x9fc8: test    al, al
0x9fca: jne     0x9f28
0x9fd0: mov     rdx, rdi
0x9fd3: jmp     0x9fa6
0x9fa6:
0x9fd5: nop     word ptr cs:[rax + rax]
0x9fe0: endbr64
0x9fe4: push    r15
0x9fe6: push    r14
0x9fe8: mov     r14, rdi
0x9feb: push    r13
0x9fed: lea     r13, [rip + 0x16c12]

```

```

0x9ff4: push    r12
0x9ff6: lea     r12, [rip + 0x209c3]
0x9ffd: push    rbp
0x9ffe: lea     rbp, [rip + 0x16bef]
0xa005: push    rbx
0xa006: mov     rbx, rdi
0xa009: sub     rsp, 0x18
0xa00d: mov     rax, qword ptr fs:[0x28]
0xa016: mov     qword ptr [rsp + 8], rax
0xa01b: xor     eax, eax
0xa01d: mov     byte ptr [rip + 0x2099c], 0
0xa024: movzx   r15d, byte ptr [rdi]
0xa028: test    r15b, r15b
0xa02b: jne     0xa0c2
0xa031: jmp     0xa0f0
0xa0f0:
0xa036: nop     word ptr cs:[rax + rax]
0xa040: cmp     r15b, 0x20
0xa044: je      0xa148
0xa04a: cmp     r15b, 0x7f
0xa04e: je      0xa160
0xa054: cmp     r15b, 0xff
0xa058: je      0xa180
0xa05e: call    0x4f00
0xa063: mov     r8, rax
0xa066: movzx   eax, r15b
0xa06a: mov     rsi, qword ptr [r8]
0xa06d: test    byte ptr [rsi + rax*2], 2
0xa071: je      0xa120
0xa077: lea     edx, [r15 + 0x40]
0xa07b: lea     rsi, [rsp + 5]
0xa080: mov     rdi, r12
0xa083: mov     byte ptr [rsp + 5], 0x5e
0xa088: mov     byte ptr [rsp + 6], dl
0xa08c: mov     edx, 0x12c
0xa091: mov     byte ptr [rsp + 7], 0
0xa096: call    0x47d0
0xa09b: add     r14, 1
0xa09f: cmp     byte ptr [r14], 0
0xa0a3: je      0xa0f0
0xa0a5: cmp     rbx, r14
0xa0a8: je      0xa0be
0xa0aa: mov     edx, 0x12c
0xa0af: lea     rsi, [rip + 0x1668f]
0xa0b6: mov     rdi, r12
0xa0b9: call    0x47d0
0xa0be: movzx   r15d, byte ptr [r14]
0xa0c2: cmp     r15b, 0x1b

```

```

0xa0c6: jne      0xa040
0xa0cc: mov      edx, 0x12c
0xa0d1: mov      rsi, rbp
0xa0d4: mov      rdi, r12
0xa0d7: add      r14, 1
0xa0db: call     0x47d0
0xa0e0: cmp      byte ptr [r14], 0
0xa0e4: jne      0xa0a5
0xa0e6: nop      word ptr cs:[rax + rax]
0xa0f0: mov      rax, qword ptr [rsp + 8]
0xa0f5: sub      rax, qword ptr fs:[0x28]
0xa0fe: jne      0xa199
0xa104: add      rsp, 0x18
0xa108: mov      rax, r12
0xa10b: pop      rbx
0xa10c: pop      rbp
0xa10d: pop      r12
0xa10f: pop      r13
0xa111: pop      r14
0xa113: pop      r15
0xa115: ret
0xa116: nop      word ptr cs:[rax + rax]
0xa120: lea      rsi, [rsp + 5]
0xa125: mov      edx, 0x12c
0xa12a: mov      rdi, r12
0xa12d: mov      byte ptr [rsp + 5], r15b
0xa132: mov      byte ptr [rsp + 6], 0
0xa137: call     0x47d0
0xa13c: jmp      0xa09b
0xa09b
0xa141: nop      dword ptr [rax]
0xa148: mov      edx, 0x12c
0xa14d: mov      rsi, r13
0xa150: mov      rdi, r12
0xa153: call     0x47d0
0xa158: jmp      0xa09b
0xa09b
0xa15d: nop      dword ptr [rax]
0xa160: mov      edx, 0x12c
0xa165: lea      rsi, [rip + 0x16a8f]
0xa16c: mov      rdi, r12
0xa16f: call     0x47d0
0xa174: jmp      0xa09b
0xa09b
0xa179: nop      dword ptr [rax]
0xa180: mov      edx, 0x12c
0xa185: lea      rsi, [rip + 0x16a79]
0xa18c: mov      rdi, r12

```

```

0xa18f: call    0x47d0
0xa194: jmp     0xa09b
0xa09b:
0xa199: call    0x49d0
0xa19e: nop
0xa1a0: endbr64
0xa1a4: mov     eax, dword ptr [rip + 0x211ae]
0xa1aa: mov     r8, rdi
0xa1ad: mov     edi, dword ptr [rip + 0x207e5]
0xa1b3: push    r13
0xa1b5: push    r12
0xa1b7: xor     r12d, r12d
0xa1ba: lea     edx, [rax + rsi]
0xa1bd: push    rbx
0xa1be: mov     dword ptr [rip + 0x21194], edx
0xa1c4: cmp     edi, eax
0xa1c6: jle     0xa21c
0xa1c8: mov     ecx, dword ptr [rip + 0x21186]
0xa1ce: cmp     ecx, dword ptr [rip + 0x207c8]
0xa1d4: jge     0xa21c
0xa1d6: mov     r12d, edi
0xa1d9: sub     r12d, eax
0xa1dc: cmp     edx, edi
0xa1de: cmovle r12d, esi
0xa1e2: imul    edi, ecx
0xa1e5: mov     rsi, r8
0xa1e8: movsxd  r13, r12d
0xa1eb: lea     ebx, [rdi + rax]
0xa1ee: mov     rdi, qword ptr [rip + 0x21143]
0xa1f5: mov     rdx, r13
0xa1f8: movsxd  rbx, ebx
0xa1fb: add     rdi, rbx
0xa1fe: call    0x4bb0
0xa203: add     rbx, qword ptr [rip + 0x21126]
0xa20a: movzx   esi, byte ptr [rip + 0x21140]
0xa211: mov     rdx, r13
0xa214: mov     rdi, rbx
0xa217: call    0x4a50
0xa21c: mov     eax, r12d
0xa21f: pop     rbx
0xa220: pop     r12
0xa222: pop     r13
0xa224: ret
0xa225: nop     word ptr cs:[rax + rax]
0xa230: endbr64
0xa234: push    r13
0xa236: push    r12
0xa238: xor     r12d, r12d

```

```

0xa23b: push    rbx
0xa23c: sub     rsp, 0x10
0xa240: mov     rax, qword ptr fs:[0x28]
0xa249: mov     qword ptr [rsp + 8], rax
0xa24e: xor     eax, eax
0xa250: mov     eax, dword ptr [rip + 0x21102]
0xa256: mov     byte ptr [rsp + 7], dil
0xa25b: mov     edi, dword ptr [rip + 0x20737]
0xa261: lea     edx, [rax + 1]
0xa264: mov     dword ptr [rip + 0x210ee], edx
0xa26a: cmp     eax, edi
0xa26c: jge     0xa2c2
0xa26e: mov     ecx, dword ptr [rip + 0x210e0]
0xa274: cmp     ecx, dword ptr [rip + 0x20722]
0xa27a: jge     0xa2c2
0xa27c: cmp     edx, edi
0xa27e: jle     0xa2e0
0xa280: mov     r12d, edi
0xa283: sub     r12d, eax
0xa286: movsxd  r13, r12d
0xa289: imul    edi, ecx
0xa28c: lea     rsi, [rsp + 7]
0xa291: mov     rdx, r13
0xa294: lea     ebx, [rdi + rax]
0xa297: mov     rdi, qword ptr [rip + 0x2109a]
0xa29e: movsxd  rbx, ebx
0xa2a1: add     rdi, rbx
0xa2a4: call    0x4bb0
0xa2a9: add     rbx, qword ptr [rip + 0x21080]
0xa2b0: movzx   esi, byte ptr [rip + 0x2109a]
0xa2b7: mov     rdx, r13
0xa2ba: mov     rdi, rbx
0xa2bd: call    0x4a50
0xa2c2: mov     rax, qword ptr [rsp + 8]
0xa2c7: sub     rax, qword ptr fs:[0x28]
0xa2d0: jne     0xa2ee
0xa2d2: add     rsp, 0x10
0xa2d6: mov     eax, r12d
0xa2d9: pop     rbx
0xa2da: pop     r12
0xa2dc: pop     r13
0xa2de: ret
0xa2df: nop
0xa2e0: mov     r13d, 1
0xa2e6: mov     r12d, 1
0xa2ec: jmp     0xa289
0xa289
0xa2ee: call    0x49d0

```

```

0xa2f3: nop      word ptr cs:[rax + rax]
0xa2fe: nop
0xa300: endbr64
0xa304: push     rbp
0xa305: lea      rdx, [rip - 0x89c]
0xa30c: push     rbx
0xa30d: sub      rsp, 8
0xa311: mov      esi, dword ptr [rip + 0x20685]
0xa317: mov      rdi, qword ptr [rip + 0x1faca]
0xa31e: call     0x4820
0xa323: mov      edx, dword ptr [rip + 0x20bdf]
0xa329: test     edx, edx
0xa32b: jle      0xa365
0xa32d: xor      ebp, ebp
0xa32f: lea      rbx, [rip + 0x20bea]
0xa336: jmp      0xa34c
0xa34c:
0xa338: nop      dword ptr [rax + rax]
0xa340: mov      edx, dword ptr [rip + 0x20bc2]
0xa346: add      ebp, eax
0xa348: cmp      edx, ebp
0xa34a: jle      0xa365
0xa34c: sub      edx, ebp
0xa34e: movsxd   rsi, ebp
0xa351: mov      edi, 1
0xa356: add      rsi, rbx
0xa359: movsxd   rdx, edx
0xa35c: call     0x18da0
0xa361: test     eax, eax
0xa363: jns      0xa340
0xa365: mov      edx, dword ptr [rip + 0x20631]
0xa36b: xor      esi, esi
0xa36d: imul     edx, dword ptr [rip + 0x20624]
0xa374: mov      dword ptr [rip + 0x20b8a], 0
0xa37e: mov      rdi, qword ptr [rip + 0x20fb3]
0xa385: movsxd   rdx, edx
0xa388: call     0x4a50
0xa38d: mov      edx, dword ptr [rip + 0x20609]
0xa393: imul     edx, dword ptr [rip + 0x205fe]
0xa39a: xor      esi, esi
0xa39c: mov      rdi, qword ptr [rip + 0x20f8d]
0xa3a3: movsxd   rdx, edx
0xa3a6: call     0x4a50
0xa3ab: mov      edx, dword ptr [rip + 0x205eb]
0xa3b1: imul     edx, dword ptr [rip + 0x205e0]
0xa3b8: xor      esi, esi
0xa3ba: mov      rdi, qword ptr [rip + 0x20f67]
0xa3c1: movsxd   rdx, edx

```

```

0xa3c4: call    0x4a50
0xa3c9: mov     edx, dword ptr [rip + 0x205cd]
0xa3cf: imul    edx, dword ptr [rip + 0x205c2]
0xa3d6: xor     esi, esi
0xa3d8: mov     rdi, qword ptr [rip + 0x20f41]
0xa3df: movsxd  rdx, edx
0xa3e2: call    0x4a50
0xa3e7: mov     dword ptr [rip + 0x20f67], 0
0xa3f1: mov     dword ptr [rip + 0x20f59], 0
0xa3fb: add     rsp, 8
0xa3ff: pop     rbx
0xa400: pop     rbp
0xa401: ret
0xa402: nop     word ptr cs:[rax + rax]
0xa40d: nop     dword ptr [rax]
0xa410: endbr64
0xa414: sub     rsp, 8
0xa418: mov     edx, dword ptr [rip + 0x2057e]
0xa41e: imul    edx, dword ptr [rip + 0x20573]
0xa425: mov     esi, 0x20
0xa42a: mov     rdi, qword ptr [rip + 0x20f07]
0xa431: movsxd  rdx, edx
0xa434: call    0x4a50
0xa439: mov     edx, dword ptr [rip + 0x2055d]
0xa43f: imul    edx, dword ptr [rip + 0x20552]
0xa446: movzx   esi, byte ptr [rip + 0x20f04]
0xa44d: mov     rdi, qword ptr [rip + 0x20edc]
0xa454: movsxd  rdx, edx
0xa457: call    0x4a50
0xa45c: mov     edx, dword ptr [rip + 0x2053a]
0xa462: imul    edx, dword ptr [rip + 0x2052f]
0xa469: xor     esi, esi
0xa46b: mov     rdi, qword ptr [rip + 0x20eb6]
0xa472: add     rsp, 8
0xa476: movsxd  rdx, edx
0xa479: jmp     0x4a50
0x4a50
0xa47e: nop
0xa480: endbr64
0xa484: mov     edx, dword ptr [rip + 0x20512]
0xa48a: imul    edx, dword ptr [rip + 0x20507]
0xa491: xor     esi, esi
0xa493: mov     rdi, qword ptr [rip + 0x20e8e]
0xa49a: movsxd  rdx, edx
0xa49d: jmp     0x4a50
0x4a50
0xa4a2: nop     word ptr cs:[rax + rax]
0xa4ad: nop     dword ptr [rax]

```



```

0xa4b0: endbr64
0xa4b4: mov     dword ptr [rip + 0x20e9a], edi
0xa4ba: mov     dword ptr [rip + 0x20e98], esi
0xa4c0: ret
0xa4c1: nop     word ptr cs:[rax + rax]
0xa4cc: nop     dword ptr [rax]
0xa4d0: endbr64
0xa4d4: mov     eax, dword ptr [rip + 0x20e7a]
0xa4da: mov     dword ptr [rdi], eax
0xa4dc: mov     eax, dword ptr [rip + 0x20e76]
0xa4e2: mov     dword ptr [rsi], eax
0xa4e4: ret
0xa4e5: nop     word ptr cs:[rax + rax]
0xa4f0: endbr64
0xa4f4: movzx   eax, byte ptr [rip + 0x20e56]
0xa4fb: and     edi, 7
0xa4fe: and     eax, 0xffffffff8
0xa501: or      eax, edi
0xa503: mov     byte ptr [rip + 0x20e48], al
0xa509: ret
0xa50a: nop     word ptr [rax + rax]
0xa510: endbr64
0xa514: movzx   eax, byte ptr [rip + 0x20e36]
0xa51b: shl     edi, 3
0xa51e: and     edi, 0x38
0xa521: and     eax, 0xffffffffc7
0xa524: or      eax, edi
0xa526: mov     byte ptr [rip + 0x20e25], al
0xa52c: ret
0xa52d: nop     dword ptr [rax]
0xa530: endbr64
0xa534: movzx   eax, byte ptr [rip + 0x20e16]
0xa53b: shl     edi, 6
0xa53e: and     edi, 0x40
0xa541: and     eax, 0xffffffffbf
0xa544: or      eax, edi
0xa546: mov     byte ptr [rip + 0x20e05], al
0xa54c: ret
0xa54d: nop     dword ptr [rax]
0xa550: endbr64
0xa554: movzx   eax, byte ptr [rip + 0x20df6]
0xa55b: shl     edi, 7
0xa55e: and     eax, 0x7f
0xa561: or      eax, edi
0xa563: mov     byte ptr [rip + 0x20de8], al
0xa569: ret
0xa56a: nop     word ptr [rax + rax]
0xa570: endbr64

```

```

0xa574: shl     edi, 6
0xa577: and     esi, 7
0xa57a: shl     edx, 3
0xa57d: mov     eax, edi
0xa57f: movzx   edi, byte ptr [rip + 0x20dcb]
0xa586: and     edx, 0x38
0xa589: and     eax, 0x40
0xa58c: and     edi, 0xffffffffb8
0xa58f: or      edi, eax
0xa591: or      edi, esi
0xa593: and     edi, 0xffffffffc7
0xa596: or      edi, edx
0xa598: mov     byte ptr [rip + 0x20db2], dil
0xa59f: ret
0xa5a0: endbr64
0xa5a4: push    rbp
0xa5a5: push    rbx
0xa5a6: sub     rsp, 8
0xa5aa: movsxd  rax, dword ptr [rip + 0x20957]
0xa5b1: cmp     eax, 0x400
0xa5b6: je      0xa610
0xa5b8: lea     edx, [rax + 1]
0xa5bb: lea     rbx, [rip + 0x2095e]
0xa5c2: mov     dword ptr [rip + 0x20940], edx
0xa5c8: mov     byte ptr [rbx + rax], 7
0xa5cc: test    edx, edx
0xa5ce: jle     0xa5fd
0xa5d0: xor     ebp, ebp
0xa5d2: jmp     0xa5e4
0xa5e4:
0xa5d4: nop     dword ptr [rax]
0xa5d8: mov     edx, dword ptr [rip + 0x2092a]
0xa5de: add     ebp, eax
0xa5e0: cmp     edx, ebp
0xa5e2: jle     0xa5fd
0xa5e4: sub     edx, ebp
0xa5e6: movsxd  rsi, ebp
0xa5e9: mov     edi, 1
0xa5ee: add     rsi, rbx
0xa5f1: movsxd  rdx, edx
0xa5f4: call    0x18da0
0xa5f9: test    eax, eax
0xa5fb: jns     0xa5d8
0xa5fd: mov     dword ptr [rip + 0x20901], 0
0xa607: add     rsp, 8
0xa60b: pop     rbx
0xa60c: pop     rbp
0xa60d: ret

```

```

0xa60e: nop
0xa610: xor     ebp, ebp
0xa612: lea     rbx, [rip + 0x20907]
0xa619: jmp     0xa62c
0xa62c:
0xa61b: nop     dword ptr [rax + rax]
0xa620: add     ebp, eax
0xa622: mov     eax, dword ptr [rip + 0x208e0]
0xa628: cmp     eax, ebp
0xa62a: jle     0xa645
0xa62c: sub     eax, ebp
0xa62e: movsxd  rsi, ebp
0xa631: mov     edi, 1
0xa636: add     rsi, rbx
0xa639: movsxd  rdx, eax
0xa63c: call    0x18da0
0xa641: test    eax, eax
0xa643: jns     0xa620
0xa645: mov     dword ptr [rip + 0x208b9], 1
0xa64f: mov     edx, 1
0xa654: mov     byte ptr [rip + 0x208c5], 7
0xa65b: jmp     0xa5d0
0xa5d0:
0xa660: endbr64
0xa664: test    edi, edi
0xa666: je      0xa688
0xa668: mov     rdi, qword ptr [rip + 0x1f759]
0xa66f: test    rdi, rdi
0xa672: je      0xa694
0xa674: lea     rdx, [rip - 0xc0b]
0xa67b: mov     esi, 1
0xa680: jmp     0x4820
0x4820:
0xa685: nop     dword ptr [rax]
0xa688: mov     rdi, qword ptr [rip + 0x1f719]
0xa68f: test    rdi, rdi
0xa692: jne     0xa674
0xa694: ret
0xa695: nop     word ptr cs:[rax + rax]
0xa6a0: endbr64
0xa6a4: movzx   eax, byte ptr [rip + 0x20ca6]
0xa6ab: mov     byte ptr [rdi], al
0xa6ad: ret
0xa6ae: nop
0xa6b0: endbr64
0xa6b4: movzx   eax, byte ptr [rdi]
0xa6b7: mov     byte ptr [rip + 0x20c94], al
0xa6bd: ret

```

```

0xa6be: nop
0xa6c0: endbr64
0xa6c4: push    rbp
0xa6c5: push    rbx
0xa6c6: sub     rsp, 8
0xa6ca: cmp     dword ptr [rip + 0x1f58f], 1
0xa6d1: je      0xa720
0xa6d3: mov     rdi, qword ptr [rip + 0x1f66e]
0xa6da: test    rdi, rdi
0xa6dd: je      0xa6f0
0xa6df: lea     rdx, [rip - 0xc76]
0xa6e6: mov     esi, 1
0xa6eb: call    0x4820
0xa6f0: mov     dword ptr [rip + 0x20c52], 0
0xa6fa: mov     dword ptr [rip + 0x20c44], 0
0xa704: mov     dword ptr [rip + 0x20c36], 0
0xa70e: mov     dword ptr [rip + 0x20c28], 0
0xa718: add     rsp, 8
0xa71c: pop     rbx
0xa71d: pop     rbp
0xa71e: ret
0xa71f: nop
0xa720: mov     edx, dword ptr [rip + 0x207e2]
0xa726: cmp     edx, 0x3fb
0xa72c: jg      0xa750
0xa72e: movsxd  rax, edx
0xa731: lea     rcx, [rip + 0x207e8]
0xa738: add     edx, 4
0xa73b: add     rax, rcx
0xa73e: mov     ecx, dword ptr [rip + 0x1f5db]
0xa744: mov     dword ptr [rip + 0x207be], edx
0xa74a: mov     dword ptr [rax], ecx
0xa74c: jmp     0xa6d3
0xa6d3:
0xa74e: nop
0xa750: xor     ebx, ebx
0xa752: lea     rbp, [rip + 0x207c7]
0xa759: jmp     0xa76c
0xa76c:
0xa75b: nop     dword ptr [rax + rax]
0xa760: mov     edx, dword ptr [rip + 0x207a2]
0xa766: add     ebx, eax
0xa768: cmp     edx, ebx
0xa76a: jle     0xa785
0xa76c: sub     edx, ebx
0xa76e: movsxd  rsi, ebx
0xa771: mov     edi, 1
0xa776: add     rsi, rbp

```

```

0xa779: movsxd  rdx, edx
0xa77c: call    0x18da0
0xa781: test    eax, eax
0xa783: jns     0xa760
0xa785: mov     edx, 4
0xa78a: lea     rax, [rip + 0x2078f]
0xa791: jmp     0xa73e
0xa73e
0xa793: nop     word ptr cs:[rax + rax]
0xa79e: nop
0xa7a0: push    r12
0xa7a2: mov     r12d, edi
0xa7a5: push    rbp
0xa7a6: mov     ebp, esi
0xa7a8: sub     rsp, 8
0xa7ac: mov     eax, dword ptr [rip + 0x1f71e]
0xa7b2: test    eax, eax
0xa7b4: je      0xa800
0xa7b6: mov     eax, ebp
0xa7b8: mov     esi, dword ptr [rip + 0x201da]
0xa7be: mov     edx, dword ptr [rip + 0x201d8]
0xa7c4: or      eax, r12d
0xa7c7: mov     rdi, qword ptr [rip + 0x1f63a]
0xa7ce: js      0xa7d4
0xa7d0: cmp     ebp, esi
0xa7d2: jl      0xa810
0xa7d4: sub     edx, 1
0xa7d7: sub     esi, 1
0xa7da: call    0x4960
0xa7df: add     rsp, 8
0xa7e3: mov     esi, 1
0xa7e8: lea     rdx, [rip - 0xd7f]
0xa7ef: pop     rbp
0xa7f0: mov     rdi, rax
0xa7f3: pop     r12
0xa7f5: jmp     0x4820
0x4820
0xa7fa: nop     word ptr [rax + rax]
0xa800: xor     eax, eax
0xa802: call    0xa6c0
0xa807: jmp     0xa7b6
0xa7b6
0xa809: nop     dword ptr [rax]
0xa810: cmp     r12d, edx
0xa813: jge     0xa7d4
0xa815: mov     edx, r12d
0xa818: mov     esi, ebp
0xa81a: jmp     0xa7da

```

```

0xa7da
0xa81c: nop      dword ptr [rax]
0xa820: endbr64
0xa824: push    r15
0xa826: push    r14
0xa828: push    r13
0xa82a: push    r12
0xa82c: push    rbp
0xa82d: push    rbx
0xa82e: sub     rsp, 0x38
0xa832: mov     ebx, dword ptr [rip + 0x20160]
0xa838: imul    ebx, dword ptr [rip + 0x2015d]
0xa83f: mov     rax, qword ptr fs:[0x28]
0xa848: mov     qword ptr [rsp + 0x28], rax
0xa84d: xor     eax, eax
0xa84f: mov     rdi, qword ptr [rip + 0x1f552]
0xa856: cmp     dword ptr [rip + 0x20aff], 1
0xa85d: sbb     ebx, 0
0xa860: test    rdi, rdi
0xa863: je      0xa876
0xa865: lea     rdx, [rip - 0xdfc]
0xa86c: mov     esi, 1
0xa871: call    0x4820
0xa876: test    ebx, ebx
0xa878: jle     0xab70
0xa87e: movsxd  rax, ebx
0xa881: mov     dword ptr [rsp + 4], 0xffffffff
0xa889: xor     ebp, ebp
0xa88b: mov     r13d, 0xffffffff
0xa891: mov     qword ptr [rsp + 8], rax
0xa896: lea     r12, [rip + 0x20683]
0xa89d: jmp     0xa9ce
0xa9ce
0xa8a2: nop     word ptr [rax + rax]
0xa8a8: cdq
0xa8a9: idiv    dword ptr [rip + 0x200e9]
0xa8af: mov     r14d, edx
0xa8b2: cmp     edx, r13d
0xa8b5: jne     0xaa20
0xa8bb: cmp     eax, dword ptr [rsp + 4]
0xa8bf: jne     0xaa20
0xa8c5: mov     rax, qword ptr [rip + 0x20a64]
0xa8cc: movzx   ebx, byte ptr [rax + rbp]
0xa8d0: mov     edi, ebx
0xa8d2: mov     r14d, ebx
0xa8d5: sar     edi, 6
0xa8d8: and     r14d, 7
0xa8dc: and     edi, 1

```

```

0xa8df: call    0x9870
0xa8e4: mov     eax, ebx
0xa8e6: and     eax, 7
0xa8e9: cmp     dword ptr [rip + 0x20a5c], 1
0xa8f0: je      0xaa40
0xa8f6: cmp     dword ptr [rip + 0x1f363], 1
0xa8fd: je      0xaa5f
0xa903: xor     edi, edi
0xa905: cmp     eax, 7
0xa908: setne   dil
0xa90c: call    0x9790
0xa911: movzx   eax, byte ptr [rip + 0x20a38]
0xa918: mov     dword ptr [rip + 0x20a2a], 1
0xa922: and     eax, 0xffffffff8
0xa925: or      eax, r14d
0xa928: mov     byte ptr [rip + 0x20a22], al
0xa92e: sar     ebx, 3
0xa931: and     ebx, 7
0xa934: cmp     dword ptr [rip + 0x20a0d], 1
0xa93b: je      0xaab0
0xa941: cmp     dword ptr [rip + 0x1f318], 1
0xa948: je      0xaad8
0xa94e: xor     edi, edi
0xa950: test    ebx, ebx
0xa952: setne   dil
0xa956: call    0x9790
0xa95b: mov     edx, dword ptr [rip + 0x205a7]
0xa961: movzx   eax, byte ptr [rip + 0x209e8]
0xa968: shl     ebx, 3
0xa96b: mov     dword ptr [rip + 0x209d3], 1
0xa975: and     eax, 0xffffffffc7
0xa978: or      ebx, eax
0xa97a: mov     byte ptr [rip + 0x209d0], bl
0xa980: mov     rax, qword ptr [rip + 0x209b1]
0xa987: movzx   ebx, byte ptr [rax + rbp]
0xa98b: cmp     edx, 0x400
0xa991: je      0xab30
0xa997: lea     eax, [rdx + 1]
0xa99a: movsxd  rdx, edx
0xa99d: add     r13d, 1
0xa9a1: cmp     dword ptr [rip + 0x1fff0], r13d
0xa9a8: mov     dword ptr [rip + 0x2055a], eax
0xa9ae: mov     byte ptr [r12 + rdx], bl
0xa9b2: jne     0xa9bf
0xa9b4: mov     dword ptr [rsp + 4], 0xffffffff
0xa9bc: xor     r13d, r13d
0xa9bf: add     rbp, 1
0xa9c3: cmp     rbp, qword ptr [rsp + 8]

```

```

0xa9c8: je      0xab70
0xa9ce: mov     rdx, qword ptr [rip + 0x20953]
0xa9d5: mov     rcx, qword ptr [rip + 0x2095c]
0xa9dc: mov     eax, ebp
0xa9de: movzx   ebx, byte ptr [rdx + rbp]
0xa9e2: cmp     byte ptr [rcx + rbp], bl
0xa9e5: jne     0xa8a8
0xa9eb: mov     rdx, qword ptr [rip + 0x2092e]
0xa9f2: mov     rcx, qword ptr [rip + 0x20937]
0xa9f9: movzx   edi, byte ptr [rdx + rbp]
0xa9fd: cmp     byte ptr [rcx + rbp], dil
0xaa01: je      0xa9bf
0xaa03: cdq
0xaa04: idiv    dword ptr [rip + 0x1ff8e]
0xaa0a: mov     r14d, edx
0xaa0d: cmp     edx, r13d
0xaa10: je      0xa8bb
0xaa16: nop     word ptr cs:[rax + rax]
0xaa20: mov     esi, r14d
0xaa23: mov     edi, eax
0xaa25: mov     dword ptr [rsp + 4], eax
0xaa29: mov     r13d, r14d
0xaa2c: call    0xa7a0
0xaa31: jmp     0xa8c5
0xa8c5
0xaa36: nop     word ptr cs:[rax + rax]
0xaa40: movzx   edx, byte ptr [rip + 0x20909]
0xaa47: and     edx, 7
0xaa4a: cmp     eax, edx
0xaa4c: je      0xa92e
0xaa52: cmp     dword ptr [rip + 0x1f207], 1
0xaa59: jne     0xa903
0xaa5f: mov     eax, dword ptr [rip + 0x164b4]
0xaa65: mov     edx, dword ptr [rip + 0x2049d]
0xaa6b: mov     dword ptr [rsp + 0x10], eax
0xaa6f: movzx   eax, byte ptr [rip + 0x164a7]
0xaa76: mov     byte ptr [rsp + 0x14], al
0xaa7a: lea     eax, [r14 + 0x30]
0xaa7e: mov     byte ptr [rsp + 0x13], al
0xaa82: cmp     edx, 0x3fa
0xaa88: jg      0xac18
0xaa8e: movsxd  rax, edx
0xaa91: add     edx, 5
0xaa94: add     rax, r12
0xaa97: mov     ecx, dword ptr [rsp + 0x10]
0xaa9b: mov     dword ptr [rip + 0x20467], edx
0xaaa1: mov     dword ptr [rax], ecx
0xaaa3: movzx   ecx, byte ptr [rsp + 0x14]

```



```

0xaaa8: mov     byte ptr [rax + 4], cl
0xaaab: jmp     0xa911
0xa911
0xaab0: movzx    eax, byte ptr [rip + 0x20899]
0xaab7: mov     edx, dword ptr [rip + 0x2044b]
0xaabd: sar     eax, 3
0xaac0: and     eax, 7
0xaac3: cmp     ebx, eax
0xaac5: je      0xa980
0xaacb: cmp     dword ptr [rip + 0x1f18e], 1
0xaad2: jne     0xa94e
0xaad8: mov     eax, dword ptr [rip + 0x16436]
0xaade: mov     edx, dword ptr [rip + 0x20424]
0xaae4: mov     dword ptr [rsp + 0x10], eax
0xaae8: movzx    eax, byte ptr [rip + 0x16429]
0xaaef: mov     byte ptr [rsp + 0x14], al
0xaaf3: lea     eax, [rbx + 0x30]
0xaaf6: mov     byte ptr [rsp + 0x13], al
0xaafa: cmp     edx, 0x3fa
0xab00: jg      0xac58
0xab06: movsxd   rax, edx
0xab09: add     edx, 5
0xab0c: add     rax, r12
0xab0f: mov     ecx, dword ptr [rsp + 0x10]
0xab13: mov     dword ptr [rip + 0x203ef], edx
0xab19: mov     dword ptr [rax], ecx
0xab1b: movzx    ecx, byte ptr [rsp + 0x14]
0xab20: mov     byte ptr [rax + 4], cl
0xab23: jmp     0xa961
0xa961
0xab28: nop     dword ptr [rax + rax]
0xab30: xor     r14d, r14d
0xab33: jmp     0xab46
0xab46
0xab35: nop     dword ptr [rax]
0xab38: mov     edx, dword ptr [rip + 0x203ca]
0xab3e: add     r14d, eax
0xab41: cmp     edx, r14d
0xab44: jle     0xab60
0xab46: sub     edx, r14d
0xab49: movsxd   rsi, r14d
0xab4c: mov     edi, 1
0xab51: add     rsi, r12
0xab54: movsxd   rdx, edx
0xab57: call    0x18da0
0xab5c: test    eax, eax
0xab5e: jns     0xab38
0xab60: mov     eax, 1

```

```

0xab65: xor     edx, edx
0xab67: jmp     0xa99a
0xa99a:
0xab6c: nop     dword ptr [rax]
0xab70: mov     edi, dword ptr [rip + 0x207de]
0xab76: mov     esi, dword ptr [rip + 0x207dc]
0xab7c: call    0xa7a0
0xab81: mov     rdi, qword ptr [rip + 0x1f240]
0xab88: test    rdi, rdi
0xab8b: je      0xab9e
0xab8d: lea     rdx, [rip - 0x1124]
0xab94: mov     esi, 1
0xab99: call    0x4820
0xab9e: mov     edx, dword ptr [rip + 0x20364]
0xaba4: test    edx, edx
0xaba6: jne     0xac98
0xabac: mov     edx, dword ptr [rip + 0x1fde6]
0xabb2: imul    edx, dword ptr [rip + 0x1fde3]
0xabb9: mov     rsi, qword ptr [rip + 0x20778]
0xabc0: mov     rdi, qword ptr [rip + 0x20761]
0xabc7: movsxd  rdx, edx
0xabca: call    0x4bb0
0xabcf: mov     edx, dword ptr [rip + 0x1fdc3]
0xabd5: imul    edx, dword ptr [rip + 0x1fdc0]
0xabdc: mov     rdi, qword ptr [rip + 0x2073d]
0xabe3: mov     rsi, qword ptr [rip + 0x20746]
0xabea: movsxd  rdx, edx
0xabed: mov     rax, qword ptr [rsp + 0x28]
0xabf2: sub     rax, qword ptr fs:[0x28]
0xabfb: jne     0xace4
0xac01: add     rsp, 0x38
0xac05: pop     rbx
0xac06: pop     rbp
0xac07: pop     r12
0xac09: pop     r13
0xac0b: pop     r14
0xac0d: pop     r15
0xac0f: jmp     0x4bb0
0x4bb0:
0xac14: nop     dword ptr [rax]
0xac18: xor     r15d, r15d
0xac1b: jmp     0xac2e
0xac2e:
0xac1d: nop     dword ptr [rax]
0xac20: mov     edx, dword ptr [rip + 0x202e2]
0xac26: add     r15d, eax
0xac29: cmp     edx, r15d
0xac2c: jle     0xac48

```

```

0xac2e: sub     edx, r15d
0xac31: movsxd  rsi, r15d
0xac34: mov     edi, 1
0xac39: add     rsi, r12
0xac3c: movsxd  rdx, edx
0xac3f: call    0x18da0
0xac44: test    eax, eax
0xac46: jns     0xac20
0xac48: mov     edx, 5
0xac4d: mov     rax, r12
0xac50: jmp     0xaa97
0xaa97:
0xac55: nop     dword ptr [rax]
0xac58: xor     r14d, r14d
0xac5b: jmp     0xac6e
0xac6e:
0xac5d: nop     dword ptr [rax]
0xac60: mov     edx, dword ptr [rip + 0x202a2]
0xac66: add     r14d, eax
0xac69: cmp     edx, r14d
0xac6c: jle     0xac88
0xac6e: sub     edx, r14d
0xac71: movsxd  rsi, r14d
0xac74: mov     edi, 1
0xac79: add     rsi, r12
0xac7c: movsxd  rdx, edx
0xac7f: call    0x18da0
0xac84: test    eax, eax
0xac86: jns     0xac60
0xac88: mov     edx, 5
0xac8d: mov     rax, r12
0xac90: jmp     0xab0f
0xab0f:
0xac95: nop     dword ptr [rax]
0xac98: mov     ebx, 0
0xac9d: lea     rbp, [rip + 0x2027c]
0xaca4: jg      0xacbc
0xaca6: jmp     0acd5
0acd5:
0xaca8: nop     dword ptr [rax + rax]
0xacb0: mov     edx, dword ptr [rip + 0x20252]
0xacb6: add     ebx, eax
0xacb8: cmp     edx, ebx
0xacba: jle     0acd5
0xacbc: sub     edx, ebx
0xacbe: movsxd  rsi, ebx
0xacc1: mov     edi, 1
0xacc6: add     rsi, rbp

```

```

0xacc9: movsxd  rdx, edx
0xaccc: call    0x18da0
0xacd1: test    eax, eax
0xacd3: jns     0xacb0
0xacd5: mov     dword ptr [rip + 0x20229], 0
0xacdf: jmp     0xabac
0xabac
0xace4: call    0x49d0
0xace9: nop     dword ptr [rax]
0xacf0: endbr64
0xacf4: push    r12
0xacf6: xor     eax, eax
0xacf8: mov     r12, rdi
0xacfb: push    rbx
0xacfc: mov     ebx, esi
0xacfe: sub     rsp, 8
0xad02: call    0xa820
0xad07: mov     rdx, qword ptr [rip + 0x1fc1a]
0xad0e: test    rdx, rdx
0xad11: je      0xad17
0xad13: xor     eax, eax
0xad15: call    rdx
0xad17: movsxd  rdx, ebx
0xad1a: mov     rsi, r12
0xad1d: xor     edi, edi
0xad1f: call    0x18d30
0xad24: mov     rdx, qword ptr [rip + 0x1fbf5]
0xad2b: mov     r12d, eax
0xad2e: test    rdx, rdx
0xad31: je      0xad37
0xad33: xor     eax, eax
0xad35: call    rdx
0xad37: add     rsp, 8
0xad3b: mov     eax, r12d
0xad3e: pop     rbx
0xad3f: pop     r12
0xad41: ret
0xad42: nop     word ptr cs:[rax + rax]
0xad4d: nop     dword ptr [rax]
0xad50: push    r12
0xad52: mov     edi, 1
0xad57: push    rbp
0xad58: lea     rbp, [rip + 0x1fda1]
0xad5f: push    rbx
0xad60: sub     rsp, 0x50
0xad64: mov     rax, qword ptr fs:[0x28]
0xad6d: mov     qword ptr [rsp + 0x48], rax
0xad72: xor     eax, eax

```

```

0xad74: mov     r12, rsp
0xad77: mov     rsi, r12
0xad7a: call    0x4d60
0xad7f: mov     edx, 0xff
0xad84: mov     esi, 1
0xad89: mov     edi, 1
0xad8e: mov     word ptr [rsp + 0x11], dx
0xad93: mov     rdx, r12
0xad96: call    0x4d70
0xad9b: mov     edi, 1
0xada0: mov     dword ptr [rip + 0x1ef7a], 0xffffffff
0xadaa: call    0x18b70
0xadaf: mov     dword ptr [rip + 0x20147], 0
0xadb9: nop     dword ptr [rax]
0xadc0: xor     eax, eax
0xadc2: call    0xa820
0xadc7: mov     rdx, qword ptr [rip + 0x1fb5a]
0xadce: test    rdx, rdx
0xadd1: je      0xadd7
0xadd3: xor     eax, eax
0xadd5: call    rdx
0xadd7: mov     edx, 0x400
0xaddc: mov     rsi, rbp
0xaddf: xor     edi, edi
0xade1: call    0x18d30
0xade6: mov     rdx, qword ptr [rip + 0x1fb33]
0xaded: mov     ebx, eax
0xadeb: test    rdx, rdx
0xadf2: je      0xadf8
0xadf4: xor     eax, eax
0xadf6: call    rdx
0xadf8: mov     dword ptr [rip + 0x20106], ebx
0xadfe: test    ebx, ebx
0xae00: js      0xadc0
0xae02: xor     edi, edi
0xae04: call    0x18b70
0xae09: mov     rsi, r12
0xae0c: mov     edi, 1
0xae11: call    0x4d60
0xae16: mov     eax, 7
0xae1b: mov     rdx, r12
0xae1e: mov     esi, 1
0xae23: mov     edi, 1
0xae28: mov     word ptr [rsp + 0x11], ax
0xae2d: call    0x4d70
0xae32: mov     eax, dword ptr [rip + 0x200cc]
0xae38: mov     dword ptr [rip + 0x1eee2], 7
0xae42: test    eax, eax

```

```

0xae44: je      0xae9f
0xae46: movsxd  rsi, dword ptr [rip + 0x200b3]
0xae4d: movsx   r8d, byte ptr [rbp + rsi]
0xae53: mov     rdx, rsi
0xae56: mov     ecx, r8d
0xae59: test    r8b, r8b
0xae5c: js      0xae90
0xae5e: sub     eax, 1
0xae61: add     edx, 1
0xae64: mov     dword ptr [rip + 0x2009a], eax
0xae6a: mov     dword ptr [rip + 0x20090], edx
0xae70: mov     rax, qword ptr [rsp + 0x48]
0xae75: sub     rax, qword ptr fs:[0x28]
0xae7e: jne     0xaea7
0xae80: add     rsp, 0x50
0xae84: mov     eax, r8d
0xae87: pop     rbx
0xae88: pop     rbp
0xae89: pop     r12
0xae8b: ret
0xae8c: nop     dword ptr [rax]
0xae90: and     ecx, 0x7f
0xae93: mov     r8d, 0x1b
0xae99: mov     byte ptr [rbp + rsi], cl
0xae9d: jmp     0xae70
0xae70:
0xae9f: mov     r8d, 0xffffffff
0xaea5: jmp     0xae70
0xae70:
0xaea7: call    0x49d0
0xaeac: nop     dword ptr [rax]
0xaeb0: endbr64
0xaeb4: xor     eax, eax
0xaeb6: sub     rsp, 8
0xaeba: call    0x18ab0
0xaebf: mov     eax, dword ptr [rip + 0x2003f]
0xaec5: test    eax, eax
0xaec7: jne     0xaed8
0xaec9: xor     eax, eax
0xaecb: add     rsp, 8
0xaecf: jmp     0xad50
0xad50:
0xaed4: nop     dword ptr [rax]
0xaed8: movsxd  rsi, dword ptr [rip + 0x20021]
0xaedf: lea     rdi, [rip + 0x1fc1a]
0xae6: movsx   r8d, byte ptr [rdi + rsi]
0xaeeb: mov     rdx, rsi
0xaeee: mov     ecx, r8d

```

```

0xaef1: test    r8b, r8b
0xaef4: js      0xaf10
0xaef6: sub     eax, 1
0xaef9: add     edx, 1
0xaefc: mov     dword ptr [rip + 0x20002], eax
0xaf02: mov     dword ptr [rip + 0x1fff8], edx
0xaf08: mov     eax, r8d
0xaf0b: add     rsp, 8
0xaf0f: ret
0xaf10: and     ecx, 0x7f
0xaf13: mov     r8d, 0x1b
0xaf19: mov     byte ptr [rdi + rsi], cl
0xaf1c: jmp     0xaf08
0xaf08
0xaf1e: nop
0xaf20: endbr64
0xaf24: push    r13
0xaf26: mov     r13, rsi
0xaf29: push    r12
0xaf2b: mov     r12, rdx
0xaf2e: push    rbp
0xaf2f: mov     rbp, rdi
0xaf32: mov     edi, 0x18
0xaf37: push    rbx
0xaf38: sub     rsp, 8
0xaf3c: call    0x1d670
0xaf41: mov     rdi, r13
0xaf44: mov     rbx, rax
0xaf47: call    0x1d7a0
0xaf4c: mov     qword ptr [rbx + 0x10], r12
0xaf50: mov     qword ptr [rbx], rax
0xaf53: mov     rax, qword ptr [rbp]
0xaf57: mov     qword ptr [rbx + 8], rax
0xaf5b: mov     qword ptr [rbp], rbx
0xaf5f: add     rsp, 8
0xaf63: pop     rbx
0xaf64: pop     rbp
0xaf65: pop     r12
0xaf67: pop     r13
0xaf69: ret
0xaf6a: nop     word ptr [rax + rax]
0xaf70: endbr64
0xaf74: cmp     byte ptr [rdi], 0
0xaf77: je      0xb030
0xaf7d: push    r13
0xaf7f: push    r12
0xaf81: mov     r12, rsi
0xaf84: push    rbp

```

```

0xaf85: mov     rbp, rdi
0xaf88: push    rbx
0xaf89: sub     rsp, 8
0xaf8d: mov     r13, qword ptr [rip + 0x1fa0c]
0xaf94: test    r13, r13
0xaf97: je      0xb010
0xaf99: mov     rbx, qword ptr [r13]
0xaf9d: mov     rsi, qword ptr [rbx]
0xafa0: call    0x4b30
0xafa5: test    eax, eax
0xafa7: jle     0xb010
0xafa9: nop     dword ptr [rax]
0xafb0: lea     r13, [rbx + 8]
0xafb4: mov     rbx, qword ptr [rbx + 8]
0xafb8: mov     qword ptr [rip + 0x1f9e1], r13
0xafbf: test    rbx, rbx
0xafc2: je      0xafd3
0xafc4: mov     rsi, qword ptr [rbx]
0xafc7: mov     rdi, rbp
0xafca: call    0x4b30
0xafcf: test    eax, eax
0xafd1: jg      0xafb0
0xafd3: mov     edi, 0x18
0xafd8: call    0x1d670
0xafdd: mov     rdi, rbp
0xafe0: mov     rbx, rax
0xafe3: call    0x1d7a0
0xafe8: mov     qword ptr [rbx + 0x10], r12
0xafec: mov     qword ptr [rbx], rax
0xafef: mov     rax, qword ptr [r13]
0xaff3: mov     qword ptr [rbx + 8], rax
0xaff7: mov     qword ptr [r13], rbx
0xaffb: add     rsp, 8
0xafff: pop     rbx
0xb000: pop     rbp
0xb001: pop     r12
0xb003: pop     r13
0xb005: ret
0xb006: nop     word ptr cs:[rax + rax]
0xb010: mov     rbx, qword ptr [rip + 0x1f969]
0xb017: lea     r13, [rip + 0x1f962]
0xb01e: mov     qword ptr [rip + 0x1f97b], r13
0xb025: test    rbx, rbx
0xb028: jne     0xafc4
0xb02a: jmp     0xafd3
0xafd3:
0xb02c: nop     dword ptr [rax]
0xb030: ret

```



```

0xb031: nop      word ptr cs:[rax + rax]
0xb03c: nop      dword ptr [rax]
0xb040: endbr64
0xb044: mov      rax, qword ptr [rip + 0x1f935]
0xb04b: mov      qword ptr [rip + 0x1f926], rax
0xb052: ret
0xb053: nop      word ptr cs:[rax + rax]
0xb05e: nop
0xb060: endbr64
0xb064: cmp      qword ptr [rip + 0x1f90c], 0
0xb06c: je       0xb073
0xb06e: jmp      0x99e0
0x99e0
0xb073: xor      eax, eax
0xb075: ret
0xb076: nop      word ptr cs:[rax + rax]
0xb080: endbr64
0xb084: mov      eax, dword ptr [rip + 0x1f912]
0xb08a: push     r12
0xb08c: mov      r12, rdi
0xb08f: push     rbp
0xb090: push     rbx
0xb091: sub      eax, 1
0xb094: movzx    ebx, byte ptr [rip + 0x202b6]
0xb09b: mov      dword ptr [rip + 0x202b3], eax
0xb0a1: mov      dword ptr [rip + 0x202ad], 0
0xb0ab: mov      eax, ebx
0xb0ad: and      eax, 0xffffffffc0
0xb0b0: or       eax, 0x38
0xb0b3: mov      byte ptr [rip + 0x20298], al
0xb0b9: mov      eax, dword ptr [rip + 0x1f8d9]
0xb0bf: lea      edi, [rax + 1]
0xb0c2: movsxd   rdi, edi
0xb0c5: call     0x1d670
0xb0ca: movsxd   rdx, dword ptr [rip + 0x1f8c7]
0xb0d1: mov      esi, 0x20
0xb0d6: mov      rdi, rax
0xb0d9: mov      rbp, rax
0xb0dc: call     0x4a50
0xb0e1: movsxd   rax, dword ptr [rip + 0x1f8b0]
0xb0e8: mov      rdi, rbp
0xb0eb: mov      byte ptr [rbp + rax], 0
0xb0f0: mov      esi, dword ptr [rip + 0x1f8a2]
0xb0f6: call     0xa1a0
0xb0fb: mov      rdi, rbp
0xb0fe: lea      rbp, [rip + 0x1f8bb]
0xb105: call     0x18d10
0xb10a: mov      eax, dword ptr [rip + 0x1f88c]

```

```

0xb110: mov     rdi, r12
0xb113: mov     dword ptr [rip + 0x2023b], 0
0xb11d: sub     eax, 1
0xb120: mov     dword ptr [rip + 0x2022e], eax
0xb126: call    0x9fe0
0xb12b: mov     esi, 0x13
0xb130: lea     rdi, [rip + 0x15ad5]
0xb137: call    0xa1a0
0xb13c: mov     rdi, rbp
0xb13f: call    0x49b0
0xb144: mov     rdi, rbp
0xb147: mov     esi, eax
0xb149: call    0xa1a0
0xb14e: mov     esi, 1
0xb153: lea     rdi, [rip + 0x155eb]
0xb15a: call    0xa1a0
0xb15f: xor     eax, eax
0xb161: call    0xa820
0xb166: mov     byte ptr [rip + 0x201e5], bl
0xb16c: pop     rbx
0xb16d: pop     rbp
0xb16e: pop     r12
0xb170: ret
0xb171: nop     word ptr cs:[rax + rax]
0xb17c: nop     dword ptr [rax]
0xb180: endbr64
0xb184: push    r15
0xb186: push    r14
0xb188: mov     r14, rdi
0xb18b: push    r13
0xb18d: push    r12
0xb18f: push    rbp
0xb190: lea     rbp, [rip + 0x1f969]
0xb197: push    rbx
0xb198: sub     rsp, 0x28
0xb19c: jmp     0xb1ae
0xb1ae:
0xb19e: nop
0xb1a0: xor     eax, eax
0xb1a2: call    0xad50
0xb1a7: mov     ebx, eax
0xb1a9: cmp     ebx, -1
0xb1ac: jne     0xb1ef
0xb1ae: xor     eax, eax
0xb1b0: call    0x18ab0
0xb1b5: mov     eax, dword ptr [rip + 0x1fd49]
0xb1bb: test    eax, eax
0xb1bd: je      0xb1a0

```

```

0xb1bf: movsxd  rsi, dword ptr [rip + 0x1fd3a]
0xb1c6: movsx  ebx, byte ptr [rbp + rsi]
0xb1cb: mov    rdx, rsi
0xb1ce: mov    ecx, ebx
0xb1d0: test   bl, bl
0xb1d2: js     0xb4a0
0xb1d8: sub    eax, 1
0xb1db: add    edx, 1
0xb1de: mov    dword ptr [rip + 0x1fd20], eax
0xb1e4: mov    dword ptr [rip + 0x1fd16], edx
0xb1ea: cmp    ebx, -1
0xb1ed: je     0xb1ae
0xb1ef: test   r14, r14
0xb1f2: je     0xb1fb
0xb1f4: mov    dword ptr [r14], 1
0xb1fb: mov    eax, dword ptr [rip + 0x201fb]
0xb201: test   ebx, ebx
0xb203: jne    0xb445
0xb209: mov    ebx, 0xff
0xb20e: test   eax, eax
0xb210: jne    0xb240
0xb212: xor    r12d, r12d
0xb215: mov    ebp, 0x1fe
0xb21a: mov    ebx, 0xff
0xb21f: call   0x4f00
0xb224: mov    rax, qword ptr [rax]
0xb227: movzx  eax, word ptr [rax + rbp]
0xb22b: test   ah, 0x40
0xb22e: jne    0xb47d
0xb234: test   r12b, r12b
0xb237: jne    0xb47d
0xb23d: nop    dword ptr [rax]
0xb240: mov    rax, qword ptr [rip + 0x1f739]
0xb247: xor    r12d, r12d
0xb24a: lea    r13, [rip + 0x1f8af]
0xb251: mov    dword ptr [rip + 0x1f751], 0
0xb25b: mov    dword ptr [rip + 0x1f743], 0
0xb265: mov    qword ptr [rip + 0x1f70c], rax
0xb26c: test   ebx, ebx
0xb26e: mov    eax, 0xffffffff
0xb273: movsxd  rbp, r12d
0xb276: cmovne  eax, ebx
0xb279: mov    rbx, qword ptr [rip + 0x200e0]
0xb280: mov    byte ptr [rbx + rbp], al
0xb283: add    rbp, 1
0xb287: mov    byte ptr [rbx + rbp], 0
0xb28b: cmp    qword ptr [rip + 0x1f6e5], 0
0xb293: je     0xb2b3

```

```

0xb295: mov     rdi, rbx
0xb298: call    0x99e0
0xb29d: mov     r15, rax
0xb2a0: cmp     rax, -1
0xb2a4: je      0xb33a
0xb2aa: test    rax, rax
0xb2ad: jne     0xb38f
0xb2b3: mov     ecx, dword ptr [rip + 0x1fc4b]
0xb2b9: test    ecx, ecx
0xb2bb: jne     0xb2ee
0xb2bd: mov     edx, dword ptr [rip + 0x1f6e5]
0xb2c3: test    edx, edx
0xb2c5: je      0xb380
0xb2cb: mov     rdi, rbx
0xb2ce: call    0xb080
0xb2d3: mov     dword ptr [rip + 0x1f6cf], 1
0xb2dd: jmp     0xb2ee
0xb2ee
0xb2df: nop
0xb2e0: xor     eax, eax
0xb2e2: call    0xad50
0xb2e7: mov     ebx, eax
0xb2e9: cmp     ebx, -1
0xb2ec: jne     0xb32c
0xb2ee: xor     eax, eax
0xb2f0: call    0x18ab0
0xb2f5: mov     eax, dword ptr [rip + 0x1fc09]
0xb2fb: test    eax, eax
0xb2fd: je      0xb2e0
0xb2ff: movsxd  rsi, dword ptr [rip + 0x1fbfa]
0xb306: movsx   ebx, byte ptr [r13 + rsi]
0xb30c: mov     rdx, rsi
0xb30f: mov     ecx, ebx
0xb311: test    bl, bl
0xb313: js      0xb360
0xb315: sub     eax, 1
0xb318: add     edx, 1
0xb31b: mov     dword ptr [rip + 0x1fbe3], eax
0xb321: mov     dword ptr [rip + 0x1fbd9], edx
0xb327: cmp     ebx, -1
0xb32a: je      0xb2ee
0xb32c: add     r12d, 1
0xb330: cmp     r12d, 0xf
0xb334: jne     0xb26c
0xb33a: mov     edi, 1
0xb33f: xor     r15d, r15d
0xb342: call    0x4aa0
0xb347: mov     dword ptr [rip + 0x1f65b], 0

```

```

0xb351: jmp      0xb433
0xb433
0xb356: nop      word ptr cs:[rax + rax]
0xb360: and      ecx, 0x7f
0xb363: add      r12d, 1
0xb367: mov      byte ptr [r13 + rsi], cl
0xb36c: cmp      r12d, 0xf
0xb370: je       0xb33a
0xb372: mov      eax, 0x1b
0xb377: jmp      0xb279
0xb279
0xb37c: nop      dword ptr [rax]
0xb380: mov      edi, 1
0xb385: call     0x4aa0
0xb38a: jmp      0xb2d3
0xb2d3
0xb38f: test     r14, r14
0xb392: je       0xb41f
0xb398: mov      ecx, dword ptr [rip + 0x1fb66]
0xb39e: cmp      r12d, ecx
0xb3a1: jge     0xb41f
0xb3a3: movsxd   rax, dword ptr [rip + 0x1fb56]
0xb3aa: lea      r9d, [r12 + 1]
0xb3af: lea      rdx, [rip + 0x1f74a]
0xb3b6: movsxd   rdi, r9d
0xb3b9: mov      qword ptr [rsp + 0x18], rdi
0xb3be: lea      r8d, [r9 + rax]
0xb3c2: lea      r13, [rax + rdx]
0xb3c6: jmp      0xb3f1
0xb3f1
0xb3c8: nop      dword ptr [rax + rax]
0xb3d0: sub      ecx, r9d
0xb3d3: add      dword ptr [r14], 1
0xb3d7: add      r13, qword ptr [rsp + 0x18]
0xb3dc: mov      dword ptr [rip + 0x1fb1d], r8d
0xb3e3: add      r8d, r9d
0xb3e6: mov      dword ptr [rip + 0x1fb18], ecx
0xb3ec: cmp      r12d, ecx
0xb3ef: jge     0xb41f
0xb3f1: mov      rdx, rbp
0xb3f4: mov      rsi, r13
0xb3f7: mov      rdi, rbx
0xb3fa: mov      dword ptr [rsp + 0x14], r8d
0xb3ff: mov      dword ptr [rsp + 0x10], r9d
0xb404: mov      dword ptr [rsp + 0xc], ecx
0xb408: call     0x4af0
0xb40d: mov      ecx, dword ptr [rsp + 0xc]
0xb411: mov      r9d, dword ptr [rsp + 0x10]

```

```

0xb416: test    eax, eax
0xb418: mov     r8d, dword ptr [rsp + 0x14]
0xb41d: je      0xb3d0
0xb41f: mov     edi, 1
0xb424: call    0x4aa0
0xb429: mov     dword ptr [rip + 0x1f579], 0
0xb433: add     rsp, 0x28
0xb437: mov     rax, r15
0xb43a: pop     rbx
0xb43b: pop     rbp
0xb43c: pop     r12
0xb43e: pop     r13
0xb440: pop     r14
0xb442: pop     r15
0xb444: ret
0xb445: test    eax, eax
0xb447: jne     0xb240
0xb44d: cmp     ebx, 0xa
0xb450: sete    bpl
0xb454: cmp     ebx, 0xd
0xb457: sete    al
0xb45a: or      ebp, eax
0xb45c: call    0x4f00
0xb461: mov     rax, qword ptr [rax]
0xb464: test    bpl, bpl
0xb467: je      0xb4c6
0xb469: mov     ebx, 0xa
0xb46e: test    byte ptr [rax + 0x15], 0x40
0xb472: je      0xb240
0xb478: mov     ebx, 0xa
0xb47d: mov     rax, qword ptr [rip + 0x1f4dc]
0xb484: lea     r15, [rip + 0x1f4d5]
0xb48b: mov     byte ptr [rax], bl
0xb48d: mov     rax, qword ptr [rip + 0x1f4cc]
0xb494: mov     byte ptr [rax + 1], 0
0xb498: jmp     0xb433
0xb433:
0xb49a: nop     word ptr [rax + rax]
0xb4a0: and     ecx, 0x7f
0xb4a3: mov     byte ptr [rbp + rsi], cl
0xb4a7: test    r14, r14
0xb4aa: je      0xb4b3
0xb4ac: mov     dword ptr [r14], 1
0xb4b3: mov     eax, dword ptr [rip + 0x1ff43]
0xb4b9: mov     ebx, 0x1b
0xb4be: test    eax, eax
0xb4c0: jne     0xb240
0xb4c6: movsxd  rbp, ebx

```

```

0xb4c9: add      rbp, rbp
0xb4cc: cmp      ebx, 7
0xb4cf: sete     r12b
0xb4d3: jmp      0xb21f
0xb21f:
0xb4d8: nop      dword ptr [rax + rax]
0xb4e0: endbr64
0xb4e4: mov      eax, dword ptr [rip + 0x1f4c2]
0xb4ea: test     eax, eax
0xb4ec: jne      0xb4f0
0xb4ee: ret
0xb4ef: nop
0xb4f0: sub      rsp, 8
0xb4f4: mov      rdi, qword ptr [rip + 0x1fe65]
0xb4fb: call     0xb080
0xb500: mov      dword ptr [rip + 0x1f49e], 1
0xb50a: add      rsp, 8
0xb50e: ret
0xb50f: nop
0xb510: endbr64
0xb514: mov      rax, qword ptr [rip + 0x1fe45]
0xb51b: ret
0xb51c: nop      dword ptr [rax]
0xb520: endbr64
0xb524: push     r15
0xb526: mov      esi, 0x5413
0xb52b: mov      edi, 1
0xb530: push     r14
0xb532: push     r13
0xb534: push     r12
0xb536: push     rbp
0xb537: push     rbx
0xb538: sub      rsp, 0x58
0xb53c: mov      rax, qword ptr fs:[0x28]
0xb545: mov      qword ptr [rsp + 0x48], rax
0xb54a: xor      eax, eax
0xb54c: lea      rdx, [rsp + 0x18]
0xb551: mov      dword ptr [rsp + 0x10], 0
0xb559: mov      dword ptr [rsp + 0x14], 0
0xb561: call     0x4a70
0xb566: test     eax, eax
0xb568: jne      0xb5a0
0xb56a: mov      ebx, eax
0xb56c: movzx    eax, word ptr [rsp + 0x1a]
0xb571: test     ax, ax
0xb574: je       0xb790
0xb57a: movzx    ebp, word ptr [rsp + 0x18]
0xb57f: test     bp, bp

```

```

0xb582: je      0xb790
0xb588: movzx    ebx, ax
0xb58b: lea      r12d, [rbp - 1]
0xb58f: lea      eax, [rbx - 1]
0xb592: mov      dword ptr [rsp + 0xc], eax
0xb596: jmp      0xb5b2
0xb5b2:
0xb598: nop      dword ptr [rax + rax]
0xb5a0: mov      dword ptr [rsp + 0xc], 0xffffffff
0xb5a8: mov      r12d, 0xffffffff
0xb5ae: xor      ebx, ebx
0xb5b0: xor      ebp, ebp
0xb5b2: lea      r13, [rip + 0x15667]
0xb5b9: lea      r15, [rip + 0x15a13]
0xb5c0: mov      rdi, r13
0xb5c3: call     0x47e0
0xb5c8: mov      rdi, rax
0xb5cb: test     rax, rax
0xb5ce: je      0xb5df
0xb5d0: lea      rdx, [rsp + 0x14]
0xb5d5: mov      rsi, r15
0xb5d8: xor      eax, eax
0xb5da: call     0x4c60
0xb5df: lea      r14, [rip + 0x15642]
0xb5e6: mov      rdi, r14
0xb5e9: call     0x47e0
0xb5ee: mov      rdi, rax
0xb5f1: test     rax, rax
0xb5f4: je      0xb605
0xb5f6: lea      rdx, [rsp + 0x10]
0xb5fb: mov      rsi, r15
0xb5fe: xor      eax, eax
0xb600: call     0x4c60
0xb605: cmp      dword ptr [rsp + 0xc], 0x3ff
0xb60d: mov      eax, dword ptr [rip + 0x1e8fd]
0xb613: jbe      0xb628
0xb615: mov      ebx, dword ptr [rsp + 0x14]
0xb619: lea      edx, [rbx - 1]
0xb61c: cmp      edx, 0x3ff
0xb622: ja      0xb7c0
0xb628: mov      dword ptr [rip + 0x1f36a], ebx
0xb62e: cmp      r12d, 0x3ff
0xb635: jbe      0xb64a
0xb637: mov      ebp, dword ptr [rsp + 0x10]
0xb63b: lea      edx, [rbp - 1]
0xb63e: cmp      edx, 0x3ff
0xb644: ja      0xb7a8
0xb64a: lea      r12, [rsp + 0x20]

```



```

0xb64f: mov     r8d, ebp
0xb652: mov     rcx, r15
0xb655: mov     edx, 0x20
0xb65a: mov     esi, 1
0xb65f: mov     rdi, r12
0xb662: xor     eax, eax
0xb664: mov     dword ptr [rip + 0x1f332], ebp
0xb66a: call    0x4f10
0xb66f: mov     rsi, r12
0xb672: mov     rdi, r14
0xb675: call    0x18380
0xb67a: mov     rcx, r15
0xb67d: mov     edx, 0x20
0xb682: mov     rdi, r12
0xb685: mov     r8d, dword ptr [rip + 0x1f30c]
0xb68c: mov     esi, 1
0xb691: xor     eax, eax
0xb693: call    0x4f10
0xb698: mov     rdi, r13
0xb69b: mov     rsi, r12
0xb69e: call    0x18380
0xb6a3: mov     rdi, qword ptr [rip + 0x1fc8e]
0xb6aa: test    rdi, rdi
0xb6ad: je      0xb6b4
0xb6af: call    0x18d10
0xb6b4: mov     rdi, qword ptr [rip + 0x1fc75]
0xb6bb: test    rdi, rdi
0xb6be: je      0xb6c5
0xb6c0: call    0x18d10
0xb6c5: mov     rdi, qword ptr [rip + 0x1fc5c]
0xb6cc: test    rdi, rdi
0xb6cf: je      0xb6d6
0xb6d1: call    0x18d10
0xb6d6: mov     rdi, qword ptr [rip + 0x1fc43]
0xb6dd: test    rdi, rdi
0xb6e0: je      0xb6e7
0xb6e2: call    0x18d10
0xb6e7: mov     edi, dword ptr [rip + 0x1f2ab]
0xb6ed: imul    edi, dword ptr [rip + 0x1f2a8]
0xb6f4: mov     esi, 1
0xb6f9: movsxd  rdi, edi
0xb6fc: call    0x1d740
0xb701: mov     edi, dword ptr [rip + 0x1f291]
0xb707: mov     esi, 1
0xb70c: imul    edi, dword ptr [rip + 0x1f289]
0xb713: mov     qword ptr [rip + 0x1fc1e], rax
0xb71a: movsxd  rdi, edi
0xb71d: call    0x1d740

```

```

0xb722: mov     edi, dword ptr [rip + 0x1f270]
0xb728: mov     esi, 1
0xb72d: imul    edi, dword ptr [rip + 0x1f268]
0xb734: mov     qword ptr [rip + 0x1fbf5], rax
0xb73b: movsxd  rdi, edi
0xb73e: call    0x1d740
0xb743: mov     edi, dword ptr [rip + 0x1f24f]
0xb749: mov     esi, 1
0xb74e: imul    edi, dword ptr [rip + 0x1f247]
0xb755: mov     qword ptr [rip + 0x1fbcc], rax
0xb75c: movsxd  rdi, edi
0xb75f: call    0x1d740
0xb764: mov     qword ptr [rip + 0x1fbb5], rax
0xb76b: mov     rax, qword ptr [rsp + 0x48]
0xb770: sub     rax, qword ptr fs:[0x28]
0xb779: jne     0xb7dc
0xb77b: add     rsp, 0x58
0xb77f: pop     rbx
0xb780: pop     rbp
0xb781: pop     r12
0xb783: pop     r13
0xb785: pop     r14
0xb787: pop     r15
0xb789: ret
0xb78a: nop     word ptr [rax + rax]
0xb790: mov     dword ptr [rsp + 0xc], 0xffffffff
0xb798: xor     ebp, ebp
0xb79a: mov     r12d, 0xffffffff
0xb7a0: jmp     0xb5b2
0xb5b2
0xb7a5: nop     dword ptr [rax]
0xb7a8: lea     edx, [rax - 1]
0xb7ab: mov     ebp, 0x18
0xb7b0: cmp     edx, 0x400
0xb7b6: cmovb   ebp, eax
0xb7b9: jmp     0xb64a
0xb64a
0xb7be: nop
0xb7c0: mov     ebx, dword ptr [rip + 0x1e72a]
0xb7c6: lea     edx, [rbx - 1]
0xb7c9: cmp     edx, 0x3ff
0xb7cf: mov     edx, 0x50
0xb7d4: cmova   ebx, edx
0xb7d7: jmp     0xb628
0xb628
0xb7dc: call    0x49d0
0xb7e1: nop     word ptr cs:[rax + rax]
0xb7ec: nop     dword ptr [rax]

```

```

0xb7f0: endbr64
0xb7f4: push    r15
0xb7f6: push    r14
0xb7f8: push    r13
0xb7fa: push    r12
0xb7fc: mov     r12, rdi
0xb7ff: mov     edi, 0x5858
0xb804: push    rbp
0xb805: push    rbx
0xb806: mov     ebx, esi
0xb808: sub     rsp, 0x38
0xb80c: mov     rax, qword ptr fs:[0x28]
0xb815: mov     qword ptr [rsp + 0x28], rax
0xb81a: movabs  rax, 0x7363762f7665642f
0xb824: mov     word ptr [rsp + 8], di
0xb829: mov     byte ptr [rsp + 0xa], 0
0xb82e: mov     dword ptr [rsp + 0x18], 0x585861
0xb836: mov     qword ptr [rsp], rax
0xb83a: mov     qword ptr [rsp + 0x10], rax
0xb83f: cmp     esi, 1
0xb842: je      0xb978
0xb848: mov     esi, dword ptr [rip + 0x1fbb2]
0xb84e: mov     r13, qword ptr [rip + 0x1d783]
0xb855: mov     r14d, 1
0xb85b: test    esi, esi
0xb85d: jne     0xb90c
0xb863: mov     rax, qword ptr [rip + 0x1f126]
0xb86a: lea     r15, [rsp + 0x10]
0xb86f: mov     esi, r14d
0xb872: mov     dword ptr [rip + 0x1fb84], 0
0xb87c: mov     rdi, r15
0xb87f: movzx   edx, byte ptr [rax + 8]
0xb883: mov     byte ptr [rsp + 0x19], dl
0xb887: movzx   eax, byte ptr [rax + 9]
0xb88b: mov     byte ptr [rsp + 0x1a], al
0xb88f: call    0x4930
0xb894: mov     ebp, eax
0xb896: cmp     eax, -1
0xb899: je      0xb8f0
0xb89b: mov     eax, dword ptr [rip + 0x1f0fb]
0xb8a1: imul    eax, dword ptr [rip + 0x1f0f0]
0xb8a8: mov     edi, ebp
0xb8aa: mov     rsi, r12
0xb8ad: lea     edx, [rax + rax + 4]
0xb8b1: xor     eax, eax
0xb8b3: call    r13
0xb8b6: mov     edi, ebp
0xb8b8: call    0x4ab0

```

```

0xb8bd: cmp     ebx, 2
0xb8c0: je      0xb990
0xb8c6: mov     eax, 1
0xb8cb: mov     rdx, qword ptr [rsp + 0x28]
0xb8d0: sub     rdx, qword ptr fs:[0x28]
0xb8d9: jne     0xbb07
0xb8df: add     rsp, 0x38
0xb8e3: pop     rbx
0xb8e4: pop     rbp
0xb8e5: pop     r12
0xb8e7: pop     r13
0xb8e9: pop     r14
0xb8eb: pop     r15
0xb8ed: ret
0xb8ee: nop
0xb8f0: mov     ecx, 0x30
0xb8f5: mov     esi, r14d
0xb8f8: mov     rdi, r15
0xb8fb: mov     word ptr [rsp + 0x19], cx
0xb900: call    0x4930
0xb905: mov     ebp, eax
0xb907: cmp     eax, -1
0xb90a: jne     0xb89b
0xb90c: mov     dword ptr [rip + 0x1faea], 1
0xb916: cmp     ebx, 2
0xb919: je      0xba30
0xb91f: mov     rax, qword ptr [rip + 0x1f06a]
0xb926: mov     r15, rsp
0xb929: mov     esi, r14d
0xb92c: mov     rdi, r15
0xb92f: movzx   edx, byte ptr [rax + 8]
0xb933: movzx   eax, byte ptr [rax + 9]
0xb937: mov     byte ptr [rsp + 8], dl
0xb93b: mov     byte ptr [rsp + 9], al
0xb93f: call    0x4930
0xb944: mov     ebp, eax
0xb946: cmp     eax, -1
0xb949: je      0xba00
0xb94f: mov     edx, dword ptr [rip + 0x1f043]
0xb955: mov     edi, ebp
0xb957: mov     rsi, r12
0xb95a: xor     eax, eax
0xb95c: imul    edx, dword ptr [rip + 0x1f039]
0xb963: add     edx, 4
0xb966: call    r13
0xb969: mov     edi, ebp
0xb96b: call    0x4ab0
0xb970: jmp     0xb8c6

```

```

0xb8c6
0xb975: nop      dword ptr [rax]
0xb978: mov      r13, qword ptr [rip + 0x1d661]
0xb97f: xor      r14d, r14d
0xb982: jmp      0xb863
0xb863
0xb987: nop      word ptr [rax + rax]
0xb990: movsx    esi, byte ptr [r12 + 2]
0xb996: movsx    edi, byte ptr [r12 + 3]
0xb99c: call     0xa7a0
0xb9a1: mov      edx, dword ptr [rip + 0x1f561]
0xb9a7: test     edx, edx
0xb9a9: jle      0xb9e5
0xb9ab: xor      ebp, ebp
0xb9ad: lea      rbx, [rip + 0x1f56c]
0xb9b4: jmp      0xb9cc
0xb9cc
0xb9b6: nop      word ptr cs:[rax + rax]
0xb9c0: mov      edx, dword ptr [rip + 0x1f542]
0xb9c6: add      ebp, eax
0xb9c8: cmp      edx, ebp
0xb9ca: jle      0xb9e5
0xb9cc: sub      edx, ebp
0xb9ce: movsxd   rsi, ebp
0xb9d1: mov      edi, 1
0xb9d6: add      rsi, rbx
0xb9d9: movsxd   rdx, edx
0xb9dc: call     0x18da0
0xb9e1: test     eax, eax
0xb9e3: jns      0xb9c0
0xb9e5: mov      dword ptr [rip + 0x1f519], 0
0xb9ef: mov      eax, 1
0xb9f4: jmp      0xb8cb
0xb8cb
0xb9f9: nop      dword ptr [rax]
0xba00: mov      edx, 0x30
0xba05: mov      esi, r14d
0xba08: mov      rdi, r15
0xba0b: mov      word ptr [rsp + 8], dx
0xba10: call     0x4930
0xba15: mov      ebp, eax
0xba17: cmp      eax, -1
0xba1a: jne      0xb94f
0xba20: xor      eax, eax
0xba22: jmp      0xb8cb
0xb8cb
0xba27: nop      word ptr [rax + rax]
0xba30: xor      eax, eax

```

```

0xba32: mov     r15, rsp
0xba35: call    0xa300
0xba3a: mov     rax, qword ptr [rip + 0x1ef4f]
0xba41: mov     esi, r14d
0xba44: mov     rdi, r15
0xba47: movzx   edx, byte ptr [rax + 8]
0xba4b: movzx   eax, byte ptr [rax + 9]
0xba4f: mov     byte ptr [rsp + 8], dl
0xba53: mov     byte ptr [rsp + 9], al
0xba57: call    0x4930
0xba5c: mov     ebp, eax
0xba5e: cmp     eax, -1
0xba61: je      0xbae0
0xba63: mov     edx, dword ptr [rip + 0x1ef2f]
0xba69: mov     rsi, r12
0xba6c: mov     edi, ebp
0xba6e: xor     eax, eax
0xba70: imul    edx, dword ptr [rip + 0x1ef25]
0xba77: add     edx, 4
0xba7a: call    r13
0xba7d: mov     edi, ebp
0xba7f: call    0x4ab0
0xba84: mov     eax, dword ptr [rip + 0x1ef12]
0xba8a: xor     esi, esi
0xba8c: lea     edi, [rax - 1]
0xba8f: call    0xa7a0
0xba94: mov     edx, dword ptr [rip + 0x1f46e]
0xba9a: test    edx, edx
0xba9c: jle     0xb9e5
0xbaa2: xor     ebp, ebp
0xbaa4: lea     rbx, [rip + 0x1f475]
0xbaab: jmp     0xbac0
0xbac0:
0xbaad: nop     dword ptr [rax]
0xbab0: mov     edx, dword ptr [rip + 0x1f452]
0xbab6: add     ebp, eax
0xbab8: cmp     edx, ebp
0xbaba: jle     0xb9e5
0xbac0: sub     edx, ebp
0xbac2: movsxd  rsi, ebp
0xbac5: mov     edi, 1
0xbaca: add     rsi, rbx
0xbacd: movsxd  rdx, edx
0xbad0: call    0x18da0
0xbad5: test    eax, eax
0xbad7: jns     0xbab0
0xbad9: jmp     0xb9e5
0xb9e5:

```

```

0xbade: nop
0xbae0: mov     eax, 0x30
0xbae5: mov     esi, r14d
0xbae8: mov     rdi, r15
0xbaeb: mov     word ptr [rsp + 8], ax
0xbaf0: call    0x4930
0xbaf5: mov     ebp, eax
0xbaf7: cmp     eax, -1
0xbafa: jne     0xba63
0xbb00: xor     eax, eax
0xbb02: jmp     0xb8cb
0xb8cb
0xbb07: call    0x49d0
0xbb0c: nop     dword ptr [rax]
0xbb10: endbr64
0xbb14: mov     eax, dword ptr [rip + 0x1ee6e]
0xbb1a: test    eax, eax
0xbb1c: jne     0xbb20
0xbb1e: ret
0xbb1f: nop
0xbb20: sub     rsp, 8
0xbb24: mov     esi, 1
0xbb29: xor     eax, eax
0xbb2b: call    0xb7f0
0xbb30: mov     dword ptr [rip + 0x1f8ce], eax
0xbb36: add     rsp, 8
0xbb3a: ret
0xbb3b: nop     dword ptr [rax + rax]
0xbb40: endbr64
0xbb44: push    rbp
0xbb45: xor     eax, eax
0xbb47: mov     rbp, rdi
0xbb4a: call    0xa6c0
0xbb4f: mov     edx, dword ptr [rip + 0x1ee33]
0xbb55: test    edx, edx
0xbb57: je      0xbb94
0xbb59: mov     eax, dword ptr [rip + 0x1f8a5]
0xbb5f: test    eax, eax
0xbb61: je      0xbb94
0xbb63: mov     edx, dword ptr [rip + 0x1ee33]
0xbb69: imul    edx, dword ptr [rip + 0x1ee28]
0xbb70: xor     esi, esi
0xbb72: mov     rdi, qword ptr [rip + 0x1f7af]
0xbb79: movsxd  rdx, edx
0xbb7c: call    0x4a50
0xbb81: xor     eax, eax
0xbb83: mov     esi, 2
0xbb88: mov     rdi, rbp

```

```

0xbb8b: call    0xb7f0
0xbb90: test    eax, eax
0xbb92: jne     0xbba0
0xbb94: xor     eax, eax
0xbb96: pop     rbp
0xbb97: jmp     0xa300
0xa300:
0xbb9c: nop     dword ptr [rax]
0xbba0: mov     edx, dword ptr [rip + 0x1edf6]
0xbba6: imul    edx, dword ptr [rip + 0x1edeb]
0xbbad: xor     esi, esi
0xbbae: mov     rdi, qword ptr [rip + 0x1f782]
0xbbb6: pop     rbp
0xbbb7: movsxd  rdx, edx
0xbbba: jmp     0x4a50
0x4a50:
0xbbbf: nop
0xbbc0: endbr64
0xbbc4: push    rbp
0xbbc5: mov     rbp, rdi
0xbbc8: push    rbx
0xbbc9: sub     rsp, 8
0xbbcd: cmp     dword ptr [rip + 0x1ed84], 1
0xbbd4: je      0xbcd0
0xbbda: xor     eax, eax
0xbbdc: call    0xa6c0
0xbbe1: test    rbp, rbp
0xbbe4: je      0xbbf4
0xbbe6: mov     eax, dword ptr [rip + 0x1ed9c]
0xbbec: test    eax, eax
0xbbee: jne     0xbcc0
0xbbf4: mov     eax, dword ptr [rip + 0x1eda2]
0xbbfa: mov     rdi, qword ptr [rip + 0x1e347]
0xbc01: lea     rdx, [rip - 0x2198]
0xbc08: lea     esi, [rax - 1]
0xbc0b: call    0x4820
0xbc10: mov     edx, dword ptr [rip + 0x1f2f2]
0xbc16: test    edx, edx
0xbc18: jle     0xbc4d
0xbc1a: xor     ebp, ebp
0xbc1c: lea     rbx, [rip + 0x1f2fd]
0xbc23: jmp     0xbc34
0xbc34:
0xbc25: nop     dword ptr [rax]
0xbc28: mov     edx, dword ptr [rip + 0x1f2da]
0xbc2e: add     ebp, eax
0xbc30: cmp     edx, ebp
0xbc32: jle     0xbc4d

```



```

0xbc34: sub     edx, ebp
0xbc36: movsxd   rsi, ebp
0xbc39: mov      edi, 1
0xbc3e: add      rsi, rbx
0xbc41: movsxd   rdx, edx
0xbc44: call     0x18da0
0xbc49: test     eax, eax
0xbc4b: jns      0xbc28
0xbc4d: mov      edi, dword ptr [rip + 0x1ed49]
0xbc53: xor      esi, esi
0xbc55: mov      dword ptr [rip + 0x1f2a9], 0
0xbc5f: call     0xa7a0
0xbc64: mov      edx, dword ptr [rip + 0x1f29e]
0xbc6a: test     edx, edx
0xbc6c: jle      0xbca5
0xbc6e: xor      ebp, ebp
0xbc70: lea      rbx, [rip + 0x1f2a9]
0xbc77: jmp      0xbc8c
0xbc8c:
0xbc79: nop      dword ptr [rax]
0xbc80: mov      edx, dword ptr [rip + 0x1f282]
0xbc86: add      ebp, eax
0xbc88: cmp      edx, ebp
0xbc8a: jle      0xbca5
0xbc8c: sub      edx, ebp
0xbc8e: movsxd   rsi, ebp
0xbc91: mov      edi, 1
0xbc96: add      rsi, rbx
0xbc99: movsxd   rdx, edx
0xbc9c: call     0x18da0
0xbca1: test     eax, eax
0xbca3: jns      0xbc80
0xbca5: mov      dword ptr [rip + 0x1f259], 0
0xbcaf: add      rsp, 8
0xbcb3: mov      edi, 0xa
0xbcb8: pop      rbx
0xbcb9: pop      rbp
0xbcba: jmp      0x4810
0x4810:
0xbcbf: nop
0xbcc0: mov      rdi, rbp
0xbcc3: call     0xbb40
0xbcc8: jmp      0xbbf4
0xbbf4:
0xbccd: nop      dword ptr [rax]
0xbcd0: lea      rdx, [rip + 0x1f6e9]
0xbcd7: mov      esi, 1
0xbcdc: mov      edi, 1

```

```

0xbce1: call    0x4d70
0xbce6: mov     dword ptr [rip + 0x1ec68], 0
0xbcf0: jmp     0xbbda
0xbbda
0xbcf5: nop     word ptr cs:[rax + rax]
0xbd00: endbr64
0xbd04: push    r13
0xbd06: lea     r13, [rip + 0x1cc33]
0xbd0d: push    r12
0xbd0f: push    rbp
0xbd10: mov     rbp, rdi
0xbd13: lea     rdi, [rip + 0x14f14]
0xbd1a: push    rbx
0xbd1b: xor     ebx, ebx
0xbd1d: sub     rsp, 8
0xbd21: jmp     0xbd2d
0xbd2d
0xbd23: nop     dword ptr [rax + rax]
0xbd28: mov     rdi, qword ptr [r13 + rbx*8]
0xbd2d: mov     rsi, rbp
0xbd30: mov     r12d, ebx
0xbd33: call    0x4b30
0xbd38: test    eax, eax
0xbd3a: je      0xbd60
0xbd3c: add     rbx, 1
0xbd40: cmp     rbx, 0xa
0xbd44: jne     0xbd28
0xbd46: add     rsp, 8
0xbd4a: mov     r12d, 0xffffffff
0xbd50: pop     rbx
0xbd51: mov     eax, r12d
0xbd54: pop     rbp
0xbd55: pop     r12
0xbd57: pop     r13
0xbd59: ret
0xbd5a: nop     word ptr [rax + rax]
0xbd60: lea     eax, [rbx - 8]
0xbd63: cmp     ebx, 7
0xbd66: cmovg   r12d, eax
0xbd6a: add     rsp, 8
0xbd6e: pop     rbx
0xbd6f: pop     rbp
0xbd70: mov     eax, r12d
0xbd73: pop     r12
0xbd75: pop     r13
0xbd77: ret
0xbd78: nop     dword ptr [rax + rax]
0xbd80: endbr64

```

```

0xbd84: push    r12
0xbd86: mov     r12, rdi
0xbd89: push    rbp
0xbd8a: lea     rbp, [rip + 0x1e1e7]
0xbd91: push    rbx
0xbd92: mov     ebx, 0x11
0xbd97: jmp     0xbdac
0xbdac:
0xbd99: nop     dword ptr [rax]
0xbda0: add     ebx, 1
0xbda3: add     rbp, 0x20
0xbda7: cmp     ebx, 0x26
0xbdaa: je      0xbdd8
0xbdac: mov     rdi, qword ptr [rbp]
0xbdb0: mov     rsi, r12
0xbdb3: call    0x4b30
0xbdb8: test    eax, eax
0xbdba: jne     0xbda0
0xbdbc: movsxd  rbx, ebx
0xbdbf: lea     rax, [rip + 0x1df7a]
0xbdc6: shl     rbx, 5
0xbdc9: mov     rax, qword ptr [rax + rbx + 8]
0xbdcf: pop     rbx
0xbdd0: pop     rbp
0xbdd1: pop     r12
0xbdd3: ret
0xbdd4: nop     dword ptr [rax]
0xbdd8: pop     rbx
0xbdd9: xor     eax, eax
0xbddb: pop     rbp
0xbddc: pop     r12
0xbdde: ret
0xbddf: nop
0xbde0: endbr64
0xbde4: push    r14
0xbde6: lea     rdi, [rip + 0x1479d]
0xbded: push    r13
0xbdef: push    r12
0xbdf1: push    rbp
0xbdf2: push    rbx
0xbdf3: sub     rsp, 0xb0
0xbdfa: mov     rax, qword ptr fs:[0x28]
0xbe03: mov     qword ptr [rsp + 0xa8], rax
0xbe0b: xor     eax, eax
0xbe0d: call    0x47e0
0xbe12: lea     rsi, [rsp + 0x10]
0xbe17: mov     edi, 1
0xbe1c: mov     rbp, rax

```

```

0xbe1f: call    0x4eb0
0xbe24: mov     rax, qword ptr [rsp + 0x38]
0xbe29: shr     rax, 8
0xbe2d: cmp     rax, 4
0xbe31: sete    al
0xbe34: movzx   eax, al
0xbe37: mov     dword ptr [rip + 0x1eb4b], eax
0xbe3d: test    rbp, rbp
0xbe40: je      0xc36d
0xbe46: mov     rdi, rbp
0xbe49: call    0x49b0
0xbe4e: cmp     rax, 0x3f
0xbe52: ja      0xc308
0xbe58: lea     r12, [rip + 0x14ff1]
0xbe5f: lea     rbx, [rip + 0x14d6f]
0xbe66: lea     r14, [rip + 0x14e02]
0xbe6d: lea     r13, [rip + 0x14e07]
0xbe74: nop     dword ptr [rax]
0xbe78: xor     edi, edi
0xbe7a: mov     rsi, rbp
0xbe7d: call    0x4e90
0xbe82: cmp     eax, -1
0xbe85: je      0xbf38
0xbe8b: test    eax, eax
0xbe8d: jne     0xbff8
0xbe93: mov     rdi, qword ptr [rip + 0x1e9e6]
0xbe9a: mov     esi, 1
0xbe9f: mov     r8, rbp
0xbea2: mov     rdx, r12
0xbea5: mov     rcx, qword ptr [rip + 0x1ea3c]
0xbeac: lea     r9, [rip + 0x1de5d]
0xeb3: call    0x4e60
0xeb8: mov     rsi, rbx
0xebb: mov     rdi, rbp
0xeb3e: call    0x4b30
0xebc3: test    eax, eax
0xebc5: je      0xbf98
0xebcb: mov     rdi, rbp
0xebce: call    0x9940
0xbed3: test    eax, eax
0xbed5: jne     0xbfc8
0xbedb: mov     rsi, r14
0xbede: mov     rdi, rbp
0xbee1: call    0x4b30
0xbee6: test    eax, eax
0xbee8: je      0xbf01
0xbeea: lea     rsi, [rip + 0x14d84]
0xbef1: mov     rdi, rbp

```

```

0xbef4: call    0x4b30
0xbef9: test    eax, eax
0xbefb: jne     0xc3b4
0xbf01: mov     rdi, qword ptr [rip + 0x1e978]
0xbf08: mov     rcx, qword ptr [rip + 0x1e9d9]
0xbf0f: mov     esi, 1
0xbf14: xor     eax, eax
0xbf16: mov     rdx, r13
0xbf19: lea     rbp, [rip + 0x14cba]
0xbf20: call    0x4e60
0xbf25: xor     edi, edi
0xbf27: mov     rsi, rbp
0xbf2a: call    0x4e90
0xbf2f: cmp     eax, -1
0xbf32: jne     0xbe8b
0xbf38: mov     rcx, qword ptr [rip + 0x1e9a9]
0xbf3f: mov     rdi, qword ptr [rip + 0x1e93a]
0xbf46: xor     eax, eax
0xbf48: lea     r8, [rip + 0x1ddc1]
0xbf4f: lea     rdx, [rip + 0x14ea2]
0xbf56: mov     esi, 1
0xbf5b: call    0x4e60
0xbf60: mov     rdi, qword ptr [rip + 0x1e919]
0xbf67: xor     eax, eax
0xbf69: mov     rcx, qword ptr [rip + 0x1e978]
0xbf70: lea     r8, [rip + 0x1dd89]
0xbf77: lea     rdx, [rip + 0x14ea2]
0xbf7e: mov     esi, 1
0xbf83: call    0x4e60
0xbf88: mov     edi, 1
0xbf8d: call    0x4e20
0xbf92: nop     word ptr [rax + rax]
0xbf98: mov     rcx, qword ptr [rip + 0x1e949]
0xbf9f: mov     rdi, qword ptr [rip + 0x1e8da]
0xbfa6: lea     rdx, [rip + 0x14c99]
0xbfad: mov     esi, 1
0xbfb2: lea     rbp, [rip + 0x14c21]
0xbfb9: call    0x4e60
0xbfbe: jmp     0xbe78
0xbe78:
0xbfc3: nop     dword ptr [rax + rax]
0xbfc8: mov     rcx, qword ptr [rip + 0x1e919]
0xbfcf: mov     rdi, qword ptr [rip + 0x1e8aa]
0xbfd6: lea     rdx, [rip + 0x14c7d]
0xbfdd: xor     eax, eax
0xbfdf: mov     esi, 1
0xbfe4: lea     rbp, [rip + 0x14bf6]
0xbfeb: call    0x4e60

```

```

0xbff0: jmp      0xbe78
0xbe78:
0xbff5: nop      dword ptr [rax]
0xbff8: mov      rdi, rbp
0xbffb: lea      r13, [rsp + 8]
0xc000: call     0x1d7a0
0xc005: mov      edi, 0x800
0xc00a: mov      qword ptr [rip + 0x1e93f], rax
0xc011: call     0x1d670
0xc016: mov      rdi, qword ptr [rip + 0x1de03]
0xc01d: mov      rsi, r13
0xc020: mov      qword ptr [rsp + 8], rax
0xc025: call     0x4ca0
0xc02a: xor      edx, edx
0xc02c: test     rax, rax
0xc02f: je       0xc034
0xc031: movzx    edx, byte ptr [rax]
0xc034: mov      rdi, qword ptr [rip + 0x1de25]
0xc03b: mov      rsi, r13
0xc03e: mov      byte ptr [rip + 0x1e904], dl
0xc044: call     0x4ca0
0xc049: mov      rdi, qword ptr [rip + 0x1ddf0]
0xc050: mov      rsi, r13
0xc053: mov      qword ptr [rip + 0x1e8e6], rax
0xc05a: call     0x4ca0
0xc05f: cmp      qword ptr [rip + 0x1e8d9], 0
0xc067: mov      qword ptr [rip + 0x1e8ca], rax
0xc06e: je       0xc2c0
0xc074: test     rax, rax
0xc077: je       0xc2c0
0xc07d: mov      rdi, qword ptr [rip + 0x1dcbc]
0xc084: mov      rsi, r13
0xc087: call     0x4ca0
0xc08c: mov      rdi, qword ptr [rip + 0x1dced]
0xc093: mov      rsi, r13
0xc096: mov      qword ptr [rip + 0x1dcab], rax
0xc09d: call     0x4ca0
0xc0a2: mov      rdi, qword ptr [rip + 0x1dcb7]
0xc0a9: mov      rsi, r13
0xc0ac: mov      qword ptr [rip + 0x1dcd5], rax
0xc0b3: call     0x4ca0
0xc0b8: cmp      qword ptr [rip + 0x1dc88], 0
0xc0c0: mov      qword ptr [rip + 0x1dca1], rax
0xc0c7: je       0xc352
0xc0cd: mov      rdi, qword ptr [rip + 0x1ddac]
0xc0d4: mov      rsi, r13
0xc0d7: call     0x4ca0
0xc0dc: xor      edx, edx

```

```

0xc0de: mov     qword ptr [rip + 0x1dda3], rax
0xc0e5: test    rax, rax
0xc0e8: je      0xc0ff
0xc0ea: mov     rdi, qword ptr [rip + 0x1ddcf]
0xc0f1: mov     qword ptr [rip + 0x1dc70], rax
0xc0f8: call    0x4c90
0xc0fd: mov     edx, eax
0xc0ff: mov     rdi, qword ptr [rip + 0x1dd9a]
0xc106: mov     dword ptr [rip + 0x1ddc4], edx
0xc10c: call    0x4c50
0xc111: mov     dword ptr [rip + 0x1dd99], eax
0xc117: test    eax, eax
0xc119: js      0xc13c
0xc11b: mov     qword ptr [rip + 0x1dc62], 0
0xc126: mov     qword ptr [rip + 0x1dc37], 0
0xc131: mov     qword ptr [rip + 0x1dc0c], 0
0xc13c: mov     rdi, qword ptr [rip + 0x1dd9d]
0xc143: call    0x4c50
0xc148: mov     rdi, qword ptr [rip + 0x1ddb1]
0xc14f: mov     dword ptr [rip + 0x1dd9b], eax
0xc155: call    0x4c50
0xc15a: mov     rdi, qword ptr [rip + 0x1dc3f]
0xc161: mov     rsi, r13
0xc164: mov     dword ptr [rip + 0x1dda6], eax
0xc16a: call    0x4ca0
0xc16f: mov     rdi, qword ptr [rip + 0x1dc4a]
0xc176: mov     rsi, r13
0xc179: mov     qword ptr [rip + 0x1dc28], rax
0xc180: call    0x4ca0
0xc185: cmp     qword ptr [rip + 0x1dc1b], 0
0xc18d: mov     qword ptr [rip + 0x1dc34], rax
0xc194: je      0xc2a0
0xc19a: test    rax, rax
0xc19d: je      0xc2a0
0xc1a3: mov     rdi, qword ptr [rip + 0x1dc36]
0xc1aa: mov     rsi, r13
0xc1ad: lea     rbp, [rip + 0x1ddac]
0xc1b4: lea     rbx, [rbp - 0x220]
0xc1bb: call    0x4ca0
0xc1c0: mov     rdi, qword ptr [rip + 0x1dc39]
0xc1c7: mov     rsi, r13
0xc1ca: lea     r12, [rbx + 0x4c0]
0xc1d1: mov     qword ptr [rip + 0x1dc10], rax
0xc1d8: call    0x4ca0
0xc1dd: mov     rdi, qword ptr [rip + 0x1dd3c]
0xc1e4: mov     rsi, r13
0xc1e7: mov     qword ptr [rip + 0x1dc1a], rax
0xc1ee: call    0x4ca0

```

```

0xc1f3: mov     rdi, qword ptr [rip + 0x1dd46]
0xc1fa: mov     rsi, r13
0xc1fd: mov     qword ptr [rip + 0x1dd24], rax
0xc204: call    0x4ca0
0xc209: mov     qword ptr [rip + 0x1dd38], rax
0xc210: mov     rdi, qword ptr [rbp]
0xc214: mov     rsi, r13
0xc217: add     rbp, 0x20
0xc21b: call    0x4ca0
0xc220: mov     qword ptr [rbp - 0x18], rax
0xc224: cmp     rbp, r12
0xc227: jne     0xc210
0xc229: xor     ebp, ebp
0xc22b: lea     r13, [rip + 0x161b3]
0xc232: lea     r14, [rip + 0x14c57]
0xc239: jmp     0xc249
0xc249
0xc23b: nop     dword ptr [rax + rax]
0xc240: add     rbx, 0x20
0xc244: cmp     rbx, r12
0xc247: je      0xc268
0xc249: cmp     qword ptr [rbx + 8], 0
0xc24e: jne     0xc240
0xc250: mov     eax, dword ptr [rbx + 0x14]
0xc253: test    eax, eax
0xc255: jne     0xc2e0
0xc25b: mov     qword ptr [rbx + 8], r13
0xc25f: add     rbx, 0x20
0xc263: cmp     rbx, r12
0xc266: jne     0xc249
0xc268: test    ebp, ebp
0xc26a: jne     0xc390
0xc270: mov     rax, qword ptr [rsp + 0xa8]
0xc278: sub     rax, qword ptr fs:[0x28]
0xc281: jne     0xc3be
0xc287: add     rsp, 0xb0
0xc28e: pop     rbx
0xc28f: pop     rbp
0xc290: pop     r12
0xc292: pop     r13
0xc294: pop     r14
0xc296: ret
0xc297: nop     word ptr [rax + rax]
0xc2a0: mov     qword ptr [rip + 0x1dafd], 0
0xc2ab: mov     qword ptr [rip + 0x1db12], 0
0xc2b6: jmp     0xc1a3
0xc1a3
0xc2bb: nop     dword ptr [rax + rax]

```



```

0xc2c0: mov     qword ptr [rip + 0x1e66d], 0
0xc2cb: mov     qword ptr [rip + 0x1e66a], 0
0xc2d6: jmp     0xc07d
0xc07d:
0xc2db: nop     dword ptr [rax + rax]
0xc2e0: mov     r8, qword ptr [rbx]
0xc2e3: mov     rcx, qword ptr [rip + 0x1e5fe]
0xc2ea: mov     rdx, r14
0xc2ed: xor     eax, eax
0xc2ef: mov     rdi, qword ptr [rip + 0x1e58a]
0xc2f6: mov     esi, 1
0xc2fb: add     ebp, 1
0xc2fe: call    0x4e60
0xc303: jmp     0xc240
0xc240:
0xc308: mov     rcx, qword ptr [rip + 0x1e5d9]
0xc30f: mov     rdi, qword ptr [rip + 0x1e56a]
0xc316: lea     rdx, [rip + 0x14aab]
0xc31d: xor     eax, eax
0xc31f: mov     esi, 1
0xc324: call    0x4e60
0xc329: mov     rcx, qword ptr [rip + 0x1e550]
0xc330: mov     edx, 0x11
0xc335: mov     esi, 1
0xc33a: lea     rdi, [rip + 0x148f3]
0xc341: lea     rbp, [rip + 0x1d9d1]
0xc348: call    0x4e40
0xc34d: jmp     0xbe58
0xbe58:
0xc352: mov     qword ptr [rip + 0x1da2b], 0
0xc35d: mov     qword ptr [rip + 0x1da00], 0
0xc368: jmp     0xc0cd
0xc0cd:
0xc36d: mov     rcx, qword ptr [rip + 0x1e574]
0xc374: mov     rdi, qword ptr [rip + 0x1e505]
0xc37b: lea     rdx, [rip + 0x14a16]
0xc382: xor     eax, eax
0xc384: mov     esi, 1
0xc389: call    0x4e60
0xc38e: jmp     0xc329
0xc329:
0xc390: mov     rcx, qword ptr [rip + 0x1e551]
0xc397: mov     rdi, qword ptr [rip + 0x1e4e2]
0xc39e: mov     r8d, ebp
0xc3a1: xor     eax, eax
0xc3a3: lea     rdx, [rip + 0x14b16]
0xc3aa: mov     esi, 1
0xc3af: call    0x4e60

```

```

0xc3b4: mov     edi, 1
0xc3b9: call    0x4e20
0xc3be: call    0x49d0
0xc3c3: nop     word ptr cs:[rax + rax]
0xc3ce: nop
0xc3d0: endbr64
0xc3d4: push    rbx
0xc3d5: mov     ebx, edi
0xc3d7: xor     edi, edi
0xc3d9: call    0x48f0
0xc3de: test    eax, eax
0xc3e0: je      0xc453
0xc3e2: mov     edi, 1
0xc3e7: call    0x48f0
0xc3ec: test    eax, eax
0xc3ee: je      0xc453
0xc3f0: mov     edi, 1
0xc3f5: call    0x4cd0
0xc3fa: mov     qword ptr [rip + 0x1e58f], rax
0xc401: test    rax, rax
0xc404: je      0xc47e
0xc406: lea     rsi, [rip + 0x1efb3]
0xc40d: mov     edi, 1
0xc412: call    0x4d60
0xc417: mov     edi, 0x40
0xc41c: call    0x1d670
0xc421: mov     dword ptr [rip + 0x1efd5], ebx
0xc427: pop     rbx
0xc428: mov     qword ptr [rip + 0x1ef31], rax
0xc42f: mov     qword ptr [rip + 0x1e52a], rax
0xc436: xor     eax, eax
0xc438: mov     qword ptr [rip + 0x1e52d], 0
0xc443: mov     qword ptr [rip + 0x1e51a], 0
0xc44e: jmp     0xbde0
0xbde0
0xc453: mov     rdi, qword ptr [rip + 0x1e426]
0xc45a: mov     rcx, qword ptr [rip + 0x1e487]
0xc461: lea     rdx, [rip + 0x14a88]
0xc468: xor     eax, eax
0xc46a: mov     esi, 1
0xc46f: call    0x4e60
0xc474: mov     edi, 1
0xc479: call    0x4e20
0xc47e: mov     rdi, qword ptr [rip + 0x1e3fb]
0xc485: mov     rcx, qword ptr [rip + 0x1e45c]
0xc48c: mov     esi, 1
0xc491: lea     rdx, [rip + 0x147f8]
0xc498: call    0x4e60

```

```

0xc49d: mov     edi, 1
0xc4a2: call    0x4e20
0xc4a7: nop     word ptr [rax + rax]
0xc4b0: endbr64
0xc4b4: push    r12
0xc4b6: push    rbp
0xc4b7: mov     rbp, rdi
0xc4ba: mov     rdi, qword ptr [rip + 0x1e48f]
0xc4c1: push    rbx
0xc4c2: call    0x9940
0xc4c7: test    eax, eax
0xc4c9: jne     0xc4d0
0xc4cb: pop     rbx
0xc4cc: pop     rbp
0xc4cd: pop     r12
0xc4cf: ret
0xc4d0: mov     rdi, rbp
0xc4d3: call    0x49b0
0xc4d8: lea     rdi, [rax + 0x81]
0xc4df: mov     rbx, rax
0xc4e2: call    0x1d670
0xc4e7: mov     rdi, rbp
0xc4ea: mov     r12, rax
0xc4ed: call    0x1d7a0
0xc4f2: mov     rsi, rbx
0xc4f5: add     ebx, 0x7f
0xc4f8: mov     rbp, rax
0xc4fb: mov     rdi, rax
0xc4fe: call    0x184d0
0xc503: push    7
0xc505: lea     rcx, [rip + 0x147a2]
0xc50c: xor     eax, eax
0xc50e: push    rbp
0xc50f: mov     rdx, -1
0xc516: mov     esi, 1
0xc51b: mov     rdi, r12
0xc51e: lea     r9, [rip + 0x14040]
0xc525: mov     r8d, 0x1b
0xc52b: call    0x4f10
0xc530: mov     eax, dword ptr [rip + 0x1e462]
0xc536: pop     rdx
0xc537: pop     rcx
0xc538: add     eax, eax
0xc53a: cmp     ebx, eax
0xc53c: jl      0xc555
0xc53e: cdqe
0xc540: mov     byte ptr [r12 + rax], 7
0xc545: mov     eax, dword ptr [rip + 0x1e44d]

```

```

0xc54b: add     eax, eax
0xc54d: cdqe
0xc54f: mov     byte ptr [r12 + rax + 1], 0
0xc555: mov     rdi, r12
0xc558: call    0x49b0
0xc55d: mov     rsi, r12
0xc560: mov     edi, 1
0xc565: mov     rdx, rax
0xc568: call    0x18da0
0xc56d: mov     rdi, rbp
0xc570: call    0x18d10
0xc575: mov     rdi, r12
0xc578: call    0x18d10
0xc57d: mov     rdi, qword ptr [rip + 0x1e2bc]
0xc584: pop     rbx
0xc585: pop     rbp
0xc586: pop     r12
0xc588: jmp     0x4c40
0x4c40:
0xc58d: nop     dword ptr [rax]
0xc590: endbr64
0xc594: sub     rsp, 8
0xc598: mov     edi, 0x18
0xc59d: call    0x1d670
0xc5a2: mov     qword ptr [rax], 0
0xc5a9: mov     qword ptr [rax + 8], 0
0xc5b1: add     rsp, 8
0xc5b5: ret
0xc5b6: nop     word ptr cs:[rax + rax]
0xc5c0: endbr64
0xc5c4: test    rdi, rdi
0xc5c7: je      0xc5d0
0xc5c9: jmp     0x18d10
0x18d10:
0xc5ce: nop
0xc5d0: ret
0xc5d1: nop     word ptr cs:[rax + rax]
0xc5dc: nop     dword ptr [rax]
0xc5e0: endbr64
0xc5e4: mov     dword ptr [rdi], esi
0xc5e6: mov     dword ptr [rdi + 4], edx
0xc5e9: mov     dword ptr [rdi + 8], ecx
0xc5ec: mov     dword ptr [rdi + 0xc], r8d
0xc5f0: ret
0xc5f1: nop     word ptr cs:[rax + rax]
0xc5fc: nop     dword ptr [rax]
0xc600: endbr64
0xc604: mov     rax, rdi

```

```

0xc607: mov     rdi, rsi
0xc60a: mov     esi, edx
0xc60c: mov     ecx, dword ptr [rax + 0x10]
0xc60f: lea     r8d, [rcx + rdx]
0xc613: mov     edx, dword ptr [rax + 0xc]
0xc616: mov     dword ptr [rax + 0x10], r8d
0xc61a: cmp     edx, ecx
0xc61c: jle     0xc640
0xc61e: mov     r9d, dword ptr [rax + 8]
0xc622: cmp     dword ptr [rax + 0x14], r9d
0xc626: jge     0xc640
0xc628: cmp     r8d, edx
0xc62b: jle     0xc631
0xc62d: sub     edx, ecx
0xc62f: mov     esi, edx
0xc631: jmp     0xa1a0
0xa1a0
0xc636: nop     word ptr cs:[rax + rax]
0xc640: xor     eax, eax
0xc642: ret
0xc643: nop     word ptr cs:[rax + rax]
0xc64e: nop
0xc650: endbr64
0xc654: mov     ecx, dword ptr [rdi + 0x10]
0xc657: lea     edx, [rcx + 1]
0xc65a: mov     dword ptr [rdi + 0x10], edx
0xc65d: cmp     edx, dword ptr [rdi + 0xc]
0xc660: jg      0xc66a
0xc662: mov     ecx, dword ptr [rdi + 8]
0xc665: cmp     dword ptr [rdi + 0x14], ecx
0xc668: jl      0xc670
0xc66a: xor     eax, eax
0xc66c: ret
0xc66d: nop     dword ptr [rax]
0xc670: mov     edi, esi
0xc672: jmp     0xa230
0xa230
0xc677: nop     word ptr [rax + rax]
0xc680: endbr64
0xc684: mov     rax, rdi
0xc687: mov     edi, esi
0xc689: mov     dword ptr [rax + 0x10], edx
0xc68c: add     edx, dword ptr [rax]
0xc68e: mov     dword ptr [rax + 0x14], esi
0xc691: add     edi, dword ptr [rax + 4]
0xc694: mov     esi, edx
0xc696: jmp     0xa4b0
0xa4b0

```

```

0xc69b: nop      dword ptr [rax + rax]
0xc6a0: endbr64
0xc6a4: mov      eax, dword ptr [rdi]
0xc6a6: ret
0xc6a7: nop      word ptr [rax + rax]
0xc6b0: endbr64
0xc6b4: mov      eax, dword ptr [rdi + 4]
0xc6b7: ret
0xc6b8: nop      dword ptr [rax + rax]
0xc6c0: endbr64
0xc6c4: mov      eax, dword ptr [rdi + 8]
0xc6c7: ret
0xc6c8: nop      dword ptr [rax + rax]
0xc6d0: endbr64
0xc6d4: mov      eax, dword ptr [rdi + 0xc]
0xc6d7: ret
0xc6d8: nop      dword ptr [rax + rax]
0xc6e0: push     r14
0xc6e2: push     r13
0xc6e4: push     r12
0xc6e6: mov      r12, qword ptr [rip + 0x1ed1b]
0xc6ed: push     rbp
0xc6ee: push     rbx
0xc6ef: mov      r13, qword ptr [r12 + 0x20]
0xc6f4: mov      rbx, qword ptr [r12 + 0x18]
0xc6f9: cmp      r13, rbx
0xc6fc: je       0xc761
0xc6fe: mov      ebp, edi
0xc700: jbe      0xc770
0xc702: test     bpl, 1
0xc706: je       0xc75c
0xc708: mov      rdi, qword ptr [r12 + 0x60]
0xc70d: test     rdi, rdi
0xc710: je       0xc71e
0xc712: call     0x18d10
0xc717: mov      r12, qword ptr [rip + 0x1ecea]
0xc71e: mov      r14, r13
0xc721: sub      r14, rbx
0xc724: lea      rdi, [r14 + 1]
0xc728: call     0x1d670
0xc72d: mov      rdx, r14
0xc730: mov      qword ptr [r12 + 0x60], rax
0xc735: mov      rax, qword ptr [rip + 0x1eccc]
0xc73c: mov      rsi, qword ptr [rax + 0x58]
0xc740: mov      rdi, qword ptr [rax + 0x60]
0xc744: add      rsi, rbx
0xc747: call     0x4bb0
0xc74c: mov      rax, qword ptr [rip + 0x1ecb5]

```

```

0xc753: mov     rax, qword ptr [rax + 0x60]
0xc757: mov     byte ptr [rax + r14], 0
0xc75c: and     ebp, 2
0xc75f: jne     0xc788
0xc761: pop     rbx
0xc762: pop     rbp
0xc763: pop     r12
0xc765: pop     r13
0xc767: pop     r14
0xc769: ret
0xc76a: nop     word ptr [rax + rax]
0xc770: mov     rax, rbx
0xc773: mov     rbx, r13
0xc776: mov     r13, rax
0xc779: test    bpl, 1
0xc77d: je      0xc75c
0xc77f: jmp     0xc708
0xc708:
0xc781: nop     dword ptr [rax]
0xc788: mov     rax, qword ptr [rip + 0x1ec79]
0xc78f: mov     rdx, rbx
0xc792: sub     rdx, r13
0xc795: add     qword ptr [rax + 0x48], rdx
0xc799: mov     rdi, qword ptr [rax + 0x58]
0xc79d: add     qword ptr [rax + 0x38], rdx
0xc7a1: mov     qword ptr [rax + 0x18], rbx
0xc7a5: lea     rsi, [rdi + r13]
0xc7a9: add     rdi, rbx
0xc7ac: mov     qword ptr [rax + 0x20], rbx
0xc7b0: call    0x48b0
0xc7b5: mov     rbx, qword ptr [rip + 0x1ec4c]
0xc7bc: mov     rax, qword ptr [rbx + 0x38]
0xc7c0: mov     rdi, qword ptr [rbx + 0x58]
0xc7c4: lea     rsi, [rax + 1]
0xc7c8: mov     qword ptr [rbx + 0x50], rsi
0xc7cc: call    0x1d690
0xc7d1: mov     qword ptr [rbx + 0x58], rax
0xc7d5: pop     rbx
0xc7d6: pop     rbp
0xc7d7: pop     r12
0xc7d9: pop     r13
0xc7db: pop     r14
0xc7dd: ret
0xc7de: nop
0xc7e0: endbr64
0xc7e4: test    rdi, rdi
0xc7e7: je      0xc818
0xc7e9: push    rbp

```

```

0xc7ea: mov     rbp, rdi
0xc7ed: mov     rdi, qword ptr [rdi + 0x58]
0xc7f1: test    rdi, rdi
0xc7f4: je      0xc7fb
0xc7f6: call    0x18d10
0xc7fb: mov     rdi, qword ptr [rbp + 0x60]
0xc7ff: test    rdi, rdi
0xc802: je      0xc809
0xc804: call    0x18d10
0xc809: mov     rdi, rbp
0xc80c: pop     rbp
0xc80d: jmp     0x18d10
0x18d10
0xc812: nop     word ptr [rax + rax]
0xc818: ret
0xc819: nop     dword ptr [rax]
0xc820: endbr64
0xc824: mov     rax, qword ptr [rip + 0x1ebdd]
0xc82b: push    rbp
0xc82c: mov     rdi, qword ptr [rax + 0x68]
0xc830: xor     eax, eax
0xc832: call    0x1aac0
0xc837: mov     rax, qword ptr [rip + 0x1ebca]
0xc83e: mov     rdi, qword ptr [rax]
0xc841: call    0xc5c0
0xc846: mov     rbp, qword ptr [rip + 0x1ebbb]
0xc84d: test    rbp, rbp
0xc850: je      0xc876
0xc852: mov     rdi, qword ptr [rbp + 0x58]
0xc856: test    rdi, rdi
0xc859: je      0xc860
0xc85b: call    0x18d10
0xc860: mov     rdi, qword ptr [rbp + 0x60]
0xc864: test    rdi, rdi
0xc867: je      0xc86e
0xc869: call    0x18d10
0xc86e: mov     rdi, rbp
0xc871: call    0x18d10
0xc876: mov     qword ptr [rip + 0x1eb87], 0
0xc881: pop     rbp
0xc882: ret
0xc883: nop     word ptr cs:[rax + rax]
0xc88e: nop
0xc890: endbr64
0xc894: mov     rax, qword ptr [rip + 0x1eb6d]
0xc89b: movsxd  rcx, edi
0xc89e: mov     edx, esi
0xc8a0: mov     r8, rcx

```



```

0xc8a3: mov     qword ptr [rax + 0x28], rcx
0xc8a7: movsxd   rcx, esi
0xc8aa: mov     rdi, qword ptr [rax]
0xc8ad: xor     esi, esi
0xc8af: mov     qword ptr [rax + 0x30], rcx
0xc8b3: mov     ecx, 1
0xc8b8: jmp     0xc5e0
0xc5e0
0xc8bd: nop     dword ptr [rax]
0xc8c0: endbr64
0xc8c4: push    r12
0xc8c6: mov     edi, 0x70
0xc8cb: push    rbx
0xc8cc: sub     rsp, 8
0xc8d0: call    0x1d670
0xc8d5: mov     rbx, qword ptr [rip + 0x1eb2c]
0xc8dc: mov     r12, rax
0xc8df: movdqu  xmm0, xmmword ptr [rbx]
0xc8e3: movups  xmmword ptr [rax], xmm0
0xc8e6: movdqu  xmm1, xmmword ptr [rbx + 0x10]
0xc8eb: movups  xmmword ptr [rax + 0x10], xmm1
0xc8ef: movdqu  xmm2, xmmword ptr [rbx + 0x20]
0xc8f4: movups  xmmword ptr [rax + 0x20], xmm2
0xc8f8: movdqu  xmm3, xmmword ptr [rbx + 0x30]
0xc8fd: movups  xmmword ptr [rax + 0x30], xmm3
0xc901: movdqu  xmm4, xmmword ptr [rbx + 0x40]
0xc906: movups  xmmword ptr [rax + 0x40], xmm4
0xc90a: movdqu  xmm5, xmmword ptr [rbx + 0x50]
0xc90f: movups  xmmword ptr [rax + 0x50], xmm5
0xc913: movdqu  xmm6, xmmword ptr [rbx + 0x60]
0xc918: cmp     qword ptr [rax + 0x58], 0
0xc91d: movups  xmmword ptr [rax + 0x60], xmm6
0xc921: je      0xc94a
0xc923: mov     rdi, qword ptr [rax + 0x50]
0xc927: call    0x1d670
0xc92c: mov     rdx, qword ptr [r12 + 0x50]
0xc931: mov     qword ptr [rbx + 0x58], rax
0xc935: mov     rax, qword ptr [rip + 0x1eacc]
0xc93c: mov     rsi, qword ptr [r12 + 0x58]
0xc941: mov     rdi, qword ptr [rax + 0x58]
0xc945: call    0x4bb0
0xc94a: mov     rdi, qword ptr [r12 + 0x60]
0xc94f: test    rdi, rdi
0xc952: je      0xc964
0xc954: mov     rbx, qword ptr [rip + 0x1eaad]
0xc95b: call    0x1d7a0
0xc960: mov     qword ptr [rbx + 0x60], rax
0xc964: add     rsp, 8

```

```

0xc968: mov     rax, r12
0xc96b: pop     rbx
0xc96c: pop     r12
0xc96e: ret
0xc96f: nop
0xc970: endbr64
0xc974: test    rdi, rdi
0xc977: je      0xc9f0
0xc979: push    r13
0xc97b: push    r12
0xc97d: push    rbp
0xc97e: push    rbx
0xc97f: mov     rbx, rdi
0xc982: sub     rsp, 8
0xc986: mov     rbp, qword ptr [rip + 0x1ea7b]
0xc98d: mov     rdi, qword ptr [rbp + 0x58]
0xc991: mov     r13, qword ptr [rbp + 0x28]
0xc995: mov     r12, qword ptr [rbp + 0x30]
0xc999: test    rdi, rdi
0xc99c: je      0xc9a3
0xc99e: call    0x18d10
0xc9a3: mov     rdi, qword ptr [rbp + 0x60]
0xc9a7: test    rdi, rdi
0xc9aa: je      0xc9b1
0xc9ac: call    0x18d10
0xc9b1: mov     rdi, rbp
0xc9b4: call    0x18d10
0xc9b9: movsxd  rax, r13d
0xc9bc: mov     edx, r12d
0xc9bf: movsxd  r12, r12d
0xc9c2: mov     qword ptr [rbx + 0x28], rax
0xc9c6: mov     rdi, qword ptr [rbx]
0xc9c9: mov     r8d, r13d
0xc9cc: mov     ecx, 1
0xc9d1: mov     qword ptr [rbx + 0x30], r12
0xc9d5: xor     esi, esi
0xc9d7: mov     qword ptr [rip + 0x1ea2a], rbx
0xc9de: add     rsp, 8
0xc9e2: pop     rbx
0xc9e3: pop     rbp
0xc9e4: pop     r12
0xc9e6: pop     r13
0xc9e8: jmp     0xc5e0
0xc5e0
0xc9ed: nop     dword ptr [rax]
0xc9f0: ret
0xc9f1: nop     word ptr cs:[rax + rax]
0xc9fc: nop     dword ptr [rax]

```

```

0xca00: endbr64
0xca04: mov     rdx, qword ptr [rip + 0x1e9fd]
0xca0b: mov     rax, qword ptr [rdx + 0x18]
0xca0f: sub     rax, qword ptr [rdx + 0x40]
0xca13: ret
0xca14: nop     word ptr cs:[rax + rax]
0xca1f: nop
0xca20: endbr64
0xca24: mov     rax, qword ptr [rip + 0x1e9dd]
0xca2b: mov     r8d, dword ptr [rax + 8]
0xca2f: mov     dword ptr [rax + 8], edi
0xca32: mov     eax, r8d
0xca35: ret
0xca36: nop     word ptr cs:[rax + rax]
0xca40: endbr64
0xca44: mov     rax, qword ptr [rip + 0x1e9bd]
0xca4b: cmp     qword ptr [rax + 0x48], 0
0xca50: sete    al
0xca53: movzx   eax, al
0xca56: ret
0xca57: nop     word ptr [rax + rax]
0xca60: endbr64
0xca64: mov     rax, qword ptr [rip + 0x1e99d]
0xca6b: mov     rdx, qword ptr [rax + 0x18]
0xca6f: mov     qword ptr [rax + 0x20], rdx
0xca73: ret
0xca74: nop     word ptr cs:[rax + rax]
0xca7f: nop
0xca80: endbr64
0xca84: sub     rsp, 8
0xca88: xor     eax, eax
0xca8a: mov     edi, 3
0xca8f: call    0xc6e0
0xca94: mov     rax, qword ptr [rip + 0x1e96d]
0xca9b: mov     dword ptr [rax + 0x10], 0x18
0xcaa2: add     rsp, 8
0xcaa6: ret
0xcaa7: nop     word ptr [rax + rax]
0xcab0: endbr64
0xcab4: sub     rsp, 8
0xcab8: xor     eax, eax
0xcaba: mov     edi, 1
0xcabf: call    0xc6e0
0xcac4: mov     rax, qword ptr [rip + 0x1e93d]
0xcacb: mov     dword ptr [rax + 0x10], 0x19
0xcad2: add     rsp, 8
0xcad6: ret
0xcad7: nop     word ptr [rax + rax]

```

```

0xcae0: endbr64
0xcae4: mov     rax, qword ptr [rip + 0x1e91d]
0xcaeb: mov     rdx, qword ptr [rax + 0x18]
0xcaef: mov     rcx, qword ptr [rax + 0x20]
0xcaf3: movsxd  rdx, edx
0xcaf6: mov     qword ptr [rax + 0x18], rcx
0xcafa: mov     qword ptr [rax + 0x20], rdx
0xcafe: ret
0xcaff: nop
0xcb00: endbr64
0xcb04: mov     rax, qword ptr [rip + 0x1e8fd]
0xcb0b: mov     rdx, qword ptr [rax + 0x18]
0xcb0f: cmp     rdx, qword ptr [rax + 0x40]
0xcb13: jbe     0xcb24
0xcb15: sub     rdx, 1
0xcb19: mov     dword ptr [rax + 0x10], 1
0xcb20: mov     qword ptr [rax + 0x18], rdx
0xcb24: ret
0xcb25: nop     word ptr cs:[rax + rax]
0xcb30: endbr64
0xcb34: mov     rax, qword ptr [rip + 0x1e8cd]
0xcb3b: mov     rdx, qword ptr [rax + 0x18]
0xcb3f: cmp     rdx, qword ptr [rax + 0x38]
0xcb43: jae     0xcb54
0xcb45: add     rdx, 1
0xcb49: mov     dword ptr [rax + 0x10], 2
0xcb50: mov     qword ptr [rax + 0x18], rdx
0xcb54: ret
0xcb55: nop     word ptr cs:[rax + rax]
0xcb60: endbr64
0xcb64: mov     rdx, qword ptr [rip + 0x1e89d]
0xcb6b: mov     rax, qword ptr [rdx + 0x18]
0xcb6f: mov     rdi, qword ptr [rdx + 0x40]
0xcb73: cmp     rax, rdi
0xcb76: jbe     0xcc20
0xcb7c: mov     r8, qword ptr [rdx + 0x58]
0xcb80: jmp     0xcba9
0xcba9:
0xcb82: nop     word ptr [rax + rax]
0xcb88: lea     ecx, [rsi - 0x30]
0xcb8b: cmp     cl, 9
0xcb8e: jbe     0xcbbf
0xcb90: sub     esi, 0x24
0xcb93: cmp     sil, 1
0xcb97: jbe     0xcbbf
0xcb99: mov     qword ptr [rdx + 0x18], rax
0xcb9d: mov     dword ptr [rdx + 0x10], 1
0xcba4: cmp     rdi, rax

```

```

0xcba7: jae      0xcbef
0xcba9: sub      rax, 1
0xcbad: movzx    esi, byte ptr [r8 + rax]
0xcbb2: mov      ecx, esi
0xcbb4: and      ecx, 0xffffffff
0xcbb7: sub      ecx, 0x41
0xcbb8: cmp      cl, 0x19
0xcbbd: ja       0xcb88
0xcbbf: mov      rax, qword ptr [rdx + 0x18]
0xcbc3: cmp      rdi, rax
0xcbc6: jae      0xcbef
0xcbc8: sub      rax, 1
0xcbcc: movzx    esi, byte ptr [r8 + rax]
0xcbd1: mov      ecx, esi
0xcbd3: and      ecx, 0xffffffff
0xcbd6: sub      ecx, 0x41
0xcbd9: cmp      cl, 0x19
0xcdbc: jbe     0xcc00
0xcbde: lea      ecx, [rsi - 0x30]
0xcbe1: cmp      cl, 9
0xcbe4: jbe     0xcc00
0xcbe6: sub      esi, 0x24
0xcbe9: cmp      sil, 1
0xcbed: jbe     0xcc00
0xcbef: mov      dword ptr [rdx + 0x10], 3
0xcbf6: ret
0xcbf7: nop      word ptr [rax + rax]
0xcc00: mov      qword ptr [rdx + 0x18], rax
0xcc04: mov      dword ptr [rdx + 0x10], 1
0xcc0b: cmp      rdi, rax
0xcc0e: jb       0xcbc8
0xcc10: mov      dword ptr [rdx + 0x10], 3
0xcc17: ret
0xcc18: nop      dword ptr [rax + rax]
0xcc20: ret
0xcc21: nop      word ptr cs:[rax + rax]
0xcc2c: nop      dword ptr [rax]
0xcc30: endbr64
0xcc34: mov      rdx, qword ptr [rip + 0x1e7cd]
0xcc3b: mov      rax, qword ptr [rdx + 0x18]
0xcc3f: mov      rdi, qword ptr [rdx + 0x38]
0xcc43: cmp      rax, rdi
0xcc46: jae      0xcce8
0xcc4c: mov      r8, qword ptr [rdx + 0x58]
0xcc50: jmp      0xcc7d
0xcc7d:
0xcc52: nop      word ptr [rax + rax]
0xcc58: lea      ecx, [rsi - 0x30]

```

```

0xcc5b: cmp     cl, 9
0xcc5e: jbe     0xcc8f
0xcc60: sub     esi, 0x24
0xcc63: cmp     sil, 1
0xcc67: jbe     0xcc8f
0xcc69: add     rax, 1
0xcc6d: mov     dword ptr [rdx + 0x10], 2
0xcc74: mov     qword ptr [rdx + 0x18], rax
0xcc78: cmp     rdi, rax
0xcc7b: je      0xccbb
0xcc7d: movzx   esi, byte ptr [r8 + rax]
0xcc82: mov     ecx, esi
0xcc84: and     ecx, 0xffffffff
0xcc87: sub     ecx, 0x41
0xcc8a: cmp     cl, 0x19
0xcc8d: ja      0xcc58
0xcc8f: mov     rax, qword ptr [rdx + 0x18]
0xcc93: cmp     rdi, rax
0xcc96: jbe     0xccbb
0xcc98: movzx   esi, byte ptr [r8 + rax]
0xcc9d: mov     ecx, esi
0xcc9f: and     ecx, 0xffffffff
0xcca2: sub     ecx, 0x41
0xcca5: cmp     cl, 0x19
0xcca8: jbe     0xccc8
0xccaa: lea     ecx, [rsi - 0x30]
0ccad: cmp     cl, 9
0xccb0: jbe     0xccc8
0xccb2: sub     esi, 0x24
0xccb5: cmp     sil, 1
0xccb9: jbe     0xccc8
0xccbb: mov     dword ptr [rdx + 0x10], 4
0ccc2: ret
0ccc3: nop     dword ptr [rax + rax]
0ccc8: add     rax, 1
0cccc: mov     dword ptr [rdx + 0x10], 2
0ccd3: mov     qword ptr [rdx + 0x18], rax
0ccd7: cmp     rdi, rax
0ccda: jne     0xcc98
0ccdc: mov     dword ptr [rdx + 0x10], 4
0xcce3: ret
0xcce4: nop     dword ptr [rax]
0xcce8: ret
0xcce9: nop     dword ptr [rax]
0ccf0: endbr64
0ccf4: mov     rax, qword ptr [rip + 0x1e70d]
0ccfb: mov     rdx, qword ptr [rax + 0x40]
0ccff: mov     dword ptr [rax + 0x10], 5

```

```

0xcd06: mov     qword ptr [rax + 0x18], rdx
0xcd0a: ret
0xcd0b: nop     dword ptr [rax + rax]
0xcd10: endbr64
0xcd14: mov     rax, qword ptr [rip + 0x1e6ed]
0xcd1b: mov     rdx, qword ptr [rax + 0x38]
0xcd1f: mov     dword ptr [rax + 0x10], 6
0xcd26: mov     qword ptr [rax + 0x18], rdx
0xcd2a: ret
0xcd2b: nop     dword ptr [rax + rax]
0xcd30: endbr64
0xcd34: push    rbp
0xcd35: push    rbx
0xcd36: mov     ebx, edi
0xcd38: sub     rsp, 8
0xcd3c: call    0x4f00
0xcd41: movsxd  rdx, ebx
0xcd44: mov     rax, qword ptr [rax]
0xcd47: test    byte ptr [rax + rdx*2 + 1], 0x40
0xcd4c: je      0xcdab
0xcd4e: mov     rbp, qword ptr [rip + 0x1e6b3]
0xcd55: mov     rsi, qword ptr [rbp + 0x38]
0xcd59: lea     rdx, [rsi + 1]
0xcd5d: cmp     rdx, qword ptr [rbp + 0x50]
0xcd61: jae     0xcdb8
0xcd63: mov     rax, qword ptr [rbp + 0x18]
0xcd67: mov     rcx, qword ptr [rbp + 0x58]
0xcd6b: sub     rdx, rax
0xcd6e: lea     rsi, [rcx + rax]
0xcd72: lea     rdi, [rcx + rax + 1]
0xcd77: call    0x4d40
0xcd7c: mov     rax, qword ptr [rip + 0x1e685]
0xcd83: mov     rdx, qword ptr [rax + 0x58]
0xcd87: mov     rax, qword ptr [rax + 0x18]
0xcd8b: mov     byte ptr [rdx + rax], bl
0xcd8e: mov     rax, qword ptr [rip + 0x1e673]
0xcd95: add     qword ptr [rax + 0x18], 1
0xcd9a: add     qword ptr [rax + 0x38], 1
0xcd9f: add     qword ptr [rax + 0x48], 1
0xcda4: mov     dword ptr [rax + 0x10], 7
0xcdab: add     rsp, 8
0xcdaf: pop     rbx
0xcdb0: pop     rbp
0xcdb1: ret
0xcdb2: nop     word ptr [rax + rax]
0xcdb8: add     rsi, 0x21
0xcdbc: mov     rdi, qword ptr [rbp + 0x58]
0xcdc0: mov     qword ptr [rbp + 0x50], rsi

```

```

0xcdc4: call    0x1d690
0xcdc9: mov     qword ptr [rbp + 0x58], rax
0xcdcd: mov     rbp, qword ptr [rip + 0x1e634]
0xcdd4: mov     rax, qword ptr [rbp + 0x38]
0xcdd8: lea     rdx, [rax + 1]
0xcddc: jmp     0xcd63
0xcd63:
0xcdde: nop
0xcde0: endbr64
0xcde4: mov     rax, qword ptr [rip + 0x1e61d]
0xcdeb: mov     rcx, qword ptr [rax + 0x18]
0xcdef: mov     rdx, qword ptr [rax + 0x38]
0xcdf3: cmp     rcx, rdx
0xcdf6: jnb     0xce00
0xcdf8: ret
0xcdf9: nop     dword ptr [rax]
0xce00: push    rbx
0xce01: mov     rax, qword ptr [rax + 0x58]
0xce05: add     rdx, 1
0xce09: sub     rdx, rcx
0xce0c: lea     rsi, [rax + rcx + 1]
0xce11: lea     rdi, [rax + rcx]
0xce15: call    0x4bb0
0xce1a: mov     rbx, qword ptr [rip + 0x1e5e7]
0xce21: mov     rsi, qword ptr [rbx + 0x38]
0xce25: sub     qword ptr [rbx + 0x48], 1
0xce2a: lea     rax, [rsi - 1]
0xce2e: mov     qword ptr [rbx + 0x38], rax
0xce32: test    al, 0xf
0xce34: je      0xce40
0xce36: mov     dword ptr [rbx + 0x10], 8
0xce3d: pop     rbx
0xce3e: ret
0xce3f: nop
0xce40: mov     qword ptr [rbx + 0x50], rsi
0xce44: mov     rdi, qword ptr [rbx + 0x58]
0xce48: call    0x1d690
0xce4d: mov     qword ptr [rbx + 0x58], rax
0xce51: mov     rbx, qword ptr [rip + 0x1e5b0]
0xce58: mov     dword ptr [rbx + 0x10], 8
0xce5f: pop     rbx
0xce60: ret
0xce61: nop     word ptr cs:[rax + rax]
0xce6c: nop     dword ptr [rax]
0xce70: endbr64
0xce74: mov     rax, qword ptr [rip + 0x1e58d]
0xce7b: mov     rcx, qword ptr [rax + 0x18]
0xce7f: cmp     rcx, qword ptr [rax + 0x40]

```



```

0xce83: ja      0xce90
0xce85: ret
0xce86: nop      word ptr cs:[rax + rax]
0xce90: push     rbx
0xce91: mov      rdx, qword ptr [rax + 0x38]
0xce95: mov      rsi, qword ptr [rax + 0x58]
0xce99: add      rdx, 1
0xce9d: lea      rdi, [rsi + rcx - 1]
0xcea2: sub      rdx, rcx
0xcea5: add      rsi, rcx
0xcea8: call     0x4bb0
0xcead: mov      rbx, qword ptr [rip + 0x1e554]
0xceb4: mov      rsi, qword ptr [rbx + 0x38]
0xceb8: sub      qword ptr [rbx + 0x18], 1
0xcebd: sub      qword ptr [rbx + 0x48], 1
0xcec2: lea      rax, [rsi - 1]
0xcec6: mov      qword ptr [rbx + 0x38], rax
0xceca: test     al, 0xf
0xcecc: je       0xcee0
0xcece: mov      dword ptr [rbx + 0x10], 9
0xcd5: pop      rbx
0xcd6: ret
0xcd7: nop      word ptr [rax + rax]
0xcee0: mov      qword ptr [rbx + 0x50], rsi
0xcee4: mov      rdi, qword ptr [rbx + 0x58]
0xcee8: call     0x1d690
0xceed: mov      qword ptr [rbx + 0x58], rax
0xcef1: mov      rbx, qword ptr [rip + 0x1e510]
0xcef8: mov      dword ptr [rbx + 0x10], 9
0xceff: pop      rbx
0xcf00: ret
0xcf01: nop      word ptr cs:[rax + rax]
0xcf0c: nop      dword ptr [rax]
0xcf10: endbr64
0xcf14: push     rbp
0xcf15: push     rbx
0xcf16: sub      rsp, 8
0xcf1a: mov      r9, qword ptr [rip + 0x1e4e7]
0xcf21: mov      rax, qword ptr [r9 + 0x18]
0xcf25: mov      rbx, qword ptr [r9 + 0x20]
0xcf29: mov      qword ptr [r9 + 0x20], rax
0xcf2d: xor      eax, eax
0xcf2f: call     0xcc30
0xcf34: xor      eax, eax
0xcf36: mov      edi, 3
0xcf3b: mov      rbp, qword ptr [r9 + 0x18]
0xcf3f: call     0xc6e0
0xcf44: mov      rax, qword ptr [rip + 0x1e4bd]

```

```

0xcf4b: mov     rdx, qword ptr [rax + 0x18]
0xcf4f: cmp     rdx, rbx
0xcf52: jae     0xcf65
0xcf54: lea     rcx, [rdx + rbx]
0xcf58: sub     rcx, rbp
0xcf5b: cmp     rbx, rbp
0xcf5e: cmovae rdx, rcx
0xcf62: mov     rbx, rdx
0xcf65: mov     qword ptr [rax + 0x20], rbx
0xcf69: mov     dword ptr [rax + 0x10], 0xa
0xcf70: add     rsp, 8
0xcf74: pop     rbx
0xcf75: pop     rbp
0xcf76: ret
0xcf77: nop     word ptr [rax + rax]
0xcf80: endbr64
0xcf84: push    rbp
0xcf85: push    rbx
0xcf86: sub     rsp, 8
0xcf8a: mov     rax, qword ptr [rip + 0x1e477]
0xcf91: mov     rbp, qword ptr [rax + 0x18]
0xcf95: mov     rbx, qword ptr [rax + 0x20]
0xcf99: mov     qword ptr [rax + 0x20], rbp
0xcf9d: xor     eax, eax
0xcf9f: call    0xcb60
0xcfa4: xor     eax, eax
0xcfa6: mov     edi, 3
0xcfab: call    0xc6e0
0xcfb0: mov     rax, qword ptr [rip + 0x1e451]
0xcfb7: mov     rdx, qword ptr [rax + 0x18]
0xcfb9: cmp     rdx, rbx
0xcfbe: jae     0xcfd1
0xcfc0: lea     rcx, [rdx + rbx]
0xcfc4: sub     rcx, rbp
0xcfc7: cmp     rbx, rbp
0xcfc9: cmovae rdx, rcx
0xcfce: mov     rbx, rdx
0xcfd1: mov     qword ptr [rax + 0x20], rbx
0xcfd5: mov     dword ptr [rax + 0x10], 0xb
0xcfdc: add     rsp, 8
0xcfe0: pop     rbx
0xcfe1: pop     rbp
0xcfe2: ret
0xcfe3: nop     word ptr cs:[rax + rax]
0xcfee: nop
0xcff0: endbr64
0xcff4: push    rbx
0xcff5: mov     rbx, qword ptr [rip + 0x1e40c]

```

```

0xcffc: mov     esi, 1
0xd001: mov     qword ptr [rbx + 0x18], 0
0xd009: mov     rdi, qword ptr [rbx + 0x58]
0xd00d: mov     qword ptr [rbx + 0x20], 0
0xd015: mov     qword ptr [rbx + 0x38], 0
0xd01d: mov     qword ptr [rbx + 0x40], 0
0xd025: mov     qword ptr [rbx + 0x48], 0
0xd02d: mov     qword ptr [rbx + 0x50], 1
0xd035: call    0x1d690
0xd03a: mov     qword ptr [rbx + 0x58], rax
0xd03e: mov     rax, qword ptr [rip + 0x1e3c3]
0xd045: mov     rax, qword ptr [rax + 0x58]
0xd049: mov     byte ptr [rax], 0
0xd04c: mov     rax, qword ptr [rip + 0x1e3b5]
0xd053: mov     dword ptr [rax + 0x10], 0xc
0xd05a: pop     rbx
0xd05b: ret
0xd05c: nop     dword ptr [rax]
0xd060: endbr64
0xd064: push    rbx
0xd065: mov     edi, 0x70
0xd06a: sub     rsp, 0x10
0xd06e: mov     rax, qword ptr fs:[0x28]
0xd077: mov     qword ptr [rsp + 8], rax
0xd07c: xor     eax, eax
0xd07e: call    0x1d670
0xd083: mov     qword ptr [rax + 8], 1
0xd08b: mov     qword ptr [rax + 0x58], 0
0xd093: mov     qword ptr [rax + 0x60], 0
0xd09b: mov     qword ptr [rax + 0x28], 0
0xd0a3: mov     qword ptr [rax + 0x30], 0
0xd0ab: mov     qword ptr [rip + 0x1e356], rax
0xd0b2: xor     eax, eax
0xd0b4: call    0xcff0
0xd0b9: mov     rbx, qword ptr [rip + 0x1e348]
0xd0c0: xor     eax, eax
0xd0c2: call    0xc590
0xd0c7: lea     rdi, [rip + 0x135e4]
0xd0ce: mov     qword ptr [rbx], rax
0xd0d1: call    0x17c50
0xd0d6: mov     ecx, 1
0xd0db: mov     rsi, rsp
0xd0de: mov     edx, 1
0xd0e3: lea     rdi, [rip + 0x13e42]
0xd0ea: call    0x17320
0xd0ef: mov     rdi, qword ptr [rsp]
0xd0f3: lea     rax, [rip + 0x13e24]
0xd0fa: mov     rbx, qword ptr [rip + 0x1e307]

```

```

0xd101: test    rdi, rdi
0xd104: cmove   rdi, rax
0xd108: xor     eax, eax
0xd10a: call    0x19940
0xd10f: lea     rdi, [rip + 0x1cb32]
0xd116: mov     qword ptr [rbx + 0x68], rax
0xd11a: mov     eax, dword ptr [rip + 0x1cb40]
0xd120: test    eax, eax
0xd122: lea     rax, [rip + 0x1cb07]
0xd129: cmove   rdi, rax
0xd12d: call    0x17c50
0xd132: lea     rdi, [rip + 0x1d0c7]
0xd139: mov     edx, 6
0xd13e: lea     rsi, [rip + 0x1d0db]
0xd145: call    0x17f70
0xd14a: xor     eax, eax
0xd14c: call    0x1a310
0xd151: mov     rax, qword ptr [rip + 0x1e2b0]
0xd158: mov     rdi, qword ptr [rax + 0x68]
0xd15c: xor     eax, eax
0xd15e: call    0x1a900
0xd163: nop     dword ptr [rax + rax]
0xd168: xor     eax, eax
0xd16a: call    0x1ab40
0xd16f: test    rax, rax
0xd172: jne     0xd168
0xd174: mov     edi, 0x200
0xd179: call    0x1a620
0xd17e: mov     rax, qword ptr [rsp + 8]
0xd183: sub     rax, qword ptr fs:[0x28]
0xd18c: jne     0xd194
0xd18e: add     rsp, 0x10
0xd192: pop     rbx
0xd193: ret
0xd194: call    0x49d0
0xd199: nop     dword ptr [rax]
0xd1a0: endbr64
0xd1a4: sub     rsp, 8
0xd1a8: mov     rax, qword ptr [rip + 0x1e259]
0xd1af: or      edi, 2
0xd1b2: mov     rdx, qword ptr [rax + 0x40]
0xd1b6: mov     dword ptr [rax + 0x10], 6
0xd1bd: mov     qword ptr [rax + 0x20], rdx
0xd1c1: mov     rdx, qword ptr [rax + 0x38]
0xd1c5: mov     qword ptr [rax + 0x18], rdx
0xd1c9: xor     eax, eax
0xd1cb: call    0xc6e0
0xd1d0: mov     rax, qword ptr [rip + 0x1e231]

```

```

0xd1d7: mov     rdx, qword ptr [rax + 0x18]
0xd1db: mov     dword ptr [rax + 0x10], 0xd
0xd1e2: mov     qword ptr [rax + 0x20], rdx
0xd1e6: add     rsp, 8
0xd1ea: ret
0xd1eb: nop     dword ptr [rax + rax]
0xd1f0: endbr64
0xd1f4: mov     rax, qword ptr [rip + 0x1e20d]
0xd1fb: push    rbx
0xd1fc: mov     edi, 3
0xd201: mov     rdx, qword ptr [rax + 0x18]
0xd205: mov     rsi, qword ptr [rax + 0x20]
0xd209: mov     dword ptr [rax + 0x10], 5
0xd210: mov     rcx, qword ptr [rax + 0x40]
0xd214: mov     qword ptr [rax + 0x20], rdx
0xd218: mov     rbx, rsi
0xd21b: mov     qword ptr [rax + 0x18], rcx
0xd21f: sub     rbx, rdx
0xd222: cmp     rsi, rdx
0xd225: cmovbe  rbx, rcx
0xd229: xor     eax, eax
0xd22b: call    0xc6e0
0xd230: mov     rdx, qword ptr [rip + 0x1e1d1]
0xd237: mov     rax, qword ptr [rdx + 0x38]
0xd23b: mov     dword ptr [rdx + 0x10], 0xe
0xd242: cmp     rax, rbx
0xd245: cmova   rax, rbx
0xd249: mov     qword ptr [rdx + 0x20], rax
0xd24d: pop     rbx
0xd24e: ret
0xd24f: nop
0xd250: endbr64
0xd254: mov     rax, qword ptr [rip + 0x1e1ad]
0xd25b: push    rbx
0xd25c: mov     edi, 3
0xd261: mov     rdx, qword ptr [rax + 0x18]
0xd265: mov     rbx, qword ptr [rax + 0x20]
0xd269: mov     dword ptr [rax + 0x10], 6
0xd270: mov     qword ptr [rax + 0x20], rdx
0xd274: mov     rdx, qword ptr [rax + 0x38]
0xd278: mov     qword ptr [rax + 0x18], rdx
0xd27c: xor     eax, eax
0xd27e: call    0xc6e0
0xd283: mov     rdx, qword ptr [rip + 0x1e17e]
0xd28a: mov     rax, qword ptr [rdx + 0x38]
0xd28e: mov     dword ptr [rdx + 0x10], 0xf
0xd295: cmp     rax, rbx
0xd298: cmova   rax, rbx

```

```

0xd29c: mov     qword ptr [rdx + 0x20], rax
0xd2a0: pop     rbx
0xd2a1: ret
0xd2a2: nop     word ptr cs:[rax + rax]
0xd2ad: nop     dword ptr [rax]
0xd2b0: endbr64
0xd2b4: mov     rax, qword ptr [rip + 0x1e14d]
0xd2bb: mov     rdx, qword ptr [rax + 0x58]
0xd2bf: mov     rax, qword ptr [rax + 0x18]
0xd2c3: cmp     byte ptr [rdx + rax], 0x20
0xd2c7: je      0xd2d0
0xd2c9: ret
0xd2ca: nop     word ptr [rax + rax]
0xd2d0: sub     rsp, 8
0xd2d4: nop     dword ptr [rax]
0xd2d8: xor     eax, eax
0xd2da: call    0xcde0
0xd2df: mov     rax, qword ptr [rip + 0x1e122]
0xd2e6: mov     rdx, qword ptr [rax + 0x58]
0xd2ea: mov     rcx, qword ptr [rax + 0x18]
0xd2ee: cmp     byte ptr [rdx + rcx], 0x20
0xd2f2: je      0xd2d8
0xd2f4: cmp     qword ptr [rax + 0x48], 0
0xd2f9: jne     0xd31d
0xd2fb: jmp     0xd324
0xd324:
0xd2fd: nop     dword ptr [rax]
0xd300: xor     eax, eax
0xd302: call    0xce70
0xd307: mov     rax, qword ptr [rip + 0x1e0fa]
0xd30e: cmp     qword ptr [rax + 0x48], 0
0xd313: je      0xd324
0xd315: mov     rdx, qword ptr [rax + 0x58]
0xd319: mov     rcx, qword ptr [rax + 0x18]
0xd31d: cmp     byte ptr [rdx + rcx - 1], 0x20
0xd322: je      0xd300
0xd324: mov     dword ptr [rax + 0x10], 0x11
0xd32b: add     rsp, 8
0xd32f: ret
0xd330: endbr64
0xd334: mov     rax, qword ptr [rip + 0x1e0cd]
0xd33b: mov     rdx, qword ptr [rax + 0x58]
0xd33f: mov     rax, qword ptr [rax + 0x18]
0xd343: cmp     byte ptr [rdx + rax], 0x20
0xd347: je      0xd350
0xd349: ret
0xd34a: nop     word ptr [rax + rax]
0xd350: sub     rsp, 8

```

```

0xd354: xor      eax, eax
0xd356: call     0xd2b0
0xd35b: mov      edi, 0x20
0xd360: call     0xcd30
0xd365: mov      rax, qword ptr [rip + 0x1e09c]
0xd36c: mov      dword ptr [rax + 0x10], 0x10
0xd373: add      rsp, 8
0xd377: ret
0xd378: nop      dword ptr [rax + rax]
0xd380: endbr64
0xd384: push     rbp
0xd385: push     rbx
0xd386: sub      rsp, 8
0xd38a: mov      rbp, qword ptr [rip + 0x1e077]
0xd391: mov      rbx, qword ptr [rbp + 0x18]
0xd395: cmp      rbx, qword ptr [rbp + 0x38]
0xd399: jb       0xd3a8
0xd39b: add      rsp, 8
0xd39f: pop      rbx
0xd3a0: pop      rbp
0xd3a1: ret
0xd3a2: nop      word ptr [rax + rax]
0xd3a8: xor      eax, eax
0xd3aa: call     0xcc30
0xd3af: cmp      rbx, qword ptr [rbp + 0x18]
0xd3b3: jae      0xd3e4
0xd3b5: call     0x4ef0
0xd3ba: nop      word ptr [rax + rax]
0xd3c0: mov      rdx, qword ptr [rbp + 0x58]
0xd3c4: mov      rcx, qword ptr [rax]
0xd3c7: add      rdx, rbx
0xd3ca: add      rbx, 1
0xd3ce: movsx    rsi, byte ptr [rdx]
0xd3d2: mov      ecx, dword ptr [rcx + rsi*4]
0xd3d5: mov      byte ptr [rdx], cl
0xd3d7: mov      rbp, qword ptr [rip + 0x1e02a]
0xd3de: cmp      qword ptr [rbp + 0x18], rbx
0xd3e2: ja       0xd3c0
0xd3e4: mov      dword ptr [rbp + 0x10], 0x12
0xd3eb: add      rsp, 8
0xd3ef: pop      rbx
0xd3f0: pop      rbp
0xd3f1: ret
0xd3f2: nop      word ptr cs:[rax + rax]
0xd3fd: nop      dword ptr [rax]
0xd400: endbr64
0xd404: push     rbp
0xd405: push     rbx

```

```

0xd406: sub      rsp, 8
0xd40a: mov      rbp, qword ptr [rip + 0x1dff7]
0xd411: mov      rbx, qword ptr [rbp + 0x18]
0xd415: cmp      rbx, qword ptr [rbp + 0x38]
0xd419: jb       0xd428
0xd41b: add      rsp, 8
0xd41f: pop      rbx
0xd420: pop      rbp
0xd421: ret
0xd422: nop      word ptr [rax + rax]
0xd428: xor      eax, eax
0xd42a: call     0xcc30
0xd42f: cmp      rbx, qword ptr [rbp + 0x18]
0xd433: jae      0xd464
0xd435: call     0x47c0
0xd43a: nop      word ptr [rax + rax]
0xd440: mov      rdx, qword ptr [rbp + 0x58]
0xd444: mov      rcx, qword ptr [rax]
0xd447: add      rdx, rbx
0xd44a: add      rbx, 1
0xd44e: movsx    rsi, byte ptr [rdx]
0xd452: mov      ecx, dword ptr [rcx + rsi*4]
0xd455: mov      byte ptr [rdx], cl
0xd457: mov      rbp, qword ptr [rip + 0x1dfaa]
0xd45e: cmp      qword ptr [rbp + 0x18], rbx
0xd462: ja       0xd440
0xd464: mov      dword ptr [rbp + 0x10], 0x13
0xd46b: add      rsp, 8
0xd46f: pop      rbx
0xd470: pop      rbp
0xd471: ret
0xd472: nop      word ptr cs:[rax + rax]
0xd47d: nop      dword ptr [rax]
0xd480: endbr64
0xd484: push     r14
0xd486: push     r13
0xd488: push     r12
0xd48a: push     rbp
0xd48b: mov      rbp, qword ptr [rip + 0x1df76]
0xd492: push     rbx
0xd493: mov      rbx, qword ptr [rbp + 0x18]
0xd497: cmp      rbx, qword ptr [rbp + 0x38]
0xd49b: jb       0xd4b0
0xd49d: pop      rbx
0xd49e: pop      rbp
0xd49f: pop      r12
0xd4a1: pop      r13
0xd4a3: pop      r14

```



```

0xd4a5: ret
0xd4a6: nop      word ptr cs:[rax + rax]
0xd4b0: xor      eax, eax
0xd4b2: call     0xcc30
0xd4b7: cmp      rbx, qword ptr [rbp + 0x18]
0xd4bb: jae      0xd528
0xd4bd: call     0x4f00
0xd4c2: mov      r14d, 1
0xd4c8: mov      r13, rax
0xd4cb: jmp      0xd4f3
0xd4f3:
0xd4cd: nop      dword ptr [rax]
0xd4d0: call     0x47c0
0xd4d5: xor      r14d, r14d
0xd4d8: mov      rax, qword ptr [rax]
0xd4db: mov      eax, dword ptr [rax + r12*4]
0xd4df: mov      byte ptr [rbp], al
0xd4e2: mov      rbp, qword ptr [rip + 0x1df1f]
0xd4e9: add      rbx, 1
0xd4ed: cmp      qword ptr [rbp + 0x18], rbx
0xd4f1: jbe      0xd528
0xd4f3: mov      rax, qword ptr [rbp + 0x58]
0xd4f7: add      rax, rbx
0xd4fa: movsx    r12, byte ptr [rax]
0xd4fe: mov      rbp, rax
0xd501: mov      rax, qword ptr [r13]
0xd505: test     byte ptr [rax + r12*2], 8
0xd50a: je       0xd4e2
0xd50c: test     r14d, r14d
0xd50f: jne      0xd4d0
0xd511: call     0x4ef0
0xd516: mov      rax, qword ptr [rax]
0xd519: mov      eax, dword ptr [rax + r12*4]
0xd51d: mov      byte ptr [rbp], al
0xd520: jmp      0xd4e2
0xd4e2:
0xd522: nop      word ptr [rax + rax]
0xd528: mov      dword ptr [rbp + 0x10], 0x14
0xd52f: pop      rbx
0xd530: pop      rbp
0xd531: pop      r12
0xd533: pop      r13
0xd535: pop      r14
0xd537: ret
0xd538: nop      dword ptr [rax + rax]
0xd540: endbr64
0xd544: push     r12
0xd546: mov      r12, rdi

```

```

0xd549: push    rbp
0xd54a: push    rbx
0xd54b: call    0x49b0
0xd550: mov     rbp, qword ptr [rip + 0x1deb1]
0xd557: mov     rbx, rax
0xd55a: mov     rsi, qword ptr [rbp + 0x40]
0xd55e: sub     rax, rsi
0xd561: add     qword ptr [rbp + 0x18], rax
0xd565: add     qword ptr [rbp + 0x20], rax
0xd569: mov     rax, qword ptr [rbp + 0x48]
0xd56d: lea     rdx, [rax + 1]
0xd571: lea     r8, [rbx + rdx]
0xd575: cmp     r8, qword ptr [rbp + 0x50]
0xd579: ja      0xd5d8
0xd57b: mov     rdi, qword ptr [rbp + 0x58]
0xd57f: add     rsi, rdi
0xd582: add     rdi, rbx
0xd585: call    0x4d40
0xd58a: mov     rax, qword ptr [rip + 0x1de77]
0xd591: mov     rdx, rbx
0xd594: mov     rsi, r12
0xd597: mov     rdi, qword ptr [rax + 0x58]
0xd59b: call    0x4bb0
0xd5a0: mov     rax, qword ptr [rip + 0x1de61]
0xd5a7: mov     rsi, rbx
0xd5aa: mov     rdi, qword ptr [rax + 0x58]
0xd5ae: call    0x184d0
0xd5b3: mov     rax, qword ptr [rip + 0x1de4e]
0xd5ba: mov     qword ptr [rax + 0x40], rbx
0xd5be: add     rbx, qword ptr [rax + 0x48]
0xd5c2: mov     qword ptr [rax + 0x38], rbx
0xd5c6: mov     dword ptr [rax + 0x10], 0x15
0xd5cd: pop     rbx
0xd5ce: pop     rbp
0xd5cf: pop     r12
0xd5d1: ret
0xd5d2: nop     word ptr [rax + rax]
0xd5d8: mov     qword ptr [rbp + 0x50], r8
0xd5dc: mov     rdi, qword ptr [rbp + 0x58]
0xd5e0: mov     rsi, r8
0xd5e3: call    0x1d690
0xd5e8: mov     qword ptr [rbp + 0x58], rax
0xd5ec: mov     rbp, qword ptr [rip + 0x1de15]
0xd5f3: mov     rax, qword ptr [rbp + 0x48]
0xd5f7: mov     rsi, qword ptr [rbp + 0x40]
0xd5fb: lea     rdx, [rax + 1]
0xd5ff: jmp     0xd57b
0xd57b

```

```

0xd604: nop      word ptr cs:[rax + rax]
0xd60f: nop
0xd610: endbr64
0xd614: test     rdi, rdi
0xd617: je       0xd6f0
0xd61d: push     r12
0xd61f: push     rbp
0xd620: mov      rbp, rdi
0xd623: push     rbx
0xd624: call     0x49b0
0xd629: mov      r12, qword ptr [rip + 0x1ddd8]
0xd630: mov      rbx, rax
0xd633: mov      rax, qword ptr [r12 + 0x38]
0xd638: lea      rdx, [rax + 1]
0xd63c: mov      rax, qword ptr [r12 + 0x50]
0xd641: lea      rcx, [rbx + rdx]
0xd645: cmp      rcx, rax
0xd648: ja       0xd6c0
0xd64a: mov      rax, qword ptr [r12 + 0x18]
0xd64f: mov      rdi, qword ptr [r12 + 0x58]
0xd654: sub      rdx, rax
0xd657: lea      rsi, [rdi + rax]
0xd65b: add      rax, rbx
0xd65e: add      rdi, rax
0xd661: call     0x4d40
0xd666: mov      rax, qword ptr [rip + 0x1dd9b]
0xd66d: mov      rdx, rbx
0xd670: mov      rsi, rbp
0xd673: mov      rdi, qword ptr [rax + 0x18]
0xd677: add      rdi, qword ptr [rax + 0x58]
0xd67b: call     0x4bb0
0xd680: mov      rax, qword ptr [rip + 0x1dd81]
0xd687: mov      rsi, rbx
0xd68a: mov      rdi, qword ptr [rax + 0x18]
0xd68e: add      rdi, qword ptr [rax + 0x58]
0xd692: call     0x184d0
0xd697: mov      rax, qword ptr [rip + 0x1dd6a]
0xd69e: add      qword ptr [rax + 0x18], rbx
0xd6a2: add      qword ptr [rax + 0x38], rbx
0xd6a6: add      qword ptr [rax + 0x48], rbx
0xd6aa: mov      dword ptr [rax + 0x10], 0x16
0xd6b1: pop      rbx
0xd6b2: pop      rbp
0xd6b3: pop      r12
0xd6b5: ret
0xd6b6: nop      word ptr cs:[rax + rax]
0xd6c0: lea      rsi, [rax + rbx + 0x21]
0xd6c5: mov      rdi, qword ptr [r12 + 0x58]

```

```

0xd6ca: mov     qword ptr [r12 + 0x50], rsi
0xd6cf: call    0x1d690
0xd6d4: mov     qword ptr [r12 + 0x58], rax
0xd6d9: mov     r12, qword ptr [rip + 0x1dd28]
0xd6e0: mov     rax, qword ptr [r12 + 0x38]
0xd6e5: lea     rdx, [rax + 1]
0xd6e9: jmp     0xd64a
0xd64a
0xd6ee: nop
0xd6f0: ret
0xd6f1: nop     word ptr cs:[rax + rax]
0xd6fc: nop     dword ptr [rax]
0xd700: endbr64
0xd704: mov     rax, qword ptr [rip + 0x1dcfd]
0xd70b: mov     rdi, qword ptr [rax + 0x60]
0xd70f: test    rdi, rdi
0xd712: je      0xd730
0xd714: sub     rsp, 8
0xd718: call    0xd610
0xd71d: mov     rax, qword ptr [rip + 0x1dce4]
0xd724: mov     dword ptr [rax + 0x10], 0x1a
0xd72b: add     rsp, 8
0xd72f: ret
0xd730: ret
0xd731: nop     word ptr cs:[rax + rax]
0xd73c: nop     dword ptr [rax]
0xd740: endbr64
0xd744: mov     rax, qword ptr [rip + 0x1dcbd]
0xd74b: push    r12
0xd74d: push    rbp
0xd74e: push    rbx
0xd74f: mov     rbx, qword ptr [rax + 0x40]
0xd753: mov     r12, qword ptr [rax + 0x28]
0xd757: mov     eax, dword ptr [rax + 0xc]
0xd75a: lea     rbp, [rbx + 3]
0xd75e: test    eax, eax
0xd760: je      0xd7f0
0xd766: mov     edx, dword ptr [rip + 0x1caa4]
0xd76c: mov     esi, dword ptr [rip + 0x1ca9a]
0xd772: mov     edi, dword ptr [rip + 0x1ca9c]
0xd778: call    0xa570
0xd77d: mov     rcx, qword ptr [rip + 0x1dc84]
0xd784: cmp     r12, rbp
0xd787: jb      0xd810
0xd78d: mov     rsi, qword ptr [rcx + 0x18]
0xd791: mov     rdx, qword ptr [rcx + 0x40]
0xd795: xor     ebx, ebx
0xd797: mov     rdi, qword ptr [rcx + 0x28]

```

```

0xd79b: mov     rax, rdi
0xd79e: sub     rax, rdx
0xd7a1: mov     edx, 1
0xd7a6: mov     r8, rax
0xd7a9: shr     r8, 2
0xd7ad: cmp     rax, 3
0xd7b1: cmova   edx, r8d
0xd7b5: xor     eax, eax
0xd7b7: cmp     rdi, rsi
0xd7ba: ja      0xd7d3
0xd7bc: mov     rax, rsi
0xd7bf: sub     rax, rdi
0xd7c2: movsxd  rdi, edx
0xd7c5: lea     r8d, [rdx + rax]
0xd7c9: xor     edx, edx
0xd7cb: div     rdi
0xd7ce: mov     eax, r8d
0xd7d1: sub     eax, edx
0xd7d3: mov     edx, esi
0xd7d5: mov     rdi, qword ptr [rcx]
0xd7d8: xor     esi, esi
0xd7da: sub     edx, eax
0xd7dc: call    0xc680
0xd7e1: cmp     r12, rbp
0xd7e4: jb      0xd840
0xd7e6: pop     rbx
0xd7e7: pop     rbp
0xd7e8: pop     r12
0xd7ea: ret
0xd7eb: nop     dword ptr [rax + rax]
0xd7f0: mov     edx, dword ptr [rip + 0x1ca0e]
0xd7f6: mov     esi, dword ptr [rip + 0x1ca04]
0xd7fc: mov     edi, dword ptr [rip + 0x1ca06]
0xd802: jmp     0xd778
0xd778:
0xd807: nop     word ptr [rax + rax]
0xd810: mov     rbx, qword ptr [rcx + 0x40]
0xd814: mov     rsi, qword ptr [rcx + 0x18]
0xd818: mov     qword ptr [rcx + 0x40], 0
0xd820: xor     edx, edx
0xd822: add     qword ptr [rcx + 0x58], rbx
0xd826: sub     rsi, rbx
0xd829: sub     qword ptr [rcx + 0x38], rbx
0xd82d: sub     qword ptr [rcx + 0x20], rbx
0xd831: movsxd  rbx, ebx
0xd834: mov     qword ptr [rcx + 0x18], rsi
0xd838: jmp     0xd797
0xd797:

```

```

0xd83d: nop      dword ptr [rax]
0xd840: mov      rax, qword ptr [rip + 0x1dbc1]
0xd847: movsxd   rbx, ebx
0xd84a: mov      qword ptr [rax + 0x40], rbx
0xd84e: sub      qword ptr [rax + 0x58], rbx
0xd852: add      qword ptr [rax + 0x38], rbx
0xd856: add      qword ptr [rax + 0x18], rbx
0xd85a: add      qword ptr [rax + 0x20], rbx
0xd85e: pop      rbx
0xd85f: pop      rbp
0xd860: pop      r12
0xd862: ret
0xd863: nop      word ptr cs:[rax + rax]
0xd86e: nop
0xd870: endbr64
0xd874: push     r15
0xd876: push     r14
0xd878: push     r13
0xd87a: push     r12
0xd87c: push     rbp
0xd87d: push     rbx
0xd87e: sub      rsp, 0x28
0xd882: mov      rax, qword ptr fs:[0x28]
0xd88b: mov      qword ptr [rsp + 0x18], rax
0xd890: mov      rax, qword ptr [rip + 0x1db71]
0xd897: lea      r14, [rsp + 0x17]
0xd89c: mov      rdi, r14
0xd89f: mov      rbx, qword ptr [rax + 0x40]
0xd8a3: mov      r13, qword ptr [rax + 0x28]
0xd8a7: call     0xa6a0
0xd8ac: mov      rax, qword ptr [rip + 0x1db55]
0xd8b3: add      rbx, 3
0xd8b7: mov      ecx, dword ptr [rax + 0xc]
0xd8ba: test     ecx, ecx
0xd8bc: je       0xda20
0xd8c2: mov      edx, dword ptr [rip + 0x1c948]
0xd8c8: mov      esi, dword ptr [rip + 0x1c93e]
0xd8ce: mov      edi, dword ptr [rip + 0x1c940]
0xd8d4: call     0xa570
0xd8d9: mov      rdx, qword ptr [rip + 0x1db28]
0xd8e0: cmp      r13, rbx
0xd8e3: jb       0xda90
0xd8e9: mov      qword ptr [rsp + 8], 0
0xd8f2: mov      rax, qword ptr [rdx + 0x18]
0xd8f6: mov      rcx, qword ptr [rdx + 0x40]
0xd8fa: mov      rdi, qword ptr [rdx + 0x28]
0xd8fe: mov      r15, rdi
0xd901: sub      r15, rcx

```

```

0xd904: cmp      r15, 3
0xd908: jbe      0xda80
0xd90e: shr      r15, 2
0xd912: movsxd   rcx, r15d
0xd915: cmp      rdi, rax
0xd918: ja      0xda70
0xd91e: mov      edx, eax
0xd920: sub      rax, rdi
0xd923: sub      edx, edi
0xd925: add      r15d, edx
0xd928: xor      edx, edx
0xd92a: div      rcx
0xd92d: sub      r15d, edx
0xd930: mov      r12d, r15d
0xd933: call     0x1d670
0xd938: mov      esi, 0x20
0xd93d: mov      rbp, rax
0xd940: mov      rax, qword ptr [rip + 0x1dac1]
0xd947: mov      rdi, rbp
0xd94a: mov      rdx, qword ptr [rax + 0x28]
0xd94e: call     0x4a50
0xd953: mov      rax, qword ptr [rip + 0x1daae]
0xd95a: mov      edx, dword ptr [rax + 8]
0xd95d: test     edx, edx
0xd95f: je      0xda40
0xd965: mov      rdx, qword ptr [rax + 0x38]
0xd969: sub      rdx, qword ptr [rax + 0x40]
0xd96d: mov      rdi, rbp
0xd970: sub      rdx, r12
0xd973: mov      rsi, qword ptr [rax + 0x28]
0xd977: sub      rsi, qword ptr [rax + 0x40]
0xd97b: cmp      rdx, rsi
0xd97e: cmova    rdx, rsi
0xd982: mov      rsi, qword ptr [rax + 0x40]
0xd986: add      rsi, r12
0xd989: add      rsi, qword ptr [rax + 0x58]
0xd98d: call     0x4bb0
0xd992: mov      rax, qword ptr [rip + 0x1da6f]
0xd999: xor      edx, edx
0xd99b: xor      esi, esi
0xd99d: mov      rdi, qword ptr [rax]
0xd9a0: call     0xc680
0xd9a5: cmp      r13, rbx
0xd9a8: jae      0xdaf0
0xd9ae: mov      rax, qword ptr [rip + 0x1da53]
0xd9b5: mov      rsi, rbp
0xd9b8: mov      rdi, qword ptr [rax]
0xd9bb: mov      rdx, qword ptr [rax + 0x28]

```

```

0xd9bf: sub     edx, dword ptr [rax + 0x40]
0xd9c2: call    0xc600
0xd9c7: mov     rax, qword ptr [rip + 0x1da3a]
0xd9ce: xor     esi, esi
0xd9d0: mov     edx, dword ptr [rax + 0x18]
0xd9d3: mov     rdi, qword ptr [rax]
0xd9d6: sub     edx, r15d
0xd9d9: call    0xc680
0xd9de: cmp     r13, rbx
0xd9e1: jb      0xdac8
0xd9e7: mov     rdi, rbp
0xd9ea: call    0x18d10
0xd9ef: mov     rdi, r14
0xd9f2: call    0xa6b0
0xd9f7: mov     rax, qword ptr [rsp + 0x18]
0xd9fc: sub     rax, qword ptr fs:[0x28]
0xda05: jne     0xdb0b
0xda0b: add     rsp, 0x28
0xda0f: pop     rbx
0xda10: pop     rbp
0xda11: pop     r12
0xda13: pop     r13
0xda15: pop     r14
0xda17: pop     r15
0xda19: ret
0xda1a: nop     word ptr [rax + rax]
0xda20: mov     edx, dword ptr [rip + 0x1c7de]
0xda26: mov     esi, dword ptr [rip + 0x1c7d4]
0xda2c: mov     edi, dword ptr [rip + 0x1c7d6]
0xda32: jmp     0xd8d4
0xd8d4
0xda37: nop     word ptr [rax + rax]
0xda40: mov     rsi, qword ptr [rax + 0x40]
0xda44: mov     rdx, qword ptr [rax + 0x38]
0xda48: mov     rdi, rbp
0xda4b: mov     rax, qword ptr [rax + 0x28]
0xda4f: sub     rdx, rsi
0xda52: sub     rdx, r12
0xda55: sub     rax, rsi
0xda58: mov     esi, 0x2a
0xda5d: cmp     rdx, rax
0xda60: cmova   rdx, rax
0xda64: call    0x4a50
0xda69: jmp     0xd992
0xd992
0xda6e: nop
0xda70: xor     r12d, r12d
0xda73: xor     r15d, r15d

```



```

0xda76: jmp      0xd933
0xd933:
0xda7b: nop      dword ptr [rax + rax]
0xda80: mov      ecx, 1
0xda85: mov      r15d, 1
0xda8b: jmp      0xd915
0xd915:
0xda90: mov      r12, qword ptr [rdx + 0x40]
0xda94: mov      rax, qword ptr [rdx + 0x18]
0xda98: mov      qword ptr [rdx + 0x40], 0
0xdaa0: add      qword ptr [rdx + 0x58], r12
0xdaa4: movsxd   rcx, r12d
0xdaa7: sub      rax, r12
0xdaaa: sub      qword ptr [rdx + 0x38], r12
0xdaae: mov      qword ptr [rsp + 8], rcx
0xdab3: xor      ecx, ecx
0xdab5: sub      qword ptr [rdx + 0x20], r12
0xdab9: mov      qword ptr [rdx + 0x18], rax
0xdabd: jmp      0xd8fa
0xd8fa:
0xdac2: nop      word ptr [rax + rax]
0xdac8: mov      rax, qword ptr [rip + 0x1d939]
0xdacf: movsxd   r12, dword ptr [rsp + 8]
0xdad4: sub      qword ptr [rax + 0x58], r12
0xdad8: add      qword ptr [rax + 0x38], r12
0xdadc: add      qword ptr [rax + 0x18], r12
0xdae0: add      qword ptr [rax + 0x20], r12
0xdae4: mov      qword ptr [rax + 0x40], r12
0xdae8: jmp      0xd9e7
0xd9e7:
0xdaed: nop      dword ptr [rax]
0xdaf0: mov      rax, qword ptr [rip + 0x1d911]
0xdaf7: mov      rsi, qword ptr [rax + 0x58]
0xdafb: mov      edx, dword ptr [rax + 0x40]
0xdafe: mov      rdi, qword ptr [rax]
0xdb01: call     0xc600
0xdb06: jmp      0xd9ae
0xd9ae:
0xdb0b: call     0x49d0
0xdb10: endbr64
0xdb14: mov      rax, qword ptr [rip + 0x1d8ed]
0xdb1b: push     rbx
0xdb1c: mov      rbx, rdi
0xdb1f: mov      rdi, qword ptr [rdi]
0xdb22: mov      rsi, qword ptr [rax + 0x48]
0xdb26: add      rsi, 1
0xdb2a: call     0x1d690
0xdb2f: mov      qword ptr [rbx], rax

```

```

0xdb32: mov     rdi, rax
0xdb35: mov     rax, qword ptr [rip + 0x1d8cc]
0xdb3c: mov     rcx, qword ptr [rax + 0x48]
0xdb40: mov     rsi, qword ptr [rax + 0x40]
0xdb44: add     rsi, qword ptr [rax + 0x58]
0xdb48: lea     rdx, [rcx + 1]
0xdb4c: call    0x4bb0
0xdb51: mov     rax, qword ptr [rip + 0x1d8b0]
0xdb58: pop     rbx
0xdb59: mov     eax, dword ptr [rax + 0x48]
0xdb5c: ret
0xdb5d: nop     dword ptr [rax]
0xdb60: endbr64
0xdb64: push    rbp
0xdb65: xor     eax, eax
0xdb67: mov     rbp, rdi
0xdb6a: call    0xcff0
0xdb6f: test    rbp, rbp
0xdb72: lea     rax, [rip + 0x133bf]
0xdb79: cmov     rbp, rax
0xdb7d: mov     rdi, rbp
0xdb80: call    0xd540
0xdb85: xor     eax, eax
0xdb87: call    0xd870
0xdb8c: xor     eax, eax
0xdb8e: pop     rbp
0xdb8f: jmp     0xd740
0xd740:
0xdb94: nop     word ptr cs:[rax + rax]
0xdb9f: nop
0xdba0: endbr64
0xdba4: mov     rax, qword ptr [rip + 0x1d85d]
0xdbab: mov     dword ptr [rax + 0xc], edi
0xdbae: ret
0xdbaf: nop
0xdbb0: endbr64
0xdbb4: push    r12
0xdbb6: push    rbp
0xdbb7: push    rbx
0xdbb8: cmp     edi, 1
0xdbbb: je      0xdbd0
0xdbbd: cmp     edi, 2
0xdbc0: je      0xdc90
0xdbc6: test    edi, edi
0xdbc8: je      0xdc40
0xdbca: pop     rbx
0xdbcb: pop     rbp
0xdbcc: pop     r12

```

```

0xdbce: ret
0xdbcf: nop
0xdbd0: mov     ecx, dword ptr [rip + 0x1d83e]
0xdbd6: test    ecx, ecx
0xdbd8: je      0xdd38
0xdbde: xor     eax, eax
0xdbe0: call    0x1ab10
0xdbe5: mov     rbp, rax
0xdbe8: test    rax, rax
0xdbeb: je      0xdbca
0xdbed: mov     r12, qword ptr [rbp]
0xdbf1: mov     rbx, qword ptr [rip + 0x1d810]
0xdbf8: mov     rdi, r12
0xdbfb: call    0x49b0
0xdc00: mov     rdi, qword ptr [rbx + 0x40]
0xdc04: mov     qword ptr [rbx + 0x48], rax
0xdc08: lea     rsi, [rax + rdi]
0xdc0c: mov     qword ptr [rbx + 0x38], rsi
0xdc10: mov     qword ptr [rbx + 0x18], rsi
0xdc14: add     rsi, 1
0xdc18: cmp     rsi, qword ptr [rbx + 0x50]
0xdc1c: ja      0xdd70
0xdc22: add     rdi, qword ptr [rbx + 0x58]
0xdc26: mov     rsi, r12
0xdc29: call    0x48b0
0xdc2e: pop     rbx
0xdc2f: xor     eax, eax
0xdc31: pop     rbp
0xdc32: pop     r12
0xdc34: jmp     0xd870
0xd870:
0xdc39: nop     dword ptr [rax]
0xdc40: mov     eax, dword ptr [rip + 0x1d7ce]
0xdc46: test    eax, eax
0xdc48: jne     0xdd18
0xdc4e: xor     eax, eax
0xdc50: call    0x1ab10
0xdc55: test    rax, rax
0xdc58: je      0xdd98
0xdc5e: mov     rdx, qword ptr [rip + 0x1d7a3]
0xdc65: mov     rsi, qword ptr [rax]
0xdc68: mov     rbp, qword ptr [rdx + 0x40]
0xdc6c: add     rbp, qword ptr [rdx + 0x58]
0xdc70: mov     rdi, rbp
0xdc73: call    0x4b30
0xdc78: test    eax, eax
0xdc7a: jne     0xdd58
0xdc80: pop     rbx

```

```

0xdc81: xor     eax, eax
0xdc83: pop     rbp
0xdc84: pop     r12
0xdc86: jmp     0x1ab40
0x1ab40
0xdc8b: nop     dword ptr [rax + rax]
0xdc90: mov     edx, dword ptr [rip + 0x1d77e]
0xdc96: test    edx, edx
0xdc98: je      0xdcf8
0xdc9a: xor     eax, eax
0xdc9c: call    0x1ab40
0xdca1: mov     rbp, rax
0xdca4: test    rax, rax
0xdca7: jne     0xdbed
0xdcad: mov     rax, qword ptr [rip + 0x1d754]
0xdcb4: mov     edi, 2
0xdcb9: mov     rdx, qword ptr [rax + 0x40]
0xdcbd: mov     dword ptr [rax + 0x10], 6
0xdcc4: mov     qword ptr [rax + 0x20], rdx
0xdcc8: mov     rdx, qword ptr [rax + 0x38]
0xdccc: mov     qword ptr [rax + 0x18], rdx
0xdcd0: xor     eax, eax
0xdcd2: call    0xc6e0
0xdcd7: mov     rax, qword ptr [rip + 0x1d72a]
0xdcde: mov     rdx, qword ptr [rax + 0x18]
0xdce2: mov     dword ptr [rax + 0x10], 0xd
0xdce9: mov     qword ptr [rax + 0x20], rdx
0xdced: jmp     0xdc2e
0xdc2e
0xdcf2: nop     word ptr [rax + rax]
0xdcf8: mov     dword ptr [rip + 0x1d712], 1
0xdd02: xor     eax, eax
0xdd04: call    0x1a570
0xdd09: mov     dword ptr [rip + 0x1d701], eax
0xdd0f: jmp     0xdc9a
0xdc9a
0xdd11: nop     dword ptr [rax]
0xdd18: mov     edi, dword ptr [rip + 0x1d6f2]
0xdd1e: xor     eax, eax
0xdd20: call    0x1ac30
0xdd25: mov     dword ptr [rip + 0x1d6e5], 0
0xdd2f: jmp     0xdc4e
0xdc4e
0xdd34: nop     dword ptr [rax]
0xdd38: mov     dword ptr [rip + 0x1d6d2], 1
0xdd42: xor     eax, eax
0xdd44: call    0x1a570
0xdd49: mov     dword ptr [rip + 0x1d6c1], eax

```

```

0xdd4f: jmp     0xdbde
0xdbde
0xdd54: nop     dword ptr [rax]
0xdd58: xor     eax, eax
0xdd5a: mov     rdi, rbp
0xdd5d: call    0x1a380
0xdd62: xor     eax, eax
0xdd64: call    0x1ab40
0xdd69: jmp     0xdc80
0xdc80
0xdd6e: nop
0xdd70: mov     qword ptr [rbx + 0x50], rsi
0xdd74: mov     rdi, qword ptr [rbx + 0x58]
0xdd78: call    0x1d690
0xdd7d: mov     qword ptr [rbx + 0x58], rax
0xdd81: mov     rbx, qword ptr [rip + 0x1d680]
0xdd88: mov     r12, qword ptr [rbp]
0xdd8c: mov     rdi, qword ptr [rbx + 0x40]
0xdd90: jmp     0xdc22
0xdc22
0xdd95: nop     dword ptr [rax]
0xdd98: mov     rax, qword ptr [rip + 0x1d669]
0xdd9f: mov     rdi, qword ptr [rax + 0x40]
0xdda3: add     rdi, qword ptr [rax + 0x58]
0xdda7: xor     eax, eax
0xdda9: call    0x1a380
0xddae: jmp     0xdc80
0xdc80
0xddb3: nop     word ptr cs:[rax + rax]
0xddbd: nop     dword ptr [rax]
0dddc0: endbr64
0dddc4: mov     eax, dword ptr [rip + 0x1be96]
0xddca: push    rbp
0ddccb: mov     rbp, rdi
0ddcce: lea     rdi, [rip + 0x1be73]
0ddd5: test    eax, eax
0ddd7: lea     rax, [rip + 0x1be52]
0dddde: cmove   rdi, rax
0dde2: call    0x17c50
0dde7: mov     edx, 9
0ddeec: lea     rsi, [rip + 0x1c4ad]
0ddf3: lea     rdi, [rip + 0x1c466]
0dddfa: call    0x17f70
0dddff: mov     rdi, rbp
0xde02: call    0x1d7a0
0xde07: mov     rdi, rax
0xde0a: mov     rbp, rax
0xde0d: mov     qword ptr [rip + 0x1d794], rax

```

```

0xde14: call    0x49b0
0xde19: mov     rdi, rbp
0xde1c: mov     rsi, rax
0xde1f: call    0x184d0
0xde24: xor     eax, eax
0xde26: call    0xc590
0xde2b: lea     rdi, [rip + 0x1d5ee]
0xde32: pop     rbp
0xde33: mov     qword ptr [rip + 0x1d78e], rax
0xde3a: jmp     0x4a30
0x4a30:
0xde3f: nop
0xde40: endbr64
0xde44: mov     rdi, qword ptr [rip + 0x1d77d]
0xde4b: jmp     0xc5c0
0xc5c0:
0xde50: endbr64
0xde54: push    rbp
0xde55: mov     rbp, rdi
0xde58: push    rbx
0xde59: mov     rbx, rsi
0xde5c: sub     rsp, 8
0xde60: mov     rdi, qword ptr [rip + 0x1d749]
0xde67: test    rdi, rdi
0xde6a: je      0xde71
0xde6c: call    0x18d10
0xde71: mov     rdi, rbp
0xde74: call    0x1d670
0xde79: mov     rdi, qword ptr [rip + 0x1d748]
0xde80: mov     edx, ebx
0xde82: xor     esi, esi
0xde84: mov     qword ptr [rip + 0x1d725], rax
0xde8b: add     rsp, 8
0xde8f: mov     r8d, ebp
0xde92: mov     ecx, 1
0xde97: pop     rbx
0xde98: pop     rbp
0xde99: jmp     0xc5e0
0xc5e0:
0xde9e: nop
0xdea0: endbr64
0xdea4: push    rbp
0xdea5: mov     rbp, rsp
0xdea8: push    r15
0xdeaa: push    r14
0xdeac: push    r13
0xdeaе: push    r12
0xdeb0: push    rbx

```

```

0xdeb1: sub    rsp, 0x68
0xdeb5: mov     rax, qword ptr fs:[0x28]
0xdebe: mov     qword ptr [rbp - 0x38], rax
0xdec2: xor     eax, eax
0xdec4: lea     rax, [rbp - 0x61]
0xdec8: mov     rdi, rax
0xdecb: mov     qword ptr [rbp - 0x80], rax
0xdecf: call    0xa6a0
0xded4: mov     rax, qword ptr [rip + 0x1d6ed]
0xdedb: mov     esi, 0x20
0xdee0: mov     rdi, qword ptr [rip + 0x1d6c9]
0xdee7: movsxd  rdx, dword ptr [rax + 0xc]
0xdeeb: call    0x4a50
0xdef0: mov     rdi, qword ptr [rip + 0x1d6c9]
0xdef7: call    0x49b0
0xdefc: lea     r14, [rax + 1]
0xdf00: mov     rdi, r14
0xdf03: call    0x1d670
0xdf08: mov     rcx, qword ptr [rip + 0x1d6b1]
0xdf0f: mov     r13, rax
0xdf12: movzx   eax, byte ptr [rcx]
0xdf15: test    al, al
0xdf17: je      0xe430
0xdf1d: xor     esi, esi
0xdf1f: xor     ebx, ebx
0xdf21: lea     r15, [rip + 0x13188]
0xdf28: xor     edx, edx
0xdf2a: mov     dword ptr [rbp - 0x74], esi
0xdf2d: jmp     0xdf48
0xdf48:
0xdf2f: nop
0xdf30: cmp     al, 9
0xdf32: je      0xdf80
0xdf34: add     dword ptr [rbp - 0x74], 1
0xdf38: mov     byte ptr [r13 + r12], al
0xdf3d: movsxd  rdx, ebx
0xdf40: movzx   eax, byte ptr [rcx + rdx]
0xdf44: test    al, al
0xdf46: je      0xdfc0
0xdf48: movsxd  r12, dword ptr [rbp - 0x74]
0xdf4c: add     ebx, 1
0xdf4f: cmp     al, 0x5c
0xdf51: jne     0xdf30
0xdf53: lea     rdx, [rcx + rdx + 1]
0xdf58: movzx   eax, byte ptr [rdx]
0xdf5b: test    al, al
0xdf5d: je      0xe320
0xdf63: sub     eax, 0x5c

```

```

0xdf66: cmp     al, 0x17
0xdf68: ja      0xe0d8
0xdf6e: movzx   eax, al
0xdf71: movsxd  rax, dword ptr [r15 + rax*4]
0xdf75: add     rax, r15
0xdf78: jmp     rax
rax
->->->                                indirect JMP
0x1ee20: endbr64
0x1ee24: sub     rsp, 0x68
0x1ee28: mov     r8d, esi
0x1ee2b: pxor    xmm0, xmm0
0x1ee2f: mov     rsi, rdx
0x1ee32: mov     rax, qword ptr fs:[0x28]
0x1ee3b: mov     qword ptr [rsp + 0x58], rax
0x1ee40: xor     eax, eax
0x1ee42: movups  xmmword ptr [rsp + 0x34], xmm0
0x1ee47: mov     rdx, -1
0x1ee4e: mov     rcx, rsp
0x1ee51: movups  xmmword ptr [rsp + 0x44], xmm0
0x1ee56: movdqa  xmm2, xmmword ptr [rsp + 0x40]
0x1ee5c: mov     eax, dword ptr [rsp + 0x50]
0x1ee60: mov     dword ptr [rsp + 0x30], r8d
0x1ee65: movdqa  xmm1, xmmword ptr [rsp + 0x30]
0x1ee6b: mov     dword ptr [rsp + 0x20], eax
0x1ee6f: movaps  xmmword ptr [rsp], xmm1
0x1ee73: movaps  xmmword ptr [rsp + 0x10], xmm2
0x1ee78: call    0x1e9e0
0x1ee7d: mov     rdx, qword ptr [rsp + 0x58]
0x1ee82: sub     rdx, qword ptr fs:[0x28]
0x1ee8b: jne     0x1ee92
0x1ee8d: add     rsp, 0x68
0x1ee91: ret
0x1ee92: call    0x49d0
0x1ee97: nop     word ptr [rax + rax]
0x1eea0: endbr64
0x1eea4: sub     rsp, 0x68
0x1eea8: mov     r8d, esi
0x1eeab: pxor    xmm0, xmm0
0x1eeaf: mov     rsi, rdx
0x1eeb2: mov     rax, qword ptr fs:[0x28]
0x1eebb: mov     qword ptr [rsp + 0x58], rax
0x1eec0: xor     eax, eax
0x1eec2: movups  xmmword ptr [rsp + 0x34], xmm0
0x1eec7: mov     rdx, rcx
0x1eeca: mov     rcx, rsp
0x1eecd: movups  xmmword ptr [rsp + 0x44], xmm0
0x1eed2: movdqa  xmm2, xmmword ptr [rsp + 0x40]

```



```

0x1eed8:    mov     eax, dword ptr [rsp + 0x50]
0x1eedc:    mov     dword ptr [rsp + 0x30], r8d
0x1eee1:    movdqa  xmm1, xmmword ptr [rsp + 0x30]
0x1eee7:    mov     dword ptr [rsp + 0x20], eax
0x1eeeb:    movaps  xmmword ptr [rsp], xmm1
0x1eeef:    movaps  xmmword ptr [rsp + 0x10], xmm2
0x1eef4:    call    0x1e9e0
0x1eef9:    mov     rdx, qword ptr [rsp + 0x58]
0x1eefe:    sub     rdx, qword ptr fs:[0x28]
0x1ef07:    jne     0x1ef0e
0x1ef09:    add     rsp, 0x68
0x1ef0d:    ret
0x1ef0e:    call    0x49d0
0x1ef13:    nop     word ptr cs:[rax + rax]
0x1ef1e:    nop
0x1ef20:    endbr64
0x1ef24:    sub     rsp, 0x68
0x1ef28:    pxor    xmm0, xmm0
0x1ef2c:    mov     rdx, -1
0x1ef33:    mov     rax, qword ptr fs:[0x28]
0x1ef3c:    mov     qword ptr [rsp + 0x58], rax
0x1ef41:    xor     eax, eax
0x1ef43:    mov     dword ptr [rsp + 0x30], edi
0x1ef47:    mov     rcx, rsp
0x1ef4a:    xor     edi, edi
0x1ef4c:    movups  xmmword ptr [rsp + 0x34], xmm0
0x1ef51:    movdqa  xmm1, xmmword ptr [rsp + 0x30]
0x1ef57:    movups  xmmword ptr [rsp + 0x44], xmm0
0x1ef5c:    movdqa  xmm2, xmmword ptr [rsp + 0x40]
0x1ef62:    mov     eax, dword ptr [rsp + 0x50]
0x1ef66:    movaps  xmmword ptr [rsp], xmm1
0x1ef6a:    mov     dword ptr [rsp + 0x20], eax
0x1ef6e:    movaps  xmmword ptr [rsp + 0x10], xmm2
0x1ef73:    call    0x1e9e0
0x1ef78:    mov     rdx, qword ptr [rsp + 0x58]
0x1ef7d:    sub     rdx, qword ptr fs:[0x28]
0x1ef86:    jne     0x1ef8d
0x1ef88:    add     rsp, 0x68
0x1ef8c:    ret
0x1ef8d:    call    0x49d0
0x1ef92:    nop     word ptr cs:[rax + rax]
0x1ef9d:    nop     dword ptr [rax]
0x1efa0:    endbr64
0x1efa4:    sub     rsp, 0x38
0x1efa8:    movdqa  xmm0, xmmword ptr [rip + 0xccb0]
0x1efb0:    movdqa  xmm1, xmmword ptr [rip + 0xccb8]
0x1efb8:    mov     ecx, esi
0x1efba:    mov     rax, qword ptr fs:[0x28]

```

```

0x1efc3:    mov     qword ptr [rsp + 0x28], rax
0x1efc8:    xor     eax, eax
0x1efca:    mov     eax, dword ptr [rip + 0xccb0]
0x1efd0:    and     ecx, 0x1f
0x1efd3:    mov     r9, rsp
0x1efd6:    movaps  xmmword ptr [rsp], xmm0
0x1efda:    mov     dword ptr [rsp + 0x20], eax
0x1efde:    mov     eax, esi
0x1efe0:    shr     al, 5
0x1efe3:    movaps  xmmword ptr [rsp + 0x10], xmm1
0x1efe8:    movzx   eax, al
0x1efeb:    lea     rdx, [rsp + rax*4 + 4]
0x1eff0:    mov     esi, dword ptr [rdx]
0x1eff2:    mov     eax, esi
0x1eff4:    shr     eax, cl
0x1eff6:    not     eax
0x1eff8:    and     eax, 1
0x1effb:    shl     eax, cl
0x1effd:    mov     rcx, r9
0x1f000:    xor     eax, esi
0x1f002:    mov     rsi, rdi
0x1f005:    xor     edi, edi
0x1f007:    mov     dword ptr [rdx], eax
0x1f009:    mov     rdx, -1
0x1f010:    call    0x1e9e0
0x1f015:    mov     rdx, qword ptr [rsp + 0x28]
0x1f01a:    sub     rdx, qword ptr fs:[0x28]
0x1f023:    jne     0x1f02a
0x1f025:    add     rsp, 0x38
0x1f029:    ret
0x1f02a:    call    0x49d0
0x1f02f:    nop
0x1f030:    endbr64
0x1f034:    mov     esi, 0x3a
0x1f039:    jmp     0x1efa0
0x1efa0:
0x1e832:    mov     eax, dword ptr [rsp + 0x10]
0x1e836:    jbe     0x1e88a
0x1e838:    sub     eax, 5
0x1e83b:    cmp     eax, 1
0x1e83e:    ja      0x1e8c8
0x1e844:    cmp     dword ptr [rsp + 0x10], 6
0x1e849:    mov     edx, 0x60
0x1e84e:    mov     eax, 0x22
0x1e853:    cmovne  eax, edx
0x1e856:    lea     rcx, [rip + 0x2a3b]
0x1e85d:    lea     rdx, [rip + 0x3ef8]
0x1e864:    cmov     rdx, rcx

```

```

0x1e868:    mov     qword ptr [rsp + 0x20], rdx
0x1e86d:    test    r10, r10
0x1e870:    je      0x1e879
0x1e872:    mov     rsi, qword ptr [rsp + 8]
0x1e877:    mov     byte ptr [rsi], al
0x1e879:    mov     r8d, 1
0x1e87f:    mov     r9d, 1
0x1e885:    jmp     0x1e0b0
0x1e0b0:
0x1e88a:    cmp     eax, 2
0x1e88d:    je      0x1e39e
0x1e893:    cmp     eax, 3
0x1e896:    jne     0x1e8c8
0x1e898:    test    r10, r10
0x1e89b:    je      0x1e8dc
0x1e89d:    mov     rax, qword ptr [rsp + 8]
0x1e8a2:    mov     r8d, 1
0x1e8a8:    mov     r9d, 1
0x1e8ae:    mov     byte ptr [rax], 0x22
0x1e8b1:    lea     rax, [rip + 0x29e0]
0x1e8b8:    mov     qword ptr [rsp + 0x20], rax
0x1e8bd:    jmp     0x1e0b0
0x1e0b0:
0x1e8c2:    nop     word ptr [rax + rax]
0x1e8c8:    mov     qword ptr [rsp + 0x20], 0
0x1e8d1:    xor     r8d, r8d
0x1e8d4:    xor     r9d, r9d
0x1e8d7:    jmp     0x1e0b0
0x1e0b0:
0x1e8dc:    lea     rax, [rip + 0x29b5]
0x1e8e3:    mov     r8d, 1
0x1e8e9:    mov     r9d, 1
0x1e8ef:    mov     qword ptr [rsp + 0x20], rax
0x1e8f4:    jmp     0x1e0b0
0x1e0b0:
0x1e8f9:    cmp     r10, rbx
0x1e8fc:    jbe     0x1e907
0x1e8fe:    mov     rax, qword ptr [rsp + 8]
0x1e903:    mov     byte ptr [rax + rbx], 0x3f
0x1e907:    lea     rax, [rbx + 1]
0x1e90b:    cmp     r10, rax
0x1e90e:    jbe     0x1e91a
0x1e910:    mov     rax, qword ptr [rsp + 8]
0x1e915:    mov     byte ptr [rax + rbx + 1], 0x5c
0x1e91a:    lea     rax, [rbx + 2]
0x1e91e:    cmp     r10, rax
0x1e921:    jbe     0x1e92d
0x1e923:    mov     rax, qword ptr [rsp + 8]

```

```

0x1e928:    mov     byte ptr [rax + rbx + 2], 0x3f
0x1e92d:    add     rbx, 3
0x1e931:    lea     r15, [r12 + 3]
0x1e936:    jmp     0x1e1a4
0x1e1a4
0x1e93b:    xor     ecx, ecx
0x1e93d:    cmp     r13, r12
0x1e940:    ja      0x1e951
0x1e942:    jmp     0x1e9c2
0x1e9c2
0x1e944:    nop     dword ptr [rax]
0x1e948:    add     rcx, 1
0x1e94c:    cmp     rdx, rcx
0x1e94f:    je      0x1e957
0x1e951:    cmp     byte ptr [rsi + rcx], 0
0x1e955:    jne     0x1e948
0x1e957:    mov     eax, r8d
0x1e95a:    jmp     0x1e630
0x1e630
0x1e95f:    cmp     rax, 1
0x1e963:    je      0x1e5ec
0x1e969:    lea     rdx, [r11 + r14]
0x1e96d:    add     rsi, rax
0x1e970:    jmp     0x1e985
0x1e985
0x1e972:    nop     word ptr [rax + rax]
0x1e978:    add     rdx, 1
0x1e97c:    cmp     rsi, rdx
0x1e97f:    je      0x1e5ec
0x1e985:    movzx   eax, byte ptr [rdx]
0x1e988:    sub     eax, 0x5b
0x1e98b:    cmp     al, 0x21
0x1e98d:    ja      0x1e978
0x1e98f:    movabs  rdi, 0x20000002b
0x1e999:    bt      rdi, rax
0x1e99d:    jae     0x1e978
0x1e99f:    mov     qword ptr [rsp + 0x20], r10
0x1e9a4:    mov     qword ptr [rsp + 0x10], r11
0x1e9a9:    call    0x49a0
0x1e9ae:    mov     r11, qword ptr [rsp + 0x10]
0x1e9b3:    mov     r10, qword ptr [rsp + 0x20]
0x1e9b8:    mov     qword ptr [rsp + 0x28], rax
0x1e9bd:    jmp     0x1e39e
0x1e39e
0x1e9c2:    mov     eax, r8d
0x1e9c5:    jmp     0x1e299
0x1e299
0x1e9ca:    call    0x49d0

```

```

0x1e9cf:      mov     r12, r15
0x1e9d2:      jmp     0x1e1e1
0x1e1e1
0x1e9d7:      nop     word ptr [rax + rax]
0x1e9e0:      push    r15
0x1e9e2:      mov     r15d, edi
0x1e9e5:      push    r14
0x1e9e7:      push    r13
0x1e9e9:      mov     r13, rdx
0x1e9ec:      push    r12
0x1e9ee:      mov     r12, rsi
0x1e9f1:      push    rbp
0x1e9f2:      mov     rbp, rcx
0x1e9f5:      push    rbx
0x1e9f6:      sub     rsp, 0x28
0x1e9fa:      call    0x4860
0x1e9ff:      mov     r14, qword ptr [rip + 0xbe0a]
0x1ea06:      mov     rbx, rax
0x1ea09:      mov     eax, dword ptr [rax]
0x1ea0b:      mov     dword ptr [rsp + 0xc], eax
0x1ea0f:      test    r15d, r15d
0x1ea12:      js      0x4f2a
0x1ea18:      cmp     dword ptr [rip + 0xbde5], r15d
0x1ea1f:      mov     r10d, dword ptr [rsp + 0xc]
0x1ea24:      ja      0x1ea89
0x1ea26:      lea     edx, [r15 + 1]
0x1ea2a:      lea     rax, [rip + 0xbdef]
0x1ea31:      mov     rsi, rdx
0x1ea34:      mov     dword ptr [rsp + 0x10], edx
0x1ea38:      shl     rsi, 4
0x1ea3c:      cmp     r14, rax
0x1ea3f:      je      0x1eb40
0x1ea45:      mov     rdi, r14
0x1ea48:      mov     qword ptr [rsp + 0x18], rdx
0x1ea4d:      call    0x1d690
0x1ea52:      mov     rdx, qword ptr [rsp + 0x18]
0x1ea57:      mov     qword ptr [rip + 0xbdb2], rax
0x1ea5e:      mov     r14, rax
0x1ea61:      mov     edi, dword ptr [rip + 0xbd9d]
0x1ea67:      xor     esi, esi
0x1ea69:      sub     rdx, rdi
0x1ea6c:      shl     rdi, 4
0x1ea70:      shl     rdx, 4
0x1ea74:      add     rdi, r14
0x1ea77:      call    0x4a50
0x1ea7c:      mov     eax, dword ptr [rsp + 0x10]
0x1ea80:      mov     r10d, dword ptr [rbx]
0x1ea83:      mov     dword ptr [rip + 0xbd7b], eax

```

```

0x1ea89:      movsxd  r8, r15d
0x1ea8c:      mov     r9, rbp
0x1ea8f:      mov     rcx, r13
0x1ea92:      mov     rdx, r12
0x1ea95:      shl     r8, 4
0x1ea99:      mov     dword ptr [rsp + 0x18], r10d
0x1ea9e:      lea     r15, [r14 + r8]
0x1eaa2:      mov     r8d, dword ptr [rbp]
0x1eaa6:      mov     r14, qword ptr [r15 + 8]
0x1eaaa:      mov     rsi, qword ptr [r15]
0x1eaad:      mov     rdi, r14
0x1eab0:      mov     qword ptr [rsp + 0x10], rsi
0x1eab5:      call    0x1e040
0x1eaba:      mov     r10d, dword ptr [rsp + 0x18]
0x1eabf:      mov     rsi, qword ptr [rsp + 0x10]
0x1eac4:      mov     dword ptr [rbx], r10d
0x1eac7:      cmp     rax, rsi
0x1eaca:      jb      0x1eb1f
0x1eacc:      lea     rsi, [rax + 1]
0x1ead0:      lea     rax, [rip + 0xd089]
0x1ead7:      mov     qword ptr [r15], rsi
0x1eada:      cmp     r14, rax
0x1eadd:      je      0x1eaf1
0x1eadf:      mov     rdi, r14
0x1eae2:      mov     qword ptr [rsp + 0x10], rsi
0x1eae7:      call    0x4800
0x1eaec:      mov     rsi, qword ptr [rsp + 0x10]
0x1eaf1:      mov     rdi, rsi
0x1eaf4:      mov     qword ptr [rsp + 0x10], rsi
0x1eaf9:      call    0x1d670
0x1eafe:      mov     r8d, dword ptr [rbp]
0x1eb02:      mov     r9, rbp
0x1eb05:      mov     rcx, r13
0x1eb08:      mov     qword ptr [r15 + 8], rax
0x1eb0c:      mov     rsi, qword ptr [rsp + 0x10]
0x1eb11:      mov     rdx, r12
0x1eb14:      mov     rdi, rax
0x1eb17:      mov     r14, rax
0x1eb1a:      call    0x1e040
0x1eb1f:      mov     eax, dword ptr [rsp + 0xc]
0x1eb23:      mov     dword ptr [rbx], eax
0x1eb25:      add     rsp, 0x28
0x1eb29:      mov     rax, r14
0x1eb2c:      pop     rbx
0x1eb2d:      pop     rbp
0x1eb2e:      pop     r12
0x1eb30:      pop     r13
0x1eb32:      pop     r14

```

```

0x1eb34:    pop     r15
0x1eb36:    ret
0x1eb37:    nop     word ptr [rax + rax]
0x1eb40:    xor     edi, edi
0x1eb42:    mov     qword ptr [rsp + 0x18], rdx
0x1eb47:    call    0x1d690
0x1eb4c:    movdqa  xmm0, xmmword ptr [rip + 0xbccc]
0x1eb54:    mov     rdx, qword ptr [rsp + 0x18]
0x1eb59:    mov     qword ptr [rip + 0xbcb0], rax
0x1eb60:    mov     r14, rax
0x1eb63:    movups  xmmword ptr [rax], xmm0
0x1eb66:    jmp     0x1ea61
0x1ea61:
0x1eb6b:    nop     dword ptr [rax + rax]
0x1eb70:    endbr64
0x1eb74:    push    r12
0x1eb76:    push    rbp
0x1eb77:    mov     rbp, rdi
0x1eb7a:    push    rbx
0x1eb7b:    call    0x4860
0x1eb80:    test    rbp, rbp
0x1eb83:    mov     esi, 0x24
0x1eb88:    mov     r12d, dword ptr [rax]
0x1eb8b:    mov     rbx, rax
0x1eb8e:    lea     rax, [rip + 0xd0cb]
0x1eb95:    cmov    rbp, rax
0x1eb99:    mov     rdi, rbp
0x1eb9c:    call    0x1d760
0x1eba1:    mov     dword ptr [rbx], r12d
0x1eba4:    pop     rbx
0x1eba5:    pop     rbp
0x1eba6:    pop     r12
0x1eba8:    ret
0x1eba9:    nop     dword ptr [rax]
0x1ebb0:    endbr64
0x1ebb4:    test    rdi, rdi
0x1ebb7:    lea     rax, [rip + 0xd0a2]
0x1ebbe:    cmov    rdi, rax
0x1ebc2:    mov     eax, dword ptr [rdi]
0x1ebc4:    ret
0x1ebc5:    nop     word ptr cs:[rax + rax]
0x1ebd0:    endbr64
0x1ebd4:    test    rdi, rdi
0x1ebd7:    lea     rax, [rip + 0xd082]
0x1ebde:    cmov    rdi, rax
0x1ebe2:    mov     dword ptr [rdi], esi
0x1ebe4:    ret
0x1ebe5:    nop     word ptr cs:[rax + rax]

```

```

0x1ebf0:      endbr64
0x1ebf4:      test     rdi, rdi
0x1ebf7:      lea       rax, [rip + 0xd062]
0x1ebfe:      mov       ecx, esi
0x1ec00:      cmov     rdi, rax
0x1ec04:      mov       eax, esi
0x1ec06:      and       ecx, 0x1f
0x1ec09:      shr       al, 5
0x1ec0c:      movzx     eax, al
0x1ec0f:      lea       rsi, [rdi + rax*4 + 4]
0x1ec14:      mov       edi, dword ptr [rsi]
0x1ec16:      mov       eax, edi
0x1ec18:      shr       eax, cl
0x1ec1a:      xor       edx, eax
0x1ec1c:      and       eax, 1
0x1ec1f:      and       edx, 1
0x1ec22:      shl       edx, cl
0x1ec24:      xor       edx, edi
0x1ec26:      mov       dword ptr [rsi], edx
0x1ec28:      ret
0x1ec29:      nop       dword ptr [rax]
0x1ec30:      endbr64
0x1ec34:      push      r15
0x1ec36:      lea       rax, [rip + 0xd023]
0x1ec3d:      push      r14
0x1ec3f:      mov       r14, rdx
0x1ec42:      push      r13
0x1ec44:      mov       r13, rsi
0x1ec47:      push      r12
0x1ec49:      mov       r12, rdi
0x1ec4c:      push      rbp
0x1ec4d:      push      rbx
0x1ec4e:      mov       rbx, r8
0x1ec51:      sub       rsp, 0x18
0x1ec55:      test     r8, r8
0x1ec58:      cmov     rbx, rax
0x1ec5c:      mov       qword ptr [rsp + 8], rcx
0x1ec61:      call     0x4860
0x1ec66:      mov       rcx, qword ptr [rsp + 8]
0x1ec6b:      mov       rdx, r14
0x1ec6e:      mov       rsi, r13
0x1ec71:      mov       r15d, dword ptr [rax]
0x1ec74:      mov       r8d, dword ptr [rbx]
0x1ec77:      mov       rbp, rax
0x1ec7a:      mov       r9, rbx
0x1ec7d:      mov       rdi, r12
0x1ec80:      call     0x1e040
0x1ec85:      mov       dword ptr [rbp], r15d

```



```

0x1ec89:      add     rsp, 0x18
0x1ec8d:      pop     rbx
0x1ec8e:      pop     rbp
0x1ec8f:      pop     r12
0x1ec91:      pop     r13
0x1ec93:      pop     r14
0x1ec95:      pop     r15
0x1ec97:      ret
0x1ec98:      nop     dword ptr [rax + rax]
0x1eca0:      endbr64
0x1eca4:      push    r15
0x1eca6:      mov     r15, rdi
0x1eca9:      push    r14
0x1ecab:      mov     r14, rsi
0x1ecae:      push    r13
0x1ecb0:      push    r12
0x1ecb2:      push    rbp
0x1ecb3:      push    rbx
0x1ecb4:      mov     rbx, rdx
0x1ecb7:      sub     rsp, 0x18
0x1ecbb:      call    0x4860
0x1ecc0:      test    rbx, rbx
0x1ecc3:      mov     rcx, r14
0x1ecc6:      mov     rdx, r15
0x1ecc9:      mov     r12d, dword ptr [rax]
0x1eccc:      mov     rbp, rax
0x1eccf:      lea     rax, [rip + 0xcf8a]
0x1ecd6:      cmov     rbx, rax
0x1ecda:      xor     esi, esi
0x1ecdc:      xor     edi, edi
0x1ecde:      mov     r8d, dword ptr [rbx]
0x1ece1:      mov     r9, rbx
0x1ece4:      call    0x1e040
0x1ece9:      mov     dword ptr [rbp], r12d
0x1eced:      lea     r13, [rax + 1]
0x1ecf1:      mov     dword ptr [rsp + 0xc], r12d
0x1ecf6:      mov     rdi, r13
0x1ecf9:      call    0x1d670
0x1ecfe:      mov     r8d, dword ptr [rbx]
0x1ed01:      mov     r9, rbx
0x1ed04:      mov     rcx, r14
0x1ed07:      mov     rdx, r15
0x1ed0a:      mov     rsi, r13
0x1ed0d:      mov     rdi, rax
0x1ed10:      mov     r12, rax
0x1ed13:      call    0x1e040
0x1ed18:      mov     eax, dword ptr [rsp + 0xc]
0x1ed1c:      mov     dword ptr [rbp], eax

```

```

0x1ed1f:      add     rsp, 0x18
0x1ed23:      mov     rax, r12
0x1ed26:      pop     rbx
0x1ed27:      pop     rbp
0x1ed28:      pop     r12
0x1ed2a:      pop     r13
0x1ed2c:      pop     r14
0x1ed2e:      pop     r15
0x1ed30:      ret
0x1ed31:      nop     word ptr cs:[rax + rax]
0x1ed3c:      nop     dword ptr [rax]
0x1ed40:      endbr64
0x1ed44:      mov     eax, dword ptr [rip + 0xbaba]
0x1ed4a:      push    r12
0x1ed4c:      mov     r12, qword ptr [rip + 0xbabd]
0x1ed53:      push    rbp
0x1ed54:      push    rbx
0x1ed55:      cmp     eax, 1
0x1ed58:      jbe     0x1ed81
0x1ed5a:      sub     eax, 2
0x1ed5d:      lea     rbx, [r12 + 0x18]
0x1ed62:      shl     rax, 4
0x1ed66:      lea     rbp, [r12 + rax + 0x28]
0x1ed6b:      nop     dword ptr [rax + rax]
0x1ed70:      mov     rdi, qword ptr [rbx]
0x1ed73:      add     rbx, 0x10
0x1ed77:      call    0x4800
0x1ed7c:      cmp     rbx, rbp
0x1ed7f:      jne     0x1ed70
0x1ed81:      mov     rdi, qword ptr [r12 + 8]
0x1ed86:      lea     rbx, [rip + 0xcdd3]
0x1ed8d:      cmp     rdi, rbx
0x1ed90:      je      0x1eda9
0x1ed92:      call    0x4800
0x1ed97:      mov     qword ptr [rip + 0xba8a], rbx
0x1ed9e:      mov     qword ptr [rip + 0xba77], 0x100
0x1eda9:      lea     rbx, [rip + 0xba70]
0x1edb0:      cmp     r12, rbx
0x1edb3:      je      0x1edc4
0x1edb5:      mov     rdi, r12
0x1edb8:      call    0x4800
0x1ebdb:      mov     qword ptr [rip + 0xba4c], rbx
0x1edc4:      mov     dword ptr [rip + 0xba36], 1
0x1edce:      pop     rbx
0x1edcf:      pop     rbp
0x1edd0:      pop     r12
0x1edd2:      ret
0x1edd3:      nop     word ptr cs:[rax + rax]

```

```

0x1edde:      nop
0x1ede0:      endbr64
0x1ede4:      lea     rcx, [rip + 0xce75]
0x1edeb:      mov     rdx, -1
0x1edf2:      jmp     0x1e9e0
0x1e9e0
0x1edf7:      nop     word ptr [rax + rax]
0x1ee00:      endbr64
0x1ee04:      mov     rsi, rdi
0x1ee07:      lea     rcx, [rip + 0xce52]
0x1ee0e:      mov     rdx, -1
0x1ee15:      xor     edi, edi
0x1ee17:      jmp     0x1e9e0
0x1e9e0
0x1ee1c:      nop     dword ptr [rax]
0x1ee20:      endbr64
0x1ee24:      sub     rsp, 0x68
0x1ee28:      mov     r8d, esi
0x1ee2b:      pxor    xmm0, xmm0
0x1ee2f:      mov     rsi, rdx
0x1ee32:      mov     rax, qword ptr fs:[0x28]
0x1ee3b:      mov     qword ptr [rsp + 0x58], rax
0x1ee40:      xor     eax, eax
0x1ee42:      movups  xmmword ptr [rsp + 0x34], xmm0
0x1ee47:      mov     rdx, -1
0x1ee4e:      mov     rcx, rsp
0x1ee51:      movups  xmmword ptr [rsp + 0x44], xmm0
0x1ee56:      movdqa  xmm2, xmmword ptr [rsp + 0x40]
0x1ee5c:      mov     eax, dword ptr [rsp + 0x50]
0x1ee60:      mov     dword ptr [rsp + 0x30], r8d
0x1ee65:      movdqa  xmm1, xmmword ptr [rsp + 0x30]
0x1ee6b:      mov     dword ptr [rsp + 0x20], eax
0x1ee6f:      movaps  xmmword ptr [rsp], xmm1
0x1ee73:      movaps  xmmword ptr [rsp + 0x10], xmm2
0x1ee78:      call    0x1e9e0
0x1ee7d:      mov     rdx, qword ptr [rsp + 0x58]
0x1ee82:      sub     rdx, qword ptr fs:[0x28]
0x1ee8b:      jne     0x1ee92
0x1ee8d:      add     rsp, 0x68
0x1ee91:      ret
0x1ee92:      call    0x49d0
0x1ee97:      nop     word ptr [rax + rax]
0x1eea0:      endbr64
0x1eea4:      sub     rsp, 0x68
0x1eea8:      mov     r8d, esi
0x1eeab:      pxor    xmm0, xmm0
0x1eeaf:      mov     rsi, rdx
0x1eeb2:      mov     rax, qword ptr fs:[0x28]

```

```

0x1eebb:    mov     qword ptr [rsp + 0x58], rax
0x1eec0:    xor     eax, eax
0x1eec2:    movups  xmmword ptr [rsp + 0x34], xmm0
0x1eec7:    mov     rdx, rcx
0x1eeca:    mov     rcx, rsp
0x1eecd:    movups  xmmword ptr [rsp + 0x44], xmm0
0x1eed2:    movdqa  xmm2, xmmword ptr [rsp + 0x40]
0x1eed8:    mov     eax, dword ptr [rsp + 0x50]
0x1eedc:    mov     dword ptr [rsp + 0x30], r8d
0x1eee1:    movdqa  xmm1, xmmword ptr [rsp + 0x30]
0x1eee7:    mov     dword ptr [rsp + 0x20], eax
0x1eeeb:    movaps  xmmword ptr [rsp], xmm1
0x1eeef:    movaps  xmmword ptr [rsp + 0x10], xmm2
0x1eef4:    call    0x1e9e0
0x1eef9:    mov     rdx, qword ptr [rsp + 0x58]
0x1eefe:    sub     rdx, qword ptr fs:[0x28]
0x1ef07:    jne     0x1ef0e
0x1ef09:    add     rsp, 0x68
0x1ef0d:    ret
0x1ef0e:    call    0x49d0
0x1ef13:    nop     word ptr cs:[rax + rax]
0x1ef1e:    nop
0x1ef20:    endbr64
0x1ef24:    sub     rsp, 0x68
0x1ef28:    pxor    xmm0, xmm0
0x1ef2c:    mov     rdx, -1
0x1ef33:    mov     rax, qword ptr fs:[0x28]
0x1ef3c:    mov     qword ptr [rsp + 0x58], rax
0x1ef41:    xor     eax, eax
0x1ef43:    mov     dword ptr [rsp + 0x30], edi
0x1ef47:    mov     rcx, rsp
0x1ef4a:    xor     edi, edi
0x1ef4c:    movups  xmmword ptr [rsp + 0x34], xmm0
0x1ef51:    movdqa  xmm1, xmmword ptr [rsp + 0x30]
0x1ef57:    movups  xmmword ptr [rsp + 0x44], xmm0
0x1ef5c:    movdqa  xmm2, xmmword ptr [rsp + 0x40]
0x1ef62:    mov     eax, dword ptr [rsp + 0x50]
0x1ef66:    movaps  xmmword ptr [rsp], xmm1
0x1ef6a:    mov     dword ptr [rsp + 0x20], eax
0x1ef6e:    movaps  xmmword ptr [rsp + 0x10], xmm2
0x1ef73:    call    0x1e9e0
0x1ef78:    mov     rdx, qword ptr [rsp + 0x58]
0x1ef7d:    sub     rdx, qword ptr fs:[0x28]
0x1ef86:    jne     0x1ef8d
0x1ef88:    add     rsp, 0x68
0x1ef8c:    ret
0x1ef8d:    call    0x49d0
0x1ef92:    nop     word ptr cs:[rax + rax]

```

```

0x1ef9d:      nop      dword ptr [rax]
0x1efa0:      endbr64
0x1efa4:      sub      rsp, 0x38
0x1efa8:      movdqa   xmm0, xmmword ptr [rip + 0xccb0]
0x1efb0:      movdqa   xmm1, xmmword ptr [rip + 0xccb8]
0x1efb8:      mov      ecx, esi
0x1efba:      mov      rax, qword ptr fs:[0x28]
0x1efc3:      mov      qword ptr [rsp + 0x28], rax
0x1efc8:      xor      eax, eax
0x1efca:      mov      eax, dword ptr [rip + 0xccb0]
0x1efd0:      and      ecx, 0x1f
0x1efd3:      mov      r9, rsp
0x1efd6:      movaps   xmmword ptr [rsp], xmm0
0x1efda:      mov      dword ptr [rsp + 0x20], eax
0x1efde:      mov      eax, esi
0x1efe0:      shr      al, 5
0x1efe3:      movaps   xmmword ptr [rsp + 0x10], xmm1
0x1efe8:      movzx    eax, al
0x1efeb:      lea      rdx, [rsp + rax*4 + 4]
0x1eff0:      mov      esi, dword ptr [rdx]
0x1eff2:      mov      eax, esi
0x1eff4:      shr      eax, cl
0x1eff6:      not      eax
0x1eff8:      and      eax, 1
0x1effb:      shl      eax, cl
0x1effd:      mov      rcx, r9
0x1f000:      xor      eax, esi
0x1f002:      mov      rsi, rdi
0x1f005:      xor      edi, edi
0x1f007:      mov      dword ptr [rdx], eax
0x1f009:      mov      rdx, -1
0x1f010:      call     0x1e9e0
0x1f015:      mov      rdx, qword ptr [rsp + 0x28]
0x1f01a:      sub      rdx, qword ptr fs:[0x28]
0x1f023:      jne      0x1f02a
0x1f025:      add      rsp, 0x38
0x1f029:      ret
0x1f02a:      call     0x49d0
0x1f02f:      nop
0x1f030:      endbr64
0x1f034:      mov      esi, 0x3a
0x1f039:      jmp      0x1efa0
0x1efa0:
0x1e7ae:      test     r12, r12
0x1e7b1:      jne      0x1e7be
0x1e7b3:      cmp      dword ptr [rsp + 0x10], 1
0x1e7b8:      je       0x1e380
0x1e7be:      mov      rsi, qword ptr [rsp + 0x20]

```

```

0x1e7c3:    mov     rax, rbx
0x1e7c6:    test    rsi, rsi
0x1e7c9:    je      0x1e7f4
0x1e7cb:    movzx   edx, byte ptr [rsi]
0x1e7ce:    test    dl, dl
0x1e7d0:    je      0x1e7f4
0x1e7d2:    sub     rsi, rbx
0x1e7d5:    mov     rcx, rsi
0x1e7d8:    mov     rsi, qword ptr [rsp + 8]
0x1e7dd:    nop     dword ptr [rax]
0x1e7e0:    cmp     r10, rax
0x1e7e3:    jbe     0x1e7e8
0x1e7e5:    mov     byte ptr [rsi + rax], dl
0x1e7e8:    add     rax, 1
0x1e7ec:    movzx   edx, byte ptr [rcx + rax]
0x1e7f0:    test    dl, dl
0x1e7f2:    jne     0x1e7e0
0x1e7f4:    cmp     rax, r10
0x1e7f7:    jae     0x1e802
0x1e7f9:    mov     rsi, qword ptr [rsp + 8]
0x1e7fe:    mov     byte ptr [rsi + rax], 0
0x1e802:    mov     rdx, qword ptr [rsp + 0x78]
0x1e807:    sub     rdx, qword ptr fs:[0x28]
0x1e810:    jne     0x1e9ca
0x1e816:    add     rsp, 0x88
0x1e81d:    pop     rbx
0x1e81e:    pop     rbp
0x1e81f:    pop     r12
0x1e821:    pop     r13
0x1e823:    pop     r14
0x1e825:    pop     r15
0x1e827:    ret
0x1e828:    mov     eax, r8d
0x1e82b:    xor     ecx, ecx
0x1e82d:    jmp     0x1e299
0x1e299:
0x1e832:    mov     eax, dword ptr [rsp + 0x10]
0x1e836:    jbe     0x1e88a
0x1e838:    sub     eax, 5
0x1e83b:    cmp     eax, 1
0x1e83e:    ja      0x1e8c8
0x1e844:    cmp     dword ptr [rsp + 0x10], 6
0x1e849:    mov     edx, 0x60
0x1e84e:    mov     eax, 0x22
0x1e853:    cmovne  eax, edx
0x1e856:    lea     rcx, [rip + 0x2a3b]
0x1e85d:    lea     rdx, [rip + 0x3ef8]
0x1e864:    cmove   rdx, rcx

```

```

0x1e868:    mov     qword ptr [rsp + 0x20], rdx
0x1e86d:    test    r10, r10
0x1e870:    je      0x1e879
0x1e872:    mov     rsi, qword ptr [rsp + 8]
0x1e877:    mov     byte ptr [rsi], al
0x1e879:    mov     r8d, 1
0x1e87f:    mov     r9d, 1
0x1e885:    jmp     0x1e0b0
0x1e0b0:
0x1e88a:    cmp     eax, 2
0x1e88d:    je      0x1e39e
0x1e893:    cmp     eax, 3
0x1e896:    jne     0x1e8c8
0x1e898:    test    r10, r10
0x1e89b:    je      0x1e8dc
0x1e89d:    mov     rax, qword ptr [rsp + 8]
0x1e8a2:    mov     r8d, 1
0x1e8a8:    mov     r9d, 1
0x1e8ae:    mov     byte ptr [rax], 0x22
0x1e8b1:    lea     rax, [rip + 0x29e0]
0x1e8b8:    mov     qword ptr [rsp + 0x20], rax
0x1e8bd:    jmp     0x1e0b0
0x1e0b0:
0x1e8c2:    nop     word ptr [rax + rax]
0x1e8c8:    mov     qword ptr [rsp + 0x20], 0
0x1e8d1:    xor     r8d, r8d
0x1e8d4:    xor     r9d, r9d
0x1e8d7:    jmp     0x1e0b0
0x1e0b0:
0x1e8dc:    lea     rax, [rip + 0x29b5]
0x1e8e3:    mov     r8d, 1
0x1e8e9:    mov     r9d, 1
0x1e8ef:    mov     qword ptr [rsp + 0x20], rax
0x1e8f4:    jmp     0x1e0b0
0x1e0b0:
0x1e8f9:    cmp     r10, rbx
0x1e8fc:    jbe     0x1e907
0x1e8fe:    mov     rax, qword ptr [rsp + 8]
0x1e903:    mov     byte ptr [rax + rbx], 0x3f
0x1e907:    lea     rax, [rbx + 1]
0x1e90b:    cmp     r10, rax
0x1e90e:    jbe     0x1e91a
0x1e910:    mov     rax, qword ptr [rsp + 8]
0x1e915:    mov     byte ptr [rax + rbx + 1], 0x5c
0x1e91a:    lea     rax, [rbx + 2]
0x1e91e:    cmp     r10, rax
0x1e921:    jbe     0x1e92d
0x1e923:    mov     rax, qword ptr [rsp + 8]

```

```

0x1e928:      mov     byte ptr [rax + rbx + 2], 0x3f
0x1e92d:      add     rbx, 3
0x1e931:      lea     r15, [r12 + 3]
0x1e936:      jmp     0x1e1a4
0x1e1a4
0x1e93b:      xor     ecx, ecx
0x1e93d:      cmp     r13, r12
0x1e940:      ja      0x1e951
0x1e942:      jmp     0x1e9c2
0x1e9c2
0x1e944:      nop     dword ptr [rax]
0x1e948:      add     rcx, 1
0x1e94c:      cmp     rdx, rcx
0x1e94f:      je      0x1e957
0x1e951:      cmp     byte ptr [rsi + rcx], 0
0x1e955:      jne     0x1e948
0x1e957:      mov     eax, r8d
0x1e95a:      jmp     0x1e630
0x1e630
0x1e95f:      cmp     rax, 1
0x1e963:      je      0x1e5ec
0x1e969:      lea     rdx, [r11 + r14]
0x1e96d:      add     rsi, rax
0x1e970:      jmp     0x1e985
0x1e985
0x1e972:      nop     word ptr [rax + rax]
0x1e978:      add     rdx, 1
0x1e97c:      cmp     rsi, rdx
0x1e97f:      je      0x1e5ec
0x1e985:      movzx   eax, byte ptr [rdx]
0x1e988:      sub     eax, 0x5b
0x1e98b:      cmp     al, 0x21
0x1e98d:      ja      0x1e978
0x1e98f:      movabs  rdi, 0x20000002b
0x1e999:      bt      rdi, rax
0x1e99d:      jae     0x1e978
0x1e99f:      mov     qword ptr [rsp + 0x20], r10
0x1e9a4:      mov     qword ptr [rsp + 0x10], r11
0x1e9a9:      call    0x49a0
0x1e9ae:      mov     r11, qword ptr [rsp + 0x10]
0x1e9b3:      mov     r10, qword ptr [rsp + 0x20]
0x1e9b8:      mov     qword ptr [rsp + 0x28], rax
0x1e9bd:      jmp     0x1e39e
0x1e39e
0x1e9c2:      mov     eax, r8d
0x1e9c5:      jmp     0x1e299
0x1e299
0x1e9ca:      call    0x49d0

```



```

0x1e9cf:      mov     r12, r15
0x1e9d2:      jmp     0x1e1e1
0x1e1e1
0x1e9d7:      nop     word ptr [rax + rax]
0x1e9e0:      push    r15
0x1e9e2:      mov     r15d, edi
0x1e9e5:      push    r14
0x1e9e7:      push    r13
0x1e9e9:      mov     r13, rdx
0x1e9ec:      push    r12
0x1e9ee:      mov     r12, rsi
0x1e9f1:      push    rbp
0x1e9f2:      mov     rbp, rcx
0x1e9f5:      push    rbx
0x1e9f6:      sub     rsp, 0x28
0x1e9fa:      call    0x4860
0x1e9ff:      mov     r14, qword ptr [rip + 0xbe0a]
0x1ea06:      mov     rbx, rax
0x1ea09:      mov     eax, dword ptr [rax]
0x1ea0b:      mov     dword ptr [rsp + 0xc], eax
0x1ea0f:      test    r15d, r15d
0x1ea12:      js      0x4f2a
0x1ea18:      cmp     dword ptr [rip + 0xbde5], r15d
0x1ea1f:      mov     r10d, dword ptr [rsp + 0xc]
0x1ea24:      ja      0x1ea89
0x1ea26:      lea     edx, [r15 + 1]
0x1ea2a:      lea     rax, [rip + 0xbdef]
0x1ea31:      mov     rsi, rdx
0x1ea34:      mov     dword ptr [rsp + 0x10], edx
0x1ea38:      shl     rsi, 4
0x1ea3c:      cmp     r14, rax
0x1ea3f:      je      0x1eb40
0x1ea45:      mov     rdi, r14
0x1ea48:      mov     qword ptr [rsp + 0x18], rdx
0x1ea4d:      call    0x1d690
0x1ea52:      mov     rdx, qword ptr [rsp + 0x18]
0x1ea57:      mov     qword ptr [rip + 0xbdb2], rax
0x1ea5e:      mov     r14, rax
0x1ea61:      mov     edi, dword ptr [rip + 0xbd9d]
0x1ea67:      xor     esi, esi
0x1ea69:      sub     rdx, rdi
0x1ea6c:      shl     rdi, 4
0x1ea70:      shl     rdx, 4
0x1ea74:      add     rdi, r14
0x1ea77:      call    0x4a50
0x1ea7c:      mov     eax, dword ptr [rsp + 0x10]
0x1ea80:      mov     r10d, dword ptr [rbx]
0x1ea83:      mov     dword ptr [rip + 0xbd7b], eax

```

```

0x1ea89:      movsxd  r8, r15d
0x1ea8c:      mov     r9, rbp
0x1ea8f:      mov     rcx, r13
0x1ea92:      mov     rdx, r12
0x1ea95:      shl     r8, 4
0x1ea99:      mov     dword ptr [rsp + 0x18], r10d
0x1ea9e:      lea     r15, [r14 + r8]
0x1eaa2:      mov     r8d, dword ptr [rbp]
0x1eaa6:      mov     r14, qword ptr [r15 + 8]
0x1eaaa:      mov     rsi, qword ptr [r15]
0x1eaad:      mov     rdi, r14
0x1eab0:      mov     qword ptr [rsp + 0x10], rsi
0x1eab5:      call    0x1e040
0x1eaba:      mov     r10d, dword ptr [rsp + 0x18]
0x1eabf:      mov     rsi, qword ptr [rsp + 0x10]
0x1eac4:      mov     dword ptr [rbx], r10d
0x1eac7:      cmp     rax, rsi
0x1eaca:      jb      0x1eb1f
0x1eacc:      lea     rsi, [rax + 1]
0x1ead0:      lea     rax, [rip + 0xd089]
0x1ead7:      mov     qword ptr [r15], rsi
0x1eada:      cmp     r14, rax
0x1eadd:      je      0x1eaf1
0x1eadf:      mov     rdi, r14
0x1eae2:      mov     qword ptr [rsp + 0x10], rsi
0x1eae7:      call    0x4800
0x1eaec:      mov     rsi, qword ptr [rsp + 0x10]
0x1eaf1:      mov     rdi, rsi
0x1eaf4:      mov     qword ptr [rsp + 0x10], rsi
0x1eaf9:      call    0x1d670
0x1eafe:      mov     r8d, dword ptr [rbp]
0x1eb02:      mov     r9, rbp
0x1eb05:      mov     rcx, r13
0x1eb08:      mov     qword ptr [r15 + 8], rax
0x1eb0c:      mov     rsi, qword ptr [rsp + 0x10]
0x1eb11:      mov     rdx, r12
0x1eb14:      mov     rdi, rax
0x1eb17:      mov     r14, rax
0x1eb1a:      call    0x1e040
0x1eb1f:      mov     eax, dword ptr [rsp + 0xc]
0x1eb23:      mov     dword ptr [rbx], eax
0x1eb25:      add     rsp, 0x28
0x1eb29:      mov     rax, r14
0x1eb2c:      pop     rbx
0x1eb2d:      pop     rbp
0x1eb2e:      pop     r12
0x1eb30:      pop     r13
0x1eb32:      pop     r14

```

```

0x1eb34:    pop     r15
0x1eb36:    ret
0x1eb37:    nop     word ptr [rax + rax]
0x1eb40:    xor     edi, edi
0x1eb42:    mov     qword ptr [rsp + 0x18], rdx
0x1eb47:    call    0x1d690
0x1eb4c:    movdqa  xmm0, xmmword ptr [rip + 0xbccc]
0x1eb54:    mov     rdx, qword ptr [rsp + 0x18]
0x1eb59:    mov     qword ptr [rip + 0xbcb0], rax
0x1eb60:    mov     r14, rax
0x1eb63:    movups  xmmword ptr [rax], xmm0
0x1eb66:    jmp     0x1ea61
0x1ea61:
0x1eb6b:    nop     dword ptr [rax + rax]
0x1eb70:    endbr64
0x1eb74:    push    r12
0x1eb76:    push    rbp
0x1eb77:    mov     rbp, rdi
0x1eb7a:    push    rbx
0x1eb7b:    call    0x4860
0x1eb80:    test    rbp, rbp
0x1eb83:    mov     esi, 0x24
0x1eb88:    mov     r12d, dword ptr [rax]
0x1eb8b:    mov     rbx, rax
0x1eb8e:    lea     rax, [rip + 0xd0cb]
0x1eb95:    cmov     rbp, rax
0x1eb99:    mov     rdi, rbp
0x1eb9c:    call    0x1d760
0x1eba1:    mov     dword ptr [rbx], r12d
0x1eba4:    pop     rbx
0x1eba5:    pop     rbp
0x1eba6:    pop     r12
0x1eba8:    ret
0x1eba9:    nop     dword ptr [rax]
0x1ebb0:    endbr64
0x1ebb4:    test    rdi, rdi
0x1ebb7:    lea     rax, [rip + 0xd0a2]
0x1ebbe:    cmov     rdi, rax
0x1ebc2:    mov     eax, dword ptr [rdi]
0x1ebc4:    ret
0x1ebc5:    nop     word ptr cs:[rax + rax]
0x1ebd0:    endbr64
0x1ebd4:    test    rdi, rdi
0x1ebd7:    lea     rax, [rip + 0xd082]
0x1ebde:    cmov     rdi, rax
0x1ebe2:    mov     dword ptr [rdi], esi
0x1ebe4:    ret
0x1ebe5:    nop     word ptr cs:[rax + rax]

```

```

0x1ebf0:      endbr64
0x1ebf4:      test     rdi, rdi
0x1ebf7:      lea      rax, [rip + 0xd062]
0x1ebfe:      mov      ecx, esi
0x1ec00:      cmov     rdi, rax
0x1ec04:      mov      eax, esi
0x1ec06:      and      ecx, 0x1f
0x1ec09:      shr      al, 5
0x1ec0c:      movzx    eax, al
0x1ec0f:      lea      rsi, [rdi + rax*4 + 4]
0x1ec14:      mov      edi, dword ptr [rsi]
0x1ec16:      mov      eax, edi
0x1ec18:      shr      eax, cl
0x1ec1a:      xor      edx, eax
0x1ec1c:      and      eax, 1
0x1ec1f:      and      edx, 1
0x1ec22:      shl      edx, cl
0x1ec24:      xor      edx, edi
0x1ec26:      mov      dword ptr [rsi], edx
0x1ec28:      ret
0x1ec29:      nop      dword ptr [rax]
0x1ec30:      endbr64
0x1ec34:      push     r15
0x1ec36:      lea      rax, [rip + 0xd023]
0x1ec3d:      push     r14
0x1ec3f:      mov      r14, rdx
0x1ec42:      push     r13
0x1ec44:      mov      r13, rsi
0x1ec47:      push     r12
0x1ec49:      mov      r12, rdi
0x1ec4c:      push     rbp
0x1ec4d:      push     rbx
0x1ec4e:      mov      rbx, r8
0x1ec51:      sub      rsp, 0x18
0x1ec55:      test     r8, r8
0x1ec58:      cmov     rbx, rax
0x1ec5c:      mov      qword ptr [rsp + 8], rcx
0x1ec61:      call     0x4860
0x1ec66:      mov      rcx, qword ptr [rsp + 8]
0x1ec6b:      mov      rdx, r14
0x1ec6e:      mov      rsi, r13
0x1ec71:      mov      r15d, dword ptr [rax]
0x1ec74:      mov      r8d, dword ptr [rbx]
0x1ec77:      mov      rbp, rax
0x1ec7a:      mov      r9, rbx
0x1ec7d:      mov      rdi, r12
0x1ec80:      call     0x1e040
0x1ec85:      mov      dword ptr [rbp], r15d

```

```

0x1ec89:      add     rsp, 0x18
0x1ec8d:      pop     rbx
0x1ec8e:      pop     rbp
0x1ec8f:      pop     r12
0x1ec91:      pop     r13
0x1ec93:      pop     r14
0x1ec95:      pop     r15
0x1ec97:      ret
0x1ec98:      nop     dword ptr [rax + rax]
0x1eca0:      endbr64
0x1eca4:      push    r15
0x1eca6:      mov     r15, rdi
0x1eca9:      push    r14
0x1ecab:      mov     r14, rsi
0x1ecae:      push    r13
0x1ecb0:      push    r12
0x1ecb2:      push    rbp
0x1ecb3:      push    rbx
0x1ecb4:      mov     rbx, rdx
0x1ecb7:      sub     rsp, 0x18
0x1ecbb:      call    0x4860
0x1ecc0:      test    rbx, rbx
0x1ecc3:      mov     rcx, r14
0x1ecc6:      mov     rdx, r15
0x1ecc9:      mov     r12d, dword ptr [rax]
0x1eccc:      mov     rbp, rax
0x1eccf:      lea     rax, [rip + 0xcf8a]
0x1ecd6:      cmov     rbx, rax
0x1ecda:      xor     esi, esi
0x1ecdc:      xor     edi, edi
0x1ecde:      mov     r8d, dword ptr [rbx]
0x1ece1:      mov     r9, rbx
0x1ece4:      call    0x1e040
0x1ece9:      mov     dword ptr [rbp], r12d
0x1eced:      lea     r13, [rax + 1]
0x1ecf1:      mov     dword ptr [rsp + 0xc], r12d
0x1ecf6:      mov     rdi, r13
0x1ecf9:      call    0x1d670
0x1ecfe:      mov     r8d, dword ptr [rbx]
0x1ed01:      mov     r9, rbx
0x1ed04:      mov     rcx, r14
0x1ed07:      mov     rdx, r15
0x1ed0a:      mov     rsi, r13
0x1ed0d:      mov     rdi, rax
0x1ed10:      mov     r12, rax
0x1ed13:      call    0x1e040
0x1ed18:      mov     eax, dword ptr [rsp + 0xc]
0x1ed1c:      mov     dword ptr [rbp], eax

```

```

0x1ed1f:      add     rsp, 0x18
0x1ed23:      mov     rax, r12
0x1ed26:      pop     rbx
0x1ed27:      pop     rbp
0x1ed28:      pop     r12
0x1ed2a:      pop     r13
0x1ed2c:      pop     r14
0x1ed2e:      pop     r15
0x1ed30:      ret
0x1ed31:      nop     word ptr cs:[rax + rax]
0x1ed3c:      nop     dword ptr [rax]
0x1ed40:      endbr64
0x1ed44:      mov     eax, dword ptr [rip + 0xbaba]
0x1ed4a:      push    r12
0x1ed4c:      mov     r12, qword ptr [rip + 0xbabd]
0x1ed53:      push    rbp
0x1ed54:      push    rbx
0x1ed55:      cmp     eax, 1
0x1ed58:      jbe     0x1ed81
0x1ed5a:      sub     eax, 2
0x1ed5d:      lea     rbx, [r12 + 0x18]
0x1ed62:      shl     rax, 4
0x1ed66:      lea     rbp, [r12 + rax + 0x28]
0x1ed6b:      nop     dword ptr [rax + rax]
0x1ed70:      mov     rdi, qword ptr [rbx]
0x1ed73:      add     rbx, 0x10
0x1ed77:      call    0x4800
0x1ed7c:      cmp     rbx, rbp
0x1ed7f:      jne     0x1ed70
0x1ed81:      mov     rdi, qword ptr [r12 + 8]
0x1ed86:      lea     rbx, [rip + 0xcdd3]
0x1ed8d:      cmp     rdi, rbx
0x1ed90:      je      0x1eda9
0x1ed92:      call    0x4800
0x1ed97:      mov     qword ptr [rip + 0xba8a], rbx
0x1ed9e:      mov     qword ptr [rip + 0xba77], 0x100
0x1eda9:      lea     rbx, [rip + 0xba70]
0x1edb0:      cmp     r12, rbx
0x1edb3:      je      0x1edc4
0x1edb5:      mov     rdi, r12
0x1edb8:      call    0x4800
0x1ebdb:      mov     qword ptr [rip + 0xba4c], rbx
0x1edc4:      mov     dword ptr [rip + 0xba36], 1
0x1edce:      pop     rbx
0x1edcf:      pop     rbp
0x1edd0:      pop     r12
0x1edd2:      ret
0x1edd3:      nop     word ptr cs:[rax + rax]

```

```

0x1edde:      nop
0x1ede0:      endbr64
0x1ede4:      lea     rcx, [rip + 0xce75]
0x1edeb:      mov     rdx, -1
0x1edf2:      jmp     0x1e9e0
0x1e9e0
0x1edf7:      nop     word ptr [rax + rax]
0x1ee00:      endbr64
0x1ee04:      mov     rsi, rdi
0x1ee07:      lea     rcx, [rip + 0xce52]
0x1ee0e:      mov     rdx, -1
0x1ee15:      xor     edi, edi
0x1ee17:      jmp     0x1e9e0
0x1e9e0
0x1ee1c:      nop     dword ptr [rax]
0x1ee20:      endbr64
0x1ee24:      sub     rsp, 0x68
0x1ee28:      mov     r8d, esi
0x1ee2b:      pxor    xmm0, xmm0
0x1ee2f:      mov     rsi, rdx
0x1ee32:      mov     rax, qword ptr fs:[0x28]
0x1ee3b:      mov     qword ptr [rsp + 0x58], rax
0x1ee40:      xor     eax, eax
0x1ee42:      movups  xmmword ptr [rsp + 0x34], xmm0
0x1ee47:      mov     rdx, -1
0x1ee4e:      mov     rcx, rsp
0x1ee51:      movups  xmmword ptr [rsp + 0x44], xmm0
0x1ee56:      movdqa  xmm2, xmmword ptr [rsp + 0x40]
0x1ee5c:      mov     eax, dword ptr [rsp + 0x50]
0x1ee60:      mov     dword ptr [rsp + 0x30], r8d
0x1ee65:      movdqa  xmm1, xmmword ptr [rsp + 0x30]
0x1ee6b:      mov     dword ptr [rsp + 0x20], eax
0x1ee6f:      movaps  xmmword ptr [rsp], xmm1
0x1ee73:      movaps  xmmword ptr [rsp + 0x10], xmm2
0x1ee78:      call    0x1e9e0
0x1ee7d:      mov     rdx, qword ptr [rsp + 0x58]
0x1ee82:      sub     rdx, qword ptr fs:[0x28]
0x1ee8b:      jne     0x1ee92
0x1ee8d:      add     rsp, 0x68
0x1ee91:      ret
0x1ee92:      call    0x49d0
0x1ee97:      nop     word ptr [rax + rax]
0x1eea0:      endbr64
0x1eea4:      sub     rsp, 0x68
0x1eea8:      mov     r8d, esi
0x1eeab:      pxor    xmm0, xmm0
0x1eeaf:      mov     rsi, rdx
0x1eeb2:      mov     rax, qword ptr fs:[0x28]

```

```

0x1eebb:    mov     qword ptr [rsp + 0x58], rax
0x1eec0:    xor     eax, eax
0x1eec2:    movups  xmmword ptr [rsp + 0x34], xmm0
0x1eec7:    mov     rdx, rcx
0x1eeca:    mov     rcx, rsp
0x1eecd:    movups  xmmword ptr [rsp + 0x44], xmm0
0x1eed2:    movdqa  xmm2, xmmword ptr [rsp + 0x40]
0x1eed8:    mov     eax, dword ptr [rsp + 0x50]
0x1eedc:    mov     dword ptr [rsp + 0x30], r8d
0x1eee1:    movdqa  xmm1, xmmword ptr [rsp + 0x30]
0x1eee7:    mov     dword ptr [rsp + 0x20], eax
0x1eeeb:    movaps  xmmword ptr [rsp], xmm1
0x1eeef:    movaps  xmmword ptr [rsp + 0x10], xmm2
0x1eef4:    call    0x1e9e0
0x1eef9:    mov     rdx, qword ptr [rsp + 0x58]
0x1eefe:    sub     rdx, qword ptr fs:[0x28]
0x1ef07:    jne     0x1ef0e
0x1ef09:    add     rsp, 0x68
0x1ef0d:    ret
0x1ef0e:    call    0x49d0
0x1ef13:    nop     word ptr cs:[rax + rax]
0x1ef1e:    nop
0x1ef20:    endbr64
0x1ef24:    sub     rsp, 0x68
0x1ef28:    pxor    xmm0, xmm0
0x1ef2c:    mov     rdx, -1
0x1ef33:    mov     rax, qword ptr fs:[0x28]
0x1ef3c:    mov     qword ptr [rsp + 0x58], rax
0x1ef41:    xor     eax, eax
0x1ef43:    mov     dword ptr [rsp + 0x30], edi
0x1ef47:    mov     rcx, rsp
0x1ef4a:    xor     edi, edi
0x1ef4c:    movups  xmmword ptr [rsp + 0x34], xmm0
0x1ef51:    movdqa  xmm1, xmmword ptr [rsp + 0x30]
0x1ef57:    movups  xmmword ptr [rsp + 0x44], xmm0
0x1ef5c:    movdqa  xmm2, xmmword ptr [rsp + 0x40]
0x1ef62:    mov     eax, dword ptr [rsp + 0x50]
0x1ef66:    movaps  xmmword ptr [rsp], xmm1
0x1ef6a:    mov     dword ptr [rsp + 0x20], eax
0x1ef6e:    movaps  xmmword ptr [rsp + 0x10], xmm2
0x1ef73:    call    0x1e9e0
0x1ef78:    mov     rdx, qword ptr [rsp + 0x58]
0x1ef7d:    sub     rdx, qword ptr fs:[0x28]
0x1ef86:    jne     0x1ef8d
0x1ef88:    add     rsp, 0x68
0x1ef8c:    ret
0x1ef8d:    call    0x49d0
0x1ef92:    nop     word ptr cs:[rax + rax]

```



```

0x1ef9d:      nop      dword ptr [rax]
0x1efa0:      endbr64
0x1efa4:      sub      rsp, 0x38
0x1efa8:      movdqa   xmm0, xmmword ptr [rip + 0xccb0]
0x1efb0:      movdqa   xmm1, xmmword ptr [rip + 0xccb8]
0x1efb8:      mov      ecx, esi
0x1efba:      mov      rax, qword ptr fs:[0x28]
0x1efc3:      mov      qword ptr [rsp + 0x28], rax
0x1efc8:      xor      eax, eax
0x1efca:      mov      eax, dword ptr [rip + 0xccb0]
0x1efd0:      and      ecx, 0x1f
0x1efd3:      mov      r9, rsp
0x1efd6:      movaps   xmmword ptr [rsp], xmm0
0x1efda:      mov      dword ptr [rsp + 0x20], eax
0x1efde:      mov      eax, esi
0x1efe0:      shr      al, 5
0x1efe3:      movaps   xmmword ptr [rsp + 0x10], xmm1
0x1efe8:      movzx    eax, al
0x1efeb:      lea      rdx, [rsp + rax*4 + 4]
0x1eff0:      mov      esi, dword ptr [rdx]
0x1eff2:      mov      eax, esi
0x1eff4:      shr      eax, cl
0x1eff6:      not      eax
0x1eff8:      and      eax, 1
0x1effb:      shl      eax, cl
0x1effd:      mov      rcx, r9
0x1f000:      xor      eax, esi
0x1f002:      mov      rsi, rdi
0x1f005:      xor      edi, edi
0x1f007:      mov      dword ptr [rdx], eax
0x1f009:      mov      rdx, -1
0x1f010:      call     0x1e9e0
0x1f015:      mov      rdx, qword ptr [rsp + 0x28]
0x1f01a:      sub      rdx, qword ptr fs:[0x28]
0x1f023:      jne      0x1f02a
0x1f025:      add      rsp, 0x38
0x1f029:      ret
0x1f02a:      call     0x49d0
0x1f02f:      nop
0x1f030:      endbr64
0x1f034:      mov      esi, 0x3a
0x1f039:      jmp      0x1efa0
0x1efa0:
0x1e4b0:      mov      ebp, edx
0x1e4b2:      cmp      dl, 0x3f
0x1e4b5:      jg       0x1e6d8
0x1e4bb:      test     dl, dl
0x1e4bd:      js       0x1e500

```

```

0x1e4bf:      cmp     dl, 0x3f
0x1e4c2:      ja      0x1e500
0x1e4c4:      lea     rdi, [rip + 0x43c5]
0x1e4cb:      movzx   eax, dl
0x1e4ce:      movsxd  rax, dword ptr [rdi + rax*4]
0x1e4d2:      add     rax, rdi
0x1e4d5:      jmp     rax
rax
->->->                                     indirect JMP
0x1e458:      cmp     dl, 0x7a
0x1e45b:      jg      0x1e220
0x1e461:      cmp     dl, 0x40
0x1e464:      je      0x1e240
0x1e46a:      lea     ecx, [rbp - 0x41]
0x1e46d:      mov     eax, 1
0x1e472:      movabs  rdi, 0x3fffffff53ffffff
0x1e47c:      shl     rax, cl
0x1e47f:      test    rax, rdi
0x1e482:      jne     0x1e1b0
0x1e488:      test    eax, 0xa4000000
0x1e48d:      jne     0x1e20a
0x1e493:      cmp     bpl, 0x5c
0x1e497:      jne     0x1e240
0x1e49d:      cmp     dword ptr [rsp + 0x10], 1
0x1e4a2:      je      0x1e380
0x1e4a8:      jmp     0x1e34d
0x1e34d
0x1e4ad:      nop     dword ptr [rax]
0x1e4b0:      mov     ebp, edx
0x1e4b2:      cmp     dl, 0x3f
0x1e4b5:      jg      0x1e6d8
0x1e4bb:      test    dl, dl
0x1e4bd:      js      0x1e500
0x1e4bf:      cmp     dl, 0x3f
0x1e4c2:      ja      0x1e500
0x1e4c4:      lea     rdi, [rip + 0x43c5]
0x1e4cb:      movzx   eax, dl
0x1e4ce:      movsxd  rax, dword ptr [rdi + rax*4]
0x1e4d2:      add     rax, rdi
0x1e4d5:      jmp     rax
rax
->->->                                     indirect JMP
0x1e240:      cmp     qword ptr [rsp + 0x28], 1
0x1e246:      mov     byte ptr [rsp + 0x30], dl
0x1e24a:      jne     0x1e518
0x1e250:      mov     qword ptr [rsp + 0x50], r11
0x1e255:      mov     qword ptr [rsp + 0x48], r10
0x1e25a:      mov     byte ptr [rsp + 0x40], r8b

```

```

0x1e25f:      mov     qword ptr [rsp + 0x38], r9
0x1e264:      call   0x4f00
0x1e269:      movzx   edx, byte ptr [rsp + 0x30]
0x1e26e:      movzx   r8d, byte ptr [rsp + 0x40]
0x1e274:      mov     ecx, 1
0x1e279:      mov     rax, qword ptr [rax]
0x1e27c:      mov     r9, qword ptr [rsp + 0x38]
0x1e281:      mov     r10, qword ptr [rsp + 0x48]
0x1e286:      mov     r11, qword ptr [rsp + 0x50]
0x1e28b:      movzx   eax, word ptr [rax + rdx*2]
0x1e28f:      shr     ax, 0xe
0x1e293:      xor     eax, 1
0x1e296:      and     eax, r8d
0x1e299:      test    al, al
0x1e29b:      je      0x1e1a4
0x1e2a1:      mov     eax, r8d
0x1e2a4:      mov     rsi, qword ptr [rsp + 8]
0x1e2a9:      add     rcx, r12
0x1e2ac:      jmp     0x1e2c6
0x1e2c6:
0x1e2ae:      nop
0x1e2b0:      cmp     rbx, r10
0x1e2b3:      jae     0x1e2b9
0x1e2b5:      mov     byte ptr [rsi + rbx], bpl
0x1e2b9:      movzx   ebp, byte ptr [r11 + r14]
0x1e2be:      add     rbx, 1
0x1e2c2:      add     r14, 1
0x1e2c6:      test    al, al
0x1e2c8:      je      0x1e30a
0x1e2ca:      cmp     rbx, r10
0x1e2cd:      jae     0x1e2d3
0x1e2cf:      mov     byte ptr [rsi + rbx], 0x5c
0x1e2d3:      lea     rdx, [rbx + 1]
0x1e2d7:      cmp     r10, rdx
0x1e2da:      jbe     0x1e2e8
0x1e2dc:      mov     edx, ebp
0x1e2de:      shr     dl, 6
0x1e2e1:      add     edx, 0x30
0x1e2e4:      mov     byte ptr [rsi + rbx + 1], dl
0x1e2e8:      lea     rdx, [rbx + 2]
0x1e2ec:      cmp     r10, rdx
0x1e2ef:      jbe     0x1e300
0x1e2f1:      mov     edx, ebp
0x1e2f3:      shr     dl, 3
0x1e2f6:      and     edx, 7
0x1e2f9:      add     edx, 0x30
0x1e2fc:      mov     byte ptr [rsi + rbx + 2], dl
0x1e300:      and     ebp, 7

```

```

0x1e303:      add     rbx, 3
0x1e307:      add     ebp, 0x30
0x1e30a:      mov     r12, r14
0x1e30d:      cmp     r14, rcx
0x1e310:      jb      0x1e2b0
0x1e312:      jmp     0x1e1e1
0x1e1e1
0x1e317:      nop     word ptr [rax + rax]
0x1e320:      mov     eax, dword ptr [rsp + 0x10]
0x1e324:      cmp     eax, 1
0x1e327:      je      0x1e380
0x1e329:      cmp     eax, 3
0x1e32c:      je      0x1e760
0x1e332:      lea     r15, [r12 + 1]
0x1e337:      mov     ebp, 0x3f
0x1e33c:      jmp     0x1e1a4
0x1e1a4
0x1e341:      mov     ebp, 0x72
0x1e346:      cmp     dword ptr [rsp + 0x10], 1
0x1e34b:      je      0x1e380
0x1e34d:      mov     r12, r14
0x1e350:      jmp     0x1e1cf
0x1e1cf
0x1e355:      mov     ebp, 0x62
0x1e35a:      jmp     0x1e34d
0x1e34d
0x1e35c:      nop     dword ptr [rax]
0x1e360:      mov     ebp, 0x23
0x1e365:      test    r12, r12
0x1e368:      jne     0x1e215
0x1e36e:      cmp     dword ptr [rsp + 0x10], 1
0x1e373:      jne     0x1e215
0x1e379:      nop     dword ptr [rax]
0x1e380:      mov     qword ptr [rsp + 0x20], r11
0x1e385:      mov     qword ptr [rsp + 0x10], r10
0x1e38a:      call    0x49a0
0x1e38f:      mov     r10, qword ptr [rsp + 0x10]
0x1e394:      mov     r11, qword ptr [rsp + 0x20]
0x1e399:      mov     qword ptr [rsp + 0x28], rax
0x1e39e:      lea     rax, [rip + 0x43b7]
0x1e3a5:      mov     dword ptr [rsp + 0x10], 2
0x1e3ad:      mov     r9d, 1
0x1e3b3:      xor     r8d, r8d
0x1e3b6:      mov     qword ptr [rsp + 0x20], rax
0x1e3bb:      test    r10, r10
0x1e3be:      je      0x1e0b0
0x1e3c4:      mov     rax, qword ptr [rsp + 8]
0x1e3c9:      mov     byte ptr [rax], 0x27

```

```

0x1e3cc:      jmp      0x1e0b0
0x1e0b0
0x1e3d1:      mov      ebp, 0x74
0x1e3d6:      jmp      0x1e346
0x1e346
0x1e3db:      mov      ebp, 0x66
0x1e3e0:      jmp      0x1e34d
0x1e34d
0x1e3e5:      mov      r12, r14
0x1e3e8:      mov      ebp, 0x76
0x1e3ed:      jmp      0x1e1cf
0x1e1cf
0x1e3f2:      mov      r12, r14
0x1e3f5:      cmp      r10, rbx
0x1e3f8:      jbe      0x1e403
0x1e3fa:      mov      rax, qword ptr [rsp + 8]
0x1e3ff:      mov      byte ptr [rax + rbx], 0x5c
0x1e403:      lea      rax, [rbx + 1]
0x1e407:      cmp      r10, rax
0x1e40a:      jbe      0x1e416
0x1e40c:      mov      rax, qword ptr [rsp + 8]
0x1e411:      mov      byte ptr [rax + rbx + 1], 0x30
0x1e416:      lea      rax, [rbx + 2]
0x1e41a:      cmp      r10, rax
0x1e41d:      jbe      0x1e429
0x1e41f:      mov      rax, qword ptr [rsp + 8]
0x1e424:      mov      byte ptr [rax + rbx + 2], 0x30
0x1e429:      add      rbx, 3
0x1e42d:      mov      eax, 0x10000
0x1e432:      mov      edx, 1
0x1e437:      mov      ebp, 0x30
0x1e43c:      jmp      0x1e1c4
0x1e1c4
0x1e441:      mov      ebp, 0x61
0x1e446:      jmp      0x1e34d
0x1e34d
0x1e44b:      mov      ebp, 0x6e
0x1e450:      jmp      0x1e346
0x1e346
0x1e455:      nop      dword ptr [rax]
0x1e458:      cmp      dl, 0x7a
0x1e45b:      jg       0x1e220
0x1e461:      cmp      dl, 0x40
0x1e464:      je       0x1e240
0x1e46a:      lea      ecx, [rbp - 0x41]
0x1e46d:      mov      eax, 1
0x1e472:      movabs   rdi, 0x3fffffff53ffffff
0x1e47c:      shl      rax, cl

```

```

0x1e47f:      test    rax, rdi
0x1e482:      jne     0x1e1b0
0x1e488:      test    eax, 0xa4000000
0x1e48d:      jne     0x1e20a
0x1e493:      cmp     bpl, 0x5c
0x1e497:      jne     0x1e240
0x1e49d:      cmp     dword ptr [rsp + 0x10], 1
0x1e4a2:      je      0x1e380
0x1e4a8:      jmp     0x1e34d
0x1e34d
0x1e4ad:      nop     dword ptr [rax]
0x1e4b0:      mov     ebp, edx
0x1e4b2:      cmp     dl, 0x3f
0x1e4b5:      jg      0x1e6d8
0x1e4bb:      test    dl, dl
0x1e4bd:      js      0x1e500
0x1e4bf:      cmp     dl, 0x3f
0x1e4c2:      ja      0x1e500
0x1e4c4:      lea     rdi, [rip + 0x43c5]
0x1e4cb:      movzx   eax, dl
0x1e4ce:      movsxd  rax, dword ptr [rdi + rax*4]
0x1e4d2:      add     rax, rdi
0x1e4d5:      jmp     rax
rax
->->->                                indirect JMP
0x104d0:      mov     rax, qword ptr [rsp + 0x298]
0x104d8:      sub     rax, qword ptr fs:[0x28]
0x104e1:      jne     0x1075a
0x104e7:      add     rsp, 0x2a0
0x104ee:      pop     rbx
0x104ef:      pop     rbp
0x104f0:      pop     r12
0x104f2:      ret
0x104f3:      nop     dword ptr [rax + rax]
0x104f8:      mov     rdx, rsp
0x104fb:      xor     eax, eax
0x104fd:      call    0xe560
0x10502:      movsxd  rax, dword ptr [rbx + 0x64]
0x10506:      mov     rdx, qword ptr [rbx + 0x20]
0x1050a:      movsxd  rcx, ebp
0x1050d:      sub     rax, rcx
0x10510:      lea     rax, [rdx + rax - 2]
0x10515:      mov     rdx, qword ptr [rsp]
0x10519:      mov     qword ptr [rax], rdx
0x1051c:      movzx   edx, word ptr [rsp + 8]
0x10521:      mov     word ptr [rax + 8], dx
0x10525:      jmp     0x104d0
0x104d0

```

```

0x10527:      nop      word ptr [rax + rax]
0x10530:      movsxd   rax, dword ptr [rdi + 0x64]
0x10534:      mov      rdx, qword ptr [rdi + 0x20]
0x10538:      movsxd   rcx, ebp
0x1053b:      movsxd   rsi, esi
0x1053e:      sub      rax, rcx
0x10541:      lea      rdx, [rdx + rax - 2]
0x10546:      lea      rax, [rsi + rsi*4]
0x1054a:      shl      rax, 4
0x1054e:      add      rax, qword ptr [rdi + 0x38]
0x10552:      lea      rcx, [rax + 0x38]
0x10556:      mov      rax, qword ptr [rax + 0x38]
0x1055a:      mov      qword ptr [rdx], rax
0x1055d:      mov      eax, dword ptr [rcx + 8]
0x10560:      mov      dword ptr [rdx + 8], eax
0x10563:      movzx    eax, word ptr [rcx + 0xc]
0x10567:      mov      word ptr [rdx + 0xc], ax
0x1056b:      movzx    eax, byte ptr [rcx + 0xe]
0x1056f:      mov      byte ptr [rdx + 0xe], al
0x10572:      jmp      0x104d0
0x104d0
0x10577:      nop      word ptr [rax + rax]
0x10580:      movsxd   rsi, esi
0x10583:      mov      edx, 0xc
0x10588:      mov      ecx, 1
0x1058d:      mov      r8d, 1
0x10593:      lea      rax, [rsi + rsi*4]
0x10597:      mov      rsi, rsp
0x1059a:      shl      rax, 4
0x1059e:      add      rax, qword ptr [rdi + 0x38]
0x105a2:      mov      rdi, qword ptr [rax + 8]
0x105a6:      mov      eax, 0x9c
0x105ab:      cmp      rdi, 0x2710
0x105b2:      cmovge   edx, eax
0x105b5:      call     0x1c520
0x105ba:      mov      rdi, rax
0x105bd:      mov      r12, rax
0x105c0:      call     0x49b0
0x105c5:      mov      edx, 0xa
0x105ca:      movsxd   rsi, dword ptr [rbx + 0x64]
0x105ce:      movsxd   rcx, ebp
0x105d1:      cmp      rax, rdx
0x105d4:      cmovbe   rdx, rax
0x105d8:      sub      rsi, rcx
0x105db:      mov      rcx, qword ptr [rbx + 0x20]
0x105df:      lea      rcx, [rcx + rsi - 2]
0x105e4:      cmp      rax, 9
0x105e8:      ja       0x10621

```

```

0x105ea:      movabs   rdi, 0x2020202020202020
0x105f4:      mov     eax, 0xa
0x105f9:      sub     eax, edx
0x105fb:      movsxd  r10, eax
0x105fe:      cmp     eax, 8
0x10601:      jae     0x106e0
0x10607:      test    al, 4
0x10609:      jne     0x10734
0x1060f:      test    eax, eax
0x10611:      je      0x1061e
0x10613:      mov     byte ptr [rcx], 0x20
0x10616:      test    al, 2
0x10618:      jne     0x10749
0x1061e:      add     rcx, r10
0x10621:      cmp     edx, 8
0x10624:      jae     0x10690
0x10626:      test    dl, 4
0x10629:      jne     0x1071e
0x1062f:      test    edx, edx
0x10631:      je      0x104d0
0x10637:      movzx   eax, byte ptr [r12]
0x1063c:      mov     byte ptr [rcx], al
0x1063e:      test    dl, 2
0x10641:      je      0x104d0
0x10647:      mov     edx, edx
0x10649:      movzx   eax, word ptr [r12 + rdx - 2]
0x1064f:      mov     word ptr [rcx + rdx - 2], ax
0x10654:      jmp     0x104d0
0x104d0:
0x10659:      nop     dword ptr [rax]
0x10660:      mov     rax, qword ptr [rsp + 0x298]
0x10668:      sub     rax, qword ptr fs:[0x28]
0x10671:      jne     0x1075a
0x10677:      add     rsp, 0x2a0
0x1067e:      lea     rdi, [rip + 0x10b67]
0x10685:      pop     rbx
0x10686:      pop     rbp
0x10687:      pop     r12
0x10689:      jmp     0x8b00
0x8b00:
0x1068e:      nop
0x10690:      mov     rax, qword ptr [r12]
0x10694:      mov     qword ptr [rcx], rax
0x10697:      mov     eax, edx
0x10699:      mov     rsi, qword ptr [r12 + rax - 8]
0x1069e:      mov     qword ptr [rcx + rax - 8], rsi
0x106a3:      lea     rsi, [rcx + 8]
0x106a7:      and     rsi, 0xfffffffffffffff8

```



```

0x106ab:    sub     rcx, rsi
0x106ae:    add     edx, ecx
0x106b0:    sub     r12, rcx
0x106b3:    and     edx, 0xffffffff8
0x106b6:    cmp     edx, 8
0x106b9:    jb      0x104d0
0x106bf:    and     edx, 0xffffffff8
0x106c2:    xor     eax, eax
0x106c4:    mov     ecx, eax
0x106c6:    add     eax, 8
0x106c9:    mov     rdi, qword ptr [r12 + rcx]
0x106cd:    mov     qword ptr [rsi + rcx], rdi
0x106d1:    cmp     eax, edx
0x106d3:    jb      0x106c4
0x106d5:    jmp     0x104d0
0x104d0:
0x106da:    nop     word ptr [rax + rax]
0x106e0:    mov     esi, eax
0x106e2:    lea     r8, [rcx + 8]
0x106e6:    mov     qword ptr [rcx], rdi
0x106e9:    and     r8, 0xffffffffffffff8
0x106ed:    mov     qword ptr [rcx + rsi - 8], rdi
0x106f2:    mov     rsi, rcx
0x106f5:    sub     rsi, r8
0x106f8:    add     eax, esi
0x106fa:    and     eax, 0xffffffff8
0x106fd:    cmp     eax, 8
0x10700:    jb      0x1061e
0x10706:    and     eax, 0xffffffff8
0x10709:    xor     esi, esi
0x1070b:    mov     r9d, esi
0x1070e:    add     esi, 8
0x10711:    mov     qword ptr [r8 + r9], rdi
0x10715:    cmp     esi, eax
0x10717:    jb      0x1070b
0x10719:    jmp     0x1061e
0x1061e:
0x1071e:    mov     eax, dword ptr [r12]
0x10722:    mov     edx, edx
0x10724:    mov     dword ptr [rcx], eax
0x10726:    mov     eax, dword ptr [r12 + rdx - 4]
0x1072b:    mov     dword ptr [rcx + rdx - 4], eax
0x1072f:    jmp     0x104d0
0x104d0:
0x10734:    mov     eax, eax
0x10736:    mov     dword ptr [rcx], 0x20202020
0x1073c:    mov     dword ptr [rcx + rax - 4], 0x20202020
0x10744:    jmp     0x1061e

```

```

0x1061e
0x10749:      mov     eax, eax
0x1074b:      mov     esi, 0x2020
0x10750:      mov     word ptr [rcx + rax - 2], si
0x10755:      jmp     0x1061e
0x1061e
0x1075a:      call    0x49d0
0x1075f:      nop
0x10760:      push    r15
0x10762:      push    r14
0x10764:      mov     r14d, 4
0x1076a:      push    r13
0x1076c:      mov     r13, rdi
0x1076f:      push    r12
0x10771:      push    rbp
0x10772:      push    rbx
0x10773:      mov     ebx, esi
0x10775:      mov     esi, 0x20
0x1077a:      sub     rsp, 0x28
0x1077e:      movsxd  rdx, dword ptr [rdi + 0x64]
0x10782:      mov     rdi, qword ptr [rdi + 0x20]
0x10786:      call    0x4a50
0x1078b:      mov     ebp, dword ptr [r13 + 0x64]
0x1078f:      cmp     ebp, 0x27
0x10792:      jle     0x107b0
0x10794:      movsxd  rax, dword ptr [r13 + 0x80]
0x1079b:      cmp     ebp, 0x49
0x1079e:      jg      0x107a5
0x107a0:      cmp     eax, 5
0x107a3:      je      0x107b0
0x107a5:      lea     rdx, [rip + 0x11234]
0x107ac:      movsxd  r14, dword ptr [rdx + rax*4]
0x107b0:      movsxd  rax, ebx
0x107b3:      lea     r12, [rax + rax*4]
0x107b7:      mov     rax, qword ptr [r13 + 0x38]
0x107bb:      shl     r12, 4
0x107bf:      add     rax, r12
0x107c2:      test    ebx, ebx
0x107c4:      jle     0x107e8
0x107c6:      cmp     byte ptr [rax + 0x48], 4
0x107ca:      ja      0x4f20
0x107d0:      movzx   edx, byte ptr [rax + 0x48]
0x107d4:      lea     rcx, [rip + 0x11101]
0x107db:      movsxd  rdx, dword ptr [rcx + rdx*4]
0x107df:      add     rdx, rcx
0x107e2:      jmp     rdx
rdx
->->->                                     indirect JMP

```

```

0x75ad: add    rsp, 0x10
0x75b1: mov     eax, r12d
0x75b4: pop     r12
0x75b6: ret
0x75b7: nop     word ptr [rax + rax]
0x75c0: xor     eax, eax
0x75c2: mov     r12d, 1
0x75c8: call    0xcb30
0x75cd: jmp     0x75ad
0x75ad
0x75cf: nop
0x75d0: xor     eax, eax
0x75d2: mov     r12d, 1
0x75d8: call    0xcb60
0x75dd: jmp     0x75ad
0x75ad
0x75df: nop
0x75e0: xor     eax, eax
0x75e2: mov     r12d, 1
0x75e8: call    0xcc30
0x75ed: jmp     0x75ad
0x75ad
0x75ef: nop
0x75f0: xor     eax, eax
0x75f2: mov     r12d, 1
0x75f8: call    0xccf0
0x75fd: jmp     0x75ad
0x75ad
0x75ff: nop
0x7600: xor     eax, eax
0x7602: mov     r12d, 1
0x7608: call    0xcd10
0x760d: jmp     0x75ad
0x75ad
0x760f: nop
0x7610: and     r12d, 1
0x7614: jne     0x7820
0x761a: mov     r12d, 1
0x7620: add     rsp, 0x10
0x7624: mov     eax, r12d
0x7627: pop     r12
0x7629: ret
0x762a: nop     word ptr [rax + rax]
0x7630: and     r12d, 1
0x7634: je      0x761a
0x7636: xor     eax, eax
0x7638: call    0xce70
0x763d: jmp     0x75ad

```

```

0x75ad
0x7642: nop      word ptr [rax + rax]
0x7648: and      r12d, 1
0x764c: je       0x761a
0x764e: xor      eax, eax
0x7650: call     0xd380
0x7655: jmp      0x75ad
0x75ad
0x765a: nop      word ptr [rax + rax]
0x7660: and      r12d, 1
0x7664: je       0x761a
0x7666: xor      eax, eax
0x7668: call     0xd400
0x766d: jmp      0x75ad
0x75ad
0x7672: nop      word ptr [rax + rax]
0x7678: and      r12d, 1
0x767c: je       0x761a
0x767e: xor      eax, eax
0x7680: call     0xd480
0x7685: jmp      0x75ad
0x75ad
0x768a: nop      word ptr [rax + rax]
0x7690: xor      eax, eax
0x7692: mov      r12d, 1
0x7698: call     0xca60
0x769d: jmp      0x75ad
0x75ad
0x76a2: nop      word ptr [rax + rax]
0x76a8: and      r12d, 1
0x76ac: je       0x761a
0x76b2: xor      eax, eax
0x76b4: call     0xca80
0x76b9: jmp      0x75ad
0x75ad
0x76be: nop
0x76c0: xor      eax, eax
0x76c2: mov      r12d, 1
0x76c8: call     0xcab0
0x76cd: jmp      0x75ad
0x75ad
0x76d2: nop      word ptr [rax + rax]
0x76d8: and      r12d, 1
0x76dc: je       0x761a
0x76e2: xor      eax, eax
0x76e4: call     0xd700
0x76e9: jmp      0x75ad
0x75ad

```

```

0x76ee: nop
0x76f0: xor     eax, eax
0x76f2: mov     r12d, 1
0x76f8: call    0xcae0
0x76fd: jmp     0x75ad
0x75ad
0x7702: nop     word ptr [rax + rax]
0x7708: and     r12d, 1
0x770c: je      0x761a
0x7712: xor     eax, eax
0x7714: call    0xd1f0
0x7719: jmp     0x75ad
0x75ad
0x771e: nop
0x7720: and     r12d, 1
0x7724: je      0x761a
0x772a: xor     eax, eax
0x772c: call    0xd250
0x7731: jmp     0x75ad
0x75ad
0x7736: nop     word ptr cs:[rax + rax]
0x7740: and     r12d, 1
0x7744: je      0x761a
0x774a: xor     eax, eax
0x774c: call    0xd330
0x7751: jmp     0x75ad
0x75ad
0x7756: nop     word ptr cs:[rax + rax]
0x7760: and     r12d, 1
0x7764: je      0x761a
0x776a: xor     eax, eax
0x776c: call    0xd2b0
0x7771: jmp     0x75ad
0x75ad
0x7776: nop     word ptr cs:[rax + rax]
0x7780: and     r12d, 1
0x7784: je      0x761a
0x778a: xor     eax, eax
0x778c: call    0xcf80
0x7791: jmp     0x75ad
0x75ad
0x7796: nop     word ptr cs:[rax + rax]
0x77a0: and     r12d, 1
0x77a4: je      0x761a
0x77aa: mov     edi, 1
0x77af: call    0xd1a0
0x77b4: jmp     0x75ad
0x75ad

```

```

0x77b9: nop      dword ptr [rax]
0x77c0: and      r12d, 1
0x77c4: je       0x761a
0x77ca: xor      eax, eax
0x77cc: call     0xcf10
0x77d1: jmp      0x75ad
0x75ad
0x77d6: nop      word ptr cs:[rax + rax]
0x77e0: mov      dword ptr [rsp + 0xc], edi
0x77e4: and      r12d, 1
0x77e8: je       0x75ad
0x77ee: call     0x4f00
0x77f3: mov      r8, rax
0x77f6: movsxd   rax, dword ptr [rsp + 0xc]
0x77fb: mov      rdx, qword ptr [r8]
0x77fe: mov      rdi, rax
0x7801: test     byte ptr [rdx + rax*2 + 1], 0x40
0x7806: je       0x7818
0x7808: call     0xcd30
0x780d: jmp      0x75ad
0x75ad
0x7812: nop      word ptr [rax + rax]
0x7818: xor      r12d, r12d
0x781b: jmp      0x75ad
0x75ad
0x7820: xor      eax, eax
0x7822: call     0xcde0
0x7827: jmp      0x75ad
0x75ad
0x782c: nop      dword ptr [rax]
0x7830: endbr64
0x7834: xor      eax, eax
0x7836: cmp      dword ptr [rip + 0x2315f], 6
0x783d: jle      0x7869
0x783f: mov      edx, dword ptr [rip + 0x22413]
0x7845: mov      eax, dword ptr [rip + 0x2314d]
0x784b: test     edx, edx
0x784d: je       0x7860
0x784f: cmp      eax, 0xb
0x7852: setg     al
0x7855: movzx    eax, al
0x7858: ret
0x7859: nop      dword ptr [rax]
0x7860: cmp      eax, 5
0x7863: setg     al
0x7866: movzx    eax, al
0x7869: ret
0x786a: nop      word ptr [rax + rax]

```

```

0x7870: endbr64
0x7874: xor     eax, eax
0x7876: cmp     dword ptr [rip + 0x23067], 1
0x787d: sete    al
0x7880: ret
0x7881: nop     word ptr cs:[rax + rax]
0x788c: nop     dword ptr [rax]
0x7890: endbr64
0x7894: push    r13
0x7896: xor     eax, eax
0x7898: push    r12
0x789a: push    rbp
0x789b: push    rbx
0x789c: mov     ebx, edi
0x789e: sub     rsp, 8
0x78a2: mov     ebp, dword ptr [rip + 0x230f4]
0x78a8: mov     r12d, dword ptr [rip + 0x230e9]
0x78af: call    0xb520
0x78b4: mov     ecx, dword ptr [rip + 0x230e2]
0x78ba: test    ebx, ebx
0x78bc: jne     0x78c6
0x78be: cmp     ecx, ebp
0x78c0: je      0x7b00
0x78c6: mov     edx, dword ptr [rip + 0x230bc]
0x78cc: test    edx, edx
0x78ce: jne     0x7a60
0x78d4: mov     rdi, qword ptr [rip + 0x22fd5]
0x78db: cmp     ecx, 1
0x78de: jle     0x79e0
0x78e4: cmp     ecx, 2
0x78e7: je      0x7b70
0x78ed: cmp     ecx, 6
0x78f0: jle     0x790d
0x78f2: mov     eax, dword ptr [rip + 0x22360]
0x78f8: movsxd  r8, dword ptr [rip + 0x23099]
0x78ff: test    eax, eax
0x7901: je      0x7a90
0x7907: cmp     r8d, 0xb
0x790b: jg      0x7950
0x790d: mov     r8d, 0x50
0x7913: mov     ecx, 2
0x7918: mov     edx, 0x10000
0x791d: mov     esi, 0x10000
0x7922: call    0xf590
0x7927: mov     r8d, 0x50
0x792d: mov     ecx, 2
0x7932: mov     rdi, qword ptr [rip + 0x22f6f]
0x7939: mov     edx, 0x10000

```

```

0x793e: mov     esi, 0x10000
0x7943: call    0xf590
0x7948: jmp     0x7ad2
0x7ad2:
0x794d: nop     dword ptr [rax]
0x7950: mov     ebp, r8d
0x7953: and     r8d, 1
0x7957: sar     ebp, 1
0x7959: lea     r13d, [r8 + rbp]
0x795d: movsxd  rbp, ebp
0x7960: call    0xf470
0x7965: movsxd  r13, r13d
0x7968: mov     rdi, rax
0x796b: call    0xc6a0
0x7970: mov     rdi, qword ptr [rip + 0x22f31]
0x7977: mov     r12d, eax
0x797a: call    0xf470
0x797f: mov     rdi, rax
0x7982: call    0xc6a0
0x7987: cmp     r12d, eax
0x798a: jg      0x7b20
0x7990: mov     eax, dword ptr [rip + 0x23006]
0x7996: mov     r8, r13
0x7999: xor     esi, esi
0x799b: mov     edx, 1
0x79a0: mov     rdi, qword ptr [rip + 0x22f09]
0x79a7: lea     ecx, [rax - 3]
0x79aa: movsxd  rcx, ecx
0x79ad: call    0xf590
0x79b2: mov     r8, rbp
0x79b5: mov     edx, 1
0x79ba: mov     rsi, r13
0x79bd: mov     eax, dword ptr [rip + 0x22fd9]
0x79c3: mov     rdi, qword ptr [rip + 0x22ede]
0x79ca: lea     ecx, [rax - 3]
0x79cd: movsxd  rcx, ecx
0x79d0: call    0xf590
0x79d5: jmp     0x7ad2
0x7ad2:
0x79da: nop     word ptr [rax + rax]
0x79e0: mov     r8d, 0x50
0x79e6: mov     ecx, 2
0x79eb: mov     edx, 0x10000
0x79f0: mov     esi, 0x10000
0x79f5: call    0xf590
0x79fa: mov     r8d, 0x50
0x7a00: mov     ecx, 2
0x7a05: mov     rdi, qword ptr [rip + 0x22e9c]

```



```

0x7a0c: mov     edx, 0x10000
0x7a11: mov     esi, 0x10000
0x7a16: call    0xf590
0x7a1b: xor     esi, esi
0x7a1d: xor     edi, edi
0x7a1f: call    0x16d30
0x7a24: mov     eax, dword ptr [rip + 0x22f72]
0x7a2a: xor     edi, edi
0x7a2c: lea     esi, [rax - 1]
0x7a2f: movsxd  rsi, esi
0x7a32: call    0xde50
0x7a37: mov     eax, dword ptr [rip + 0x22f5f]
0x7a3d: mov     edi, dword ptr [rip + 0x22f55]
0x7a43: cmp     eax, 1
0x7a46: lea     esi, [rax - 2]
0x7a49: mov     eax, 0
0x7a4e: cmov     esi, eax
0x7a51: add     rsp, 8
0x7a55: pop     rbx
0x7a56: pop     rbp
0x7a57: pop     r12
0x7a59: pop     r13
0x7a5b: jmp     0xc890
0xc890
0x7a60: imul    ecx, dword ptr [rip + 0x22f31]
0x7a67: mov     rdi, qword ptr [rip + 0x22e5a]
0x7a6e: lea     esi, [rcx + rcx + 4]
0x7a72: movsxd  rsi, esi
0x7a75: call    0x1d690
0x7a7a: mov     ecx, dword ptr [rip + 0x22f1c]
0x7a80: mov     qword ptr [rip + 0x22e41], rax
0x7a87: jmp     0x78d4
0x78d4
0x7a8c: nop     dword ptr [rax]
0x7a90: cmp     r8d, 5
0x7a94: jle     0x790d
0x7a9a: sub     ecx, 3
0x7a9d: mov     edx, 1
0x7aa2: xor     esi, esi
0x7aa4: movsxd  rcx, ecx
0x7aa7: call    0xf590
0x7aac: mov     eax, dword ptr [rip + 0x22eea]
0x7ab2: mov     edx, 1
0x7ab7: xor     esi, esi
0x7ab9: movsxd  r8, dword ptr [rip + 0x22ed8]
0x7ac0: mov     rdi, qword ptr [rip + 0x22de1]
0x7ac7: lea     ecx, [rax - 3]
0x7aca: movsxd  rcx, ecx

```

```

0x7acd: call    0xf590
0x7ad2: movsxd   rdi, dword ptr [rip + 0x22ebf]
0x7ad9: xor      esi, esi
0x7adb: call    0x16d30
0x7ae0: mov      eax, dword ptr [rip + 0x22eb6]
0x7ae6: lea      esi, [rax - 1]
0x7ae9: movsxd   rsi, esi
0x7aec: movsxd   rdi, dword ptr [rip + 0x22ea5]
0x7af3: jmp      0x7a32
0x7a32:
0x7af8: nop      dword ptr [rax + rax]
0x7b00: cmp      dword ptr [rip + 0x22e91], r12d
0x7b07: jne      0x78c6
0x7b0d: add      rsp, 8
0x7b11: pop      rbx
0x7b12: pop      rbp
0x7b13: pop      r12
0x7b15: pop      r13
0x7b17: ret
0x7b18: nop      dword ptr [rax + rax]
0x7b20: mov      eax, dword ptr [rip + 0x22e76]
0x7b26: mov      rdi, qword ptr [rip + 0x22d83]
0x7b2d: mov      r8, rbp
0x7b30: mov      rsi, r13
0x7b33: mov      edx, 1
0x7b38: lea      ecx, [rax - 3]
0x7b3b: movsxd   rcx, ecx
0x7b3e: call    0xf590
0x7b43: mov      eax, dword ptr [rip + 0x22e53]
0x7b49: mov      r8, r13
0x7b4c: xor      esi, esi
0x7b4e: mov      rdi, qword ptr [rip + 0x22d53]
0x7b55: mov      edx, 1
0x7b5a: lea      ecx, [rax - 3]
0x7b5d: movsxd   rcx, ecx
0x7b60: call    0xf590
0x7b65: jmp      0x7ad2
0x7ad2:
0x7b6a: nop      word ptr [rax + rax]
0x7b70: mov      r8d, 0x50
0x7b76: mov      ecx, 2
0x7b7b: mov      edx, 0x10000
0x7b80: mov      esi, 0x10000
0x7b85: call    0xf590
0x7b8a: mov      r8d, 0x50
0x7b90: mov      ecx, 2
0x7b95: mov      rdi, qword ptr [rip + 0x22d0c]
0x7b9c: mov      edx, 0x10000

```

```

0x7ba1: mov     esi, 0x10000
0x7ba6: call    0xf590
0x7bab: xor     esi, esi
0x7bad: xor     edi, edi
0x7baf: call    0x16d30
0x7bb4: mov     eax, dword ptr [rip + 0x22de2]
0x7bba: lea     esi, [rax - 1]
0x7bbd: movsxd  rsi, esi
0x7bc0: jmp     0x7aec
0x7aec:
0x7bc5: nop     word ptr cs:[rax + rax]
0x7bd0: endbr64
0x7bd4: push    rbx
0x7bd5: xor     eax, eax
0x7bd7: mov     ebx, edi
0x7bd9: xor     edi, edi
0x7bdb: call    0x7890
0x7be0: cmp     ebx, 0x12
0x7be3: je      0x7cc8
0x7be9: mov     esi, dword ptr [rip + 0x22ced]
0x7bef: test    esi, esi
0x7bf1: je      0x7bf8
0x7bf3: pop     rbx
0x7bf4: ret
0x7bf5: nop     dword ptr [rax]
0x7bf8: mov     rdi, qword ptr [rip + 0x22cb1]
0x7bff: call    0xf480
0x7c04: mov     rdi, qword ptr [rip + 0x22c9d]
0x7c0b: call    0xf480
0x7c10: mov     rdi, qword ptr [rip + 0x22c99]
0x7c17: call    0xf530
0x7c1c: mov     rdi, qword ptr [rip + 0x22c85]
0x7c23: call    0xf530
0x7c28: mov     ecx, dword ptr [rip + 0x22cb6]
0x7c2e: test    ecx, ecx
0x7c30: jne     0x7ce0
0x7c36: cmp     dword ptr [rip + 0x22d5f], 6
0x7c3d: jle     0x7c60
0x7c3f: mov     edx, dword ptr [rip + 0x22013]
0x7c45: mov     eax, dword ptr [rip + 0x22d4d]
0x7c4b: test    edx, edx
0x7c4d: je      0x7d00
0x7c53: cmp     eax, 0xb
0x7c56: jg      0x7c6e
0x7c58: nop     dword ptr [rax + rax]
0x7c60: xor     eax, eax
0x7c62: call    0xa6c0
0x7c67: xor     eax, eax

```

```

0x7c69: call    0xa300
0x7c6e: xor      eax, eax
0x7c70: call    0x16d50
0x7c75: mov     rdi, qword ptr [rip + 0x22c34]
0x7c7c: call    0x11010
0x7c81: mov     eax, dword ptr [rip + 0x21fd1]
0x7c87: test    eax, eax
0x7c89: jne     0x7cf0
0x7c8b: xor     eax, eax
0x7c8d: call    0xdea0
0x7c92: xor     eax, eax
0x7c94: call    0xd870
0x7c99: xor     eax, eax
0x7c9b: call    0xd740
0x7ca0: xor     eax, eax
0x7ca2: call    0xa820
0x7ca7: cmp     ebx, 0x12
0x7caa: jne     0x7bf3
0x7cb0: mov     rdi, qword ptr [rip + 0x22bf9]
0x7cb7: call    0x11100
0x7cbc: pop     rbx
0x7cbd: mov     rdi, rax
0x7cc0: jmp     0xc4b0
0xc4b0
0x7cc5: nop     dword ptr [rax]
0x7cc8: mov     edi, 1
0x7ccd: call    0x9b00
0x7cd2: xor     eax, eax
0x7cd4: call    0xa6c0
0x7cd9: jmp     0x7be9
0x7be9
0x7cde: nop
0x7ce0: mov     rdi, qword ptr [rip + 0x22be1]
0x7ce7: call    0xbb40
0x7cec: jmp     0x7c8b
0x7c8b
0x7cee: nop
0x7cf0: mov     rdi, qword ptr [rip + 0x22bb1]
0x7cf7: call    0x11010
0x7cfc: jmp     0x7c8b
0x7c8b
0x7cfe: nop
0x7d00: cmp     eax, 5
0x7d03: jle     0x7c60
0x7d09: jmp     0x7c6e
0x7c6e
0x7d0e: nop
0x7d10: endbr64

```

```

0x7d14: push    r12
0x7d16: xor      eax, eax
0x7d18: mov      r12, rdi
0x7d1b: push     rbp
0x7d1c: push     rbx
0x7d1d: call     0xb510
0x7d22: mov      rdi, rax
0x7d25: mov      rbp, rax
0x7d28: call     0x49b0
0x7d2d: test     rax, rax
0x7d30: je       0x7d39
0x7d32: cmp      byte ptr [rbp + rax - 1], 7
0x7d37: jne      0x7d70
0x7d39: xor      eax, eax
0x7d3b: call     0xa5a0
0x7d40: test     r12, r12
0x7d43: je       0x7ded
0x7d49: xor      edx, edx
0x7d4b: xor      esi, esi
0x7d4d: mov      rdi, r12
0x7d50: call     0xe480
0x7d55: xor      eax, eax
0x7d57: call     0xd740
0x7d5c: pop      rbx
0x7d5d: xor      eax, eax
0x7d5f: pop      rbp
0x7d60: pop      r12
0x7d62: jmp      0xa820
0xa820:
0x7d67: nop      word ptr [rax + rax]
0x7d70: mov      rdi, rbp
0x7d73: call     0x9fe0
0x7d78: mov      rdi, rax
0x7d7b: mov      rbx, rax
0x7d7e: call     0x49b0
0x7d83: lea      rdi, [rax + 0x80]
0x7d8a: call     0x1d670
0x7d8f: mov      r8, rbx
0x7d92: mov      esi, 1
0x7d97: lea      rcx, [rip + 0x18762]
0x7d9e: mov      rbp, rax
0x7da1: mov      rdi, rax
0x7da4: mov      rdx, -1
0x7dab: xor      eax, eax
0x7dad: call     0x4f10
0x7db2: mov      edx, 1
0x7db7: mov      esi, 2
0x7dbc: mov      rdi, rbp

```

```

0x7dbf: call    0xe480
0x7dc4: mov     rdi, rbp
0x7dc7: call    0x18d10
0x7dcc: xor     eax, eax
0x7dce: call    0xa5a0
0x7dd3: xor     eax, eax
0x7dd5: call    0xa820
0x7dda: mov     edi, 1
0x7ddf: call    0x4ea0
0x7de4: test    r12, r12
0x7de7: jne     0x7d49
0x7ded: xor     eax, eax
0x7def: call    0xe4e0
0x7df4: jmp     0x7d55
0x7d55:
0x7df9: nop     dword ptr [rax]
0x7e00: endbr64
0x7e04: push    r13
0x7e06: mov     edx, 1
0x7e0b: push    r12
0x7e0d: mov     r12, rsi
0x7e10: push    rbp
0x7e11: mov     rbp, rdi
0x7e14: sub     rsp, 0x10
0x7e18: mov     rax, qword ptr fs:[0x28]
0x7e21: mov     qword ptr [rsp + 8], rax
0x7e26: xor     eax, eax
0x7e28: mov     r13, rsp
0x7e2b: mov     rsi, r13
0x7e2e: call    0x19200
0x7e33: test    rax, rax
0x7e36: je      0x7e48
0x7e38: mov     rdi, qword ptr [rsp]
0x7e3c: mov     rsi, r12
0x7e3f: call    0x4b30
0x7e44: test    eax, eax
0x7e46: je      0x7e5f
0x7e48: mov     rdi, r12
0x7e4b: call    0x1d7a0
0x7e50: mov     rsi, r13
0x7e53: mov     rdi, rbp
0x7e56: mov     qword ptr [rsp], rax
0x7e5a: call    0x19150
0x7e5f: mov     rax, qword ptr [rsp + 8]
0x7e64: sub     rax, qword ptr fs:[0x28]
0x7e6d: jne     0x7e79
0x7e6f: add     rsp, 0x10
0x7e73: pop     rbp

```

```

0x7e74: pop      r12
0x7e76: pop      r13
0x7e78: ret
0x7e79: call     0x49d0
0x7e7e: nop
0x7e80: endbr64
0x7e84: sub       rsp, 0x18
0x7e88: mov        edx, esi
0x7e8a: mov        rax, qword ptr fs:[0x28]
0x7e93: mov        qword ptr [rsp + 8], rax
0x7e98: xor        eax, eax
0x7e9a: mov        rsi, rsp
0x7e9d: call     0x19200
0x7ea2: test       rax, rax
0x7ea5: cmovne    rax, qword ptr [rsp]
0x7eaa: mov        rdx, qword ptr [rsp + 8]
0x7eaf: sub        rdx, qword ptr fs:[0x28]
0x7eb8: jne        0x7ebf
0x7eba: add        rsp, 0x18
0x7ebe: ret
0x7ebf: call     0x49d0
0x7ec4: nop        word ptr cs:[rax + rax]
0x7ecf: nop
0x7ed0: endbr64
0x7ed4: test       sil, 2
0x7ed8: je         0x7ee0
0x7eda: jmp        0x7570
0x7570:
0x7edf: nop
0x7ee0: xor        eax, eax
0x7ee2: ret
0x7ee3: nop        word ptr cs:[rax + rax]
0x7eee: nop
0x7ef0: endbr64
0x7ef4: push       r15
0x7ef6: push       r14
0x7ef8: push       r13
0x7efa: push       r12
0x7efc: push       rbp
0x7efd: push       rbx
0x7efe: sub        rsp, 8
0x7f02: test       rdi, rdi
0x7f05: je         0x7fc8
0x7f0b: mov        rbp, rdi
0x7f0e: call     0x49b0
0x7f13: lea        r13, [rax + 1]
0x7f17: mov        rdi, r13
0x7f1a: call     0x1d670

```

```

0x7f1f: movzx    ebx, byte ptr [rbp]
0x7f23: mov      r15, rax
0x7f26: test     bl, bl
0x7f28: je       0x7fd0
0x7f2e: add      rbp, 1
0x7f32: xor      r12d, r12d
0x7f35: jmp      0x7f6d
0x7f6d
0x7f37: nop      word ptr [rax + rax]
0x7f40: call     0x4f00
0x7f45: movsx    rdx, bl
0x7f49: add      r12d, 1
0x7f4d: mov      rax, qword ptr [rax]
0x7f50: test     byte ptr [rax + rdx*2 + 1], 0x40
0x7f55: mov      eax, 0x3f
0x7f5a: cmov     ebx, eax
0x7f5d: add      rbp, 1
0x7f61: mov      byte ptr [r15 + r14], bl
0x7f65: movzx    ebx, byte ptr [rbp - 1]
0x7f69: test     bl, bl
0x7f6b: je       0x7fa5
0x7f6d: movsxd   r14, r12d
0x7f70: cmp      bl, 9
0x7f73: jne      0x7f40
0x7f75: add      r13, 8
0x7f79: mov      rdi, r15
0x7f7c: add      rbp, 1
0x7f80: add      r12d, 8
0x7f84: mov      rsi, r13
0x7f87: call     0x1d690
0x7f8c: mov      r15, rax
0x7f8f: movabs   rax, 0x2020202020202020
0x7f99: mov      qword ptr [r15 + r14], rax
0x7f9d: movzx    ebx, byte ptr [rbp - 1]
0x7fa1: test     bl, bl
0x7fa3: jne      0x7f6d
0x7fa5: movsxd   r12, r12d
0x7fa8: add      r12, r15
0x7fab: mov      byte ptr [r12], 0
0x7fb0: add      rsp, 8
0x7fb4: mov      rax, r15
0x7fb7: pop      rbx
0x7fb8: pop      rbp
0x7fb9: pop      r12
0x7fbb: pop      r13
0x7fbd: pop      r14
0x7fbf: pop      r15
0x7fc1: ret

```



```

0x7fc2: nop      word ptr [rax + rax]
0x7fc8: xor       r15d, r15d
0x7fcb: jmp       0x7fb0
0x7fb0:
0x7fcd: nop      dword ptr [rax]
0x7fd0: mov       r12, rax
0x7fd3: jmp       0x7fab
0x7fab:
0x7fd5: nop      word ptr cs:[rax + rax]
0x7fe0: endbr64
0x7fe4: push     rbx
0x7fe5: mov      rbx, rdi
0x7fe8: call     0x49b0
0x7fed: lea      rdi, [rax + rax*2 + 9]
0x7ff2: call     0x1d670
0x7ff7: mov     byte ptr [rax], 0x28
0x7ffa: cmp     byte ptr [rbx + 1], 0
0x7ffe: je      0x8050
0x8000: mov     edx, 1
0x8005: nop     dword ptr [rax]
0x8008: movzx   ecx, byte ptr [rbx]
0x800b: mov     byte ptr [rax + rdx + 1], 0x2c
0x8010: add     rbx, 1
0x8014: mov     byte ptr [rax + rdx], cl
0x8017: mov     rcx, rdx
0x801a: add     rdx, 3
0x801e: mov     byte ptr [rax + rdx - 1], 0x20
0x8023: cmp     byte ptr [rbx + 1], 0
0x8027: jne     0x8008
0x8029: lea     r8, [rcx + 4]
0x802d: lea     rdi, [rcx + 5]
0x8031: add     rcx, 6
0x8035: movzx   esi, byte ptr [rbx]
0x8038: mov     byte ptr [rax + rdx], sil
0x803c: mov     byte ptr [rax + r8], 0x29
0x8041: mov     byte ptr [rax + rdi], 0x20
0x8045: mov     byte ptr [rax + rcx], 0
0x8049: pop     rbx
0x804a: ret
0x804b: nop     dword ptr [rax + rax]
0x8050: mov     ecx, 4
0x8055: mov     edi, 3
0x805a: mov     r8d, 2
0x8060: mov     edx, 1
0x8065: jmp     0x8035
0x8035:
0x8067: nop     word ptr [rax + rax]
0x8070: endbr64

```

```

0x8074: push    r15
0x8076: push    r14
0x8078: push    r13
0x807a: push    r12
0x807c: mov     r12, rdi
0x807f: push    rbp
0x8080: mov     rbp, rsi
0x8083: push    rbx
0x8084: mov     ebx, edx
0x8086: sub     rsp, 0x28
0x808a: mov     rax, qword ptr fs:[0x28]
0x8093: mov     qword ptr [rsp + 0x18], rax
0x8098: mov     eax, edx
0x809a: and     eax, 0x10
0x809d: mov     dword ptr [rsp + 0xc], eax
0x80a1: jne     0x82f0
0x80a7: xor     r13d, r13d
0x80aa: xor     eax, eax
0x80ac: call    0xcff0
0x80b1: test    r12, r12
0x80b4: je     0x8103
0x80b6: mov     rdi, r12
0x80b9: call    0x7ef0
0x80be: mov     r12, rax
0x80c1: test    bl, 8
0x80c4: jne     0x8330
0x80ca: mov     rdi, r12
0x80cd: call    0xd610
0x80d2: test    bl, 0x20
0x80d5: jne     0x8310
0x80db: mov     rdi, r12
0x80de: call    0x18d10
0x80e3: test    rbp, rbp
0x80e6: je     0x8103
0x80e8: mov     rdi, rbp
0x80eb: call    0x7fe0
0x80f0: mov     r12, rax
0x80f3: mov     rdi, rax
0x80f6: call    0xd610
0x80fb: mov     rdi, r12
0x80fe: call    0x18d10
0x8103: xor     eax, eax
0x8105: call    0xd870
0x810a: xor     eax, eax
0x810c: call    0xd740
0x8111: xor     eax, eax
0x8113: call    0xa820
0x8118: test    bl, 4

```

```

0x811b: jne      0x8300
0x8121: mov      r14d, ebx
0x8124: lea       r12, [rsp + 0x14]
0x8129: and       r14d, 2
0x812d: mov      rdi, r12
0x8130: call     0xb180
0x8135: test     rax, rax
0x8138: je       0x81b2
0x813a: mov      rdx, qword ptr [rax]
0x813d: mov      rax, qword ptr [rax + 0x10]
0x8141: movzx    r15d, byte ptr [rdx]
0x8145: test     rax, rax
0x8148: je       0x8250
0x814e: cmp      byte ptr [rax + 0x1b], 0
0x8152: je       0x8250
0x8158: mov      rcx, qword ptr [rax]
0x815b: lea      rax, [rip + 0x20ebe]
0x8162: sub      rcx, rax
0x8165: movabs   rax, 0xea0ea0ea0ea0ea1
0x816f: imul     rcx
0x8172: sar      rcx, 0x3f
0x8176: sar      rdx, 1
0x8179: sub      rdx, rcx
0x817c: mov      r15d, edx
0x817f: not      r15d
0x8182: cmp      edx, 7
0x8185: jne      0x81d0
0x8187: xor      edi, edi
0x8189: xor      eax, eax
0x818b: call     0x7bd0
0x8190: xor      eax, eax
0x8192: call     0xdea0
0x8197: xor      eax, eax
0x8199: call     0xd740
0x819e: xor      eax, eax
0x81a0: call     0xa820
0x81a5: mov      rdi, r12
0x81a8: call     0xb180
0x81ad: test     rax, rax
0x81b0: jne      0x813a
0x81b2: xor      eax, eax
0x81b4: call     0xa5a0
0x81b9: xor      eax, eax
0x81bb: call     0xdea0
0x81c0: xor      eax, eax
0x81c2: call     0xd740
0x81c7: jmp      0x812d
0x812d

```

```

0x81cc: nop      dword ptr [rax]
0x81d0: cmp      r15d, 7
0x81d4: je       0x82b0
0x81da: cmp      r15d, -0x3d
0x81de: jne      0x8256
0x81e0: test     rbp, rbp
0x81e3: je       0x8360
0x81e9: xor      edi, edi
0x81eb: movsx   r15d, byte ptr [rbp]
0x81f0: call     0xdba0
0x81f5: mov      esi, dword ptr [rsp + 0xc]
0x81f9: test     esi, esi
0x81fb: je       0x821f
0x81fd: test     r13, r13
0x8200: je       0x821f
0x8202: mov      rdi, r13
0x8205: call     0xc970
0x820a: xor      eax, eax
0x820c: call     0xd870
0x8211: xor      eax, eax
0x8213: call     0xd740
0x8218: xor      eax, eax
0x821a: call     0xa820
0x821f: cmp      r15d, 7
0x8223: je       0x82e1
0x8229: mov      eax, r15d
0x822c: mov      rdx, qword ptr [rsp + 0x18]
0x8231: sub      rdx, qword ptr fs:[0x28]
0x823a: jne      0x83d9
0x8240: add      rsp, 0x28
0x8244: pop      rbx
0x8245: pop      rbp
0x8246: pop      r12
0x8248: pop      r13
0x824a: pop      r14
0x824c: pop      r15
0x824e: ret
0x824f: nop
0x8250: cmp      r15d, 7
0x8254: je       0x82b0
0x8256: mov      eax, dword ptr [rsp + 0x14]
0x825a: lea      edx, [rax - 1]
0x825d: mov      dword ptr [rsp + 0x14], edx
0x8261: test     eax, eax
0x8263: je       0x83a8
0x8269: test     r14d, r14d
0x826c: je       0x8280
0x826e: mov      esi, ebx

```

```

0x8270: mov     edi, r15d
0x8273: call    0x7570
0x8278: test     eax, eax
0x827a: jne      0x8256
0x827c: nop      dword ptr [rax]
0x8280: test     rbp, rbp
0x8283: je       0x83b8
0x8289: mov     esi, r15d
0x828c: mov     rdi, rbp
0x828f: call    0x4a00
0x8294: test     rax, rax
0x8297: jne      0x83b8
0x829d: call    0xa5a0
0x82a2: jmp      0x8190
0x8190
0x82a7: nop      word ptr [rax + rax]
0x82b0: xor      edi, edi
0x82b2: call    0xdba0
0x82b7: mov     ecx, dword ptr [rsp + 0xc]
0x82bb: test     ecx, ecx
0x82bd: je       0x82e1
0x82bf: test     r13, r13
0x82c2: je       0x82e1
0x82c4: mov     rdi, r13
0x82c7: call    0xc970
0x82cc: xor      eax, eax
0x82ce: call    0xd870
0x82d3: xor      eax, eax
0x82d5: call    0xd740
0x82da: xor      eax, eax
0x82dc: call    0xa820
0x82e1: xor      eax, eax
0x82e3: jmp      0x822c
0x822c
0x82e8: nop      dword ptr [rax + rax]
0x82f0: xor      eax, eax
0x82f2: call    0xc8c0
0x82f7: mov     r13, rax
0x82fa: jmp      0x80aa
0x80aa
0x82ff: nop
0x8300: xor      eax, eax
0x8302: call    0xa5a0
0x8307: jmp      0x8121
0x8121
0x830c: nop      dword ptr [rax]
0x8310: xor      eax, eax
0x8312: call    0xccf0

```

```

0x8317: mov     rdi, r12
0x831a: call    0x18d10
0x831f: test    rbp, rbp
0x8322: jne     0x80e8
0x8328: jmp     0x8103
0x8103
0x832d: nop     dword ptr [rax]
0x8330: lea     rdi, [rip + 0x181da]
0x8337: call    0xd610
0x833c: mov     edi, 1
0x8341: call    0xdba0
0x8346: mov     rdi, r12
0x8349: call    0xd610
0x834e: test    bl, 0x20
0x8351: je      0x80db
0x8357: jmp     0x8310
0x8310
0x8359: nop     dword ptr [rax]
0x8360: xor     edi, edi
0x8362: call    0xdba0
0x8367: mov     edx, dword ptr [rsp + 0xc]
0x836b: mov     eax, 0xffffffffc3
0x8370: test    edx, edx
0x8372: je      0x822c
0x8378: test    r13, r13
0x837b: je      0x822c
0x8381: mov     rdi, r13
0x8384: call    0xc970
0x8389: xor     eax, eax
0x838b: call    0xd870
0x8390: xor     eax, eax
0x8392: call    0xd740
0x8397: xor     eax, eax
0x8399: call    0xa820
0x839e: jmp     0x8229
0x8229
0x83a3: nop     dword ptr [rax + rax]
0x83a8: call    0xd870
0x83ad: jmp     0x8190
0x8190
0x83b2: nop     word ptr [rax + rax]
0x83b8: xor     edi, edi
0x83ba: call    0xdba0
0x83bf: mov     eax, dword ptr [rsp + 0xc]
0x83c3: test    eax, eax
0x83c5: je      0x8229
0x83cb: test    r13, r13
0x83ce: jne     0x8202

```

```

0x83d4: jmp      0x8229
0x8229
0x83d9: call     0x49d0
0x83de: nop
0x83e0: endbr64
0x83e4: push     r15
0x83e6: push     r14
0x83e8: mov      r14, rdi
0x83eb: push     r13
0x83ed: mov      r13, rsi
0x83f0: push     r12
0x83f2: mov      r12, rdx
0x83f5: push     rbp
0x83f6: mov      rbp, rcx
0x83f9: push     rbx
0x83fa: sub      rsp, 0x38
0x83fe: mov      rax, qword ptr fs:[0x28]
0x8407: mov      qword ptr [rsp + 0x28], rax
0x840c: xor      eax, eax
0x840e: call     0xcff0
0x8413: test     r14, r14
0x8416: je       0x8420
0x8418: mov      rdi, r14
0x841b: call     0xd540
0x8420: test     r12, r12
0x8423: je       0x8450
0x8425: mov      rdi, r12
0x8428: call     0xd610
0x842d: test     rbp, rbp
0x8430: je       0x8450
0x8432: mov      rsi, r12
0x8435: mov      rdi, rbp
0x8438: xor      eax, eax
0x843a: mov      r14d, 1
0x8440: call     0x7e00
0x8445: jmp      0x8453
0x8453
0x8447: nop      word ptr [rax + rax]
0x8450: xor      r14d, r14d
0x8453: xor      eax, eax
0x8455: lea      r12, [rsp + 0x1c]
0x845a: lea      rbx, [rip + 0x17c03]
0x8461: call     0xd870
0x8466: xor      eax, eax
0x8468: call     0xd740
0x846d: xor      eax, eax
0x846f: call     0xa820
0x8474: lea      rax, [rsp + 0x20]

```

```

0x8479: mov     qword ptr [rsp], rax
0x847d: mov     rdi, r12
0x8480: call    0xb180
0x8485: test     rax, rax
0x8488: je       0x853f
0x848e: mov     rdx, qword ptr [rax]
0x8491: mov     rax, qword ptr [rax + 0x10]
0x8495: movzx   r15d, byte ptr [rdx]
0x8499: test     rax, rax
0x849c: je       0x84d7
0x849e: cmp     byte ptr [rax + 0x1b], 0
0x84a2: je       0x84d7
0x84a4: mov     rcx, qword ptr [rax]
0x84a7: lea     rax, [rip + 0x20b72]
0x84ae: sub     rcx, rax
0x84b1: movabs  rax, 0xea0ea0ea0ea0ea1
0x84bb: imul    rcx
0x84be: sar     rcx, 0x3f
0x84c2: sar     rdx, 1
0x84c5: sub     rdx, rcx
0x84c8: mov     r15d, edx
0x84cb: not     r15d
0x84ce: cmp     edx, 0x3c
0x84d1: je       0x8560
0x84d7: lea     eax, [r15 + 0x22]
0x84db: cmp     eax, 0x29
0x84de: ja      0x8505
0x84e0: movsxd  rax, dword ptr [rbx + rax*4]
0x84e4: add     rax, rbx
0x84e7: jmp     rax
rax
->->->                                indirect JMP
0x1e1a4: test     r8b, r8b
0x1e1a7: je       0x1e9cf
0x1e1ad: nop     dword ptr [rax]
0x1e1b0: mov     edx, ebp
0x1e1b2: mov     eax, 1
0x1e1b7: mov     ecx, ebp
0x1e1b9: mov     r12, r15
0x1e1bc: shr     dl, 5
0x1e1bf: shl     eax, cl
0x1e1c1: movzx   edx, dl
0x1e1c4: mov     rsi, qword ptr [rsp + 0x18]
0x1e1c9: and     eax, dword ptr [rsi + rdx*4 + 4]
0x1e1cd: je       0x1e1e1
0x1e1cf: cmp     rbx, r10
0x1e1d2: jae     0x1e1dd
0x1e1d4: mov     rax, qword ptr [rsp + 8]

```



```

0x1e1d9:      mov     byte ptr [rax + rbx], 0x5c
0x1e1dd:      add     rbx, 1
0x1e1e1:      cmp     rbx, r10
0x1e1e4:      jae     0x1e1ef
0x1e1e6:      mov     rax, qword ptr [rsp + 8]
0x1e1eb:      mov     byte ptr [rax + rbx], bpl
0x1e1ef:      add     rbx, 1
0x1e1f3:      jmp     0x1e0c0
0x1e0c0
0x1e1f8:      cmp     dl, 0x7b
0x1e1fb:      je      0x1e72c
0x1e201:      cmp     dl, 0x7c
0x1e204:      jne     0x1e500
0x1e20a:      cmp     dword ptr [rsp + 0x10], 1
0x1e20f:      je      0x1e380
0x1e215:      lea     r15, [r12 + 1]
0x1e21a:      jmp     0x1e1a4
0x1e1a4
0x1e21c:      nop     dword ptr [rax]
0x1e220:      cmp     dl, 0x7d
0x1e223:      je      0x1e72c
0x1e229:      jle     0x1e727
0x1e22f:      cmp     dl, 0x7e
0x1e232:      je      0x1e365
0x1e238:      nop     dword ptr [rax + rax]
0x1e240:      cmp     qword ptr [rsp + 0x28], 1
0x1e246:      mov     byte ptr [rsp + 0x30], dl
0x1e24a:      jne     0x1e518
0x1e250:      mov     qword ptr [rsp + 0x50], r11
0x1e255:      mov     qword ptr [rsp + 0x48], r10
0x1e25a:      mov     byte ptr [rsp + 0x40], r8b
0x1e25f:      mov     qword ptr [rsp + 0x38], r9
0x1e264:      call    0x4f00
0x1e269:      movzx   edx, byte ptr [rsp + 0x30]
0x1e26e:      movzx   r8d, byte ptr [rsp + 0x40]
0x1e274:      mov     ecx, 1
0x1e279:      mov     rax, qword ptr [rax]
0x1e27c:      mov     r9, qword ptr [rsp + 0x38]
0x1e281:      mov     r10, qword ptr [rsp + 0x48]
0x1e286:      mov     r11, qword ptr [rsp + 0x50]
0x1e28b:      movzx   eax, word ptr [rax + rdx*2]
0x1e28f:      shr     ax, 0xe
0x1e293:      xor     eax, 1
0x1e296:      and     eax, r8d
0x1e299:      test    al, al
0x1e29b:      je      0x1e1a4
0x1e2a1:      mov     eax, r8d
0x1e2a4:      mov     rsi, qword ptr [rsp + 8]

```

```

0x1e2a9:      add     rcx, r12
0x1e2ac:      jmp     0x1e2c6
0x1e2c6
0x1e2ae:      nop
0x1e2b0:      cmp     rbx, r10
0x1e2b3:      jae     0x1e2b9
0x1e2b5:      mov     byte ptr [rsi + rbx], bpl
0x1e2b9:      movzx   ebp, byte ptr [r11 + r14]
0x1e2be:      add     rbx, 1
0x1e2c2:      add     r14, 1
0x1e2c6:      test    al, al
0x1e2c8:      je      0x1e30a
0x1e2ca:      cmp     rbx, r10
0x1e2cd:      jae     0x1e2d3
0x1e2cf:      mov     byte ptr [rsi + rbx], 0x5c
0x1e2d3:      lea     rdx, [rbx + 1]
0x1e2d7:      cmp     r10, rdx
0x1e2da:      jbe     0x1e2e8
0x1e2dc:      mov     edx, ebp
0x1e2de:      shr     dl, 6
0x1e2e1:      add     edx, 0x30
0x1e2e4:      mov     byte ptr [rsi + rbx + 1], dl
0x1e2e8:      lea     rdx, [rbx + 2]
0x1e2ec:      cmp     r10, rdx
0x1e2ef:      jbe     0x1e300
0x1e2f1:      mov     edx, ebp
0x1e2f3:      shr     dl, 3
0x1e2f6:      and     edx, 7
0x1e2f9:      add     edx, 0x30
0x1e2fc:      mov     byte ptr [rsi + rbx + 2], dl
0x1e300:      and     ebp, 7
0x1e303:      add     rbx, 3
0x1e307:      add     ebp, 0x30
0x1e30a:      mov     r12, r14
0x1e30d:      cmp     r14, rcx
0x1e310:      jb      0x1e2b0
0x1e312:      jmp     0x1e1e1
0x1e1e1
0x1e317:      nop     word ptr [rax + rax]
0x1e320:      mov     eax, dword ptr [rsp + 0x10]
0x1e324:      cmp     eax, 1
0x1e327:      je      0x1e380
0x1e329:      cmp     eax, 3
0x1e32c:      je      0x1e760
0x1e332:      lea     r15, [r12 + 1]
0x1e337:      mov     ebp, 0x3f
0x1e33c:      jmp     0x1e1a4
0x1e1a4

```

```

0x1e341:    mov     ebp, 0x72
0x1e346:    cmp     dword ptr [rsp + 0x10], 1
0x1e34b:    je      0x1e380
0x1e34d:    mov     r12, r14
0x1e350:    jmp     0x1e1cf
0x1e1cf
0x1e355:    mov     ebp, 0x62
0x1e35a:    jmp     0x1e34d
0x1e34d
0x1e35c:    nop     dword ptr [rax]
0x1e360:    mov     ebp, 0x23
0x1e365:    test    r12, r12
0x1e368:    jne     0x1e215
0x1e36e:    cmp     dword ptr [rsp + 0x10], 1
0x1e373:    jne     0x1e215
0x1e379:    nop     dword ptr [rax]
0x1e380:    mov     qword ptr [rsp + 0x20], r11
0x1e385:    mov     qword ptr [rsp + 0x10], r10
0x1e38a:    call    0x49a0
0x1e38f:    mov     r10, qword ptr [rsp + 0x10]
0x1e394:    mov     r11, qword ptr [rsp + 0x20]
0x1e399:    mov     qword ptr [rsp + 0x28], rax
0x1e39e:    lea     rax, [rip + 0x43b7]
0x1e3a5:    mov     dword ptr [rsp + 0x10], 2
0x1e3ad:    mov     r9d, 1
0x1e3b3:    xor     r8d, r8d
0x1e3b6:    mov     qword ptr [rsp + 0x20], rax
0x1e3bb:    test    r10, r10
0x1e3be:    je      0x1e0b0
0x1e3c4:    mov     rax, qword ptr [rsp + 8]
0x1e3c9:    mov     byte ptr [rax], 0x27
0x1e3cc:    jmp     0x1e0b0
0x1e0b0
0x1e3d1:    mov     ebp, 0x74
0x1e3d6:    jmp     0x1e346
0x1e346
0x1e3db:    mov     ebp, 0x66
0x1e3e0:    jmp     0x1e34d
0x1e34d
0x1e3e5:    mov     r12, r14
0x1e3e8:    mov     ebp, 0x76
0x1e3ed:    jmp     0x1e1cf
0x1e1cf
0x1e3f2:    mov     r12, r14
0x1e3f5:    cmp     r10, rbx
0x1e3f8:    jbe     0x1e403
0x1e3fa:    mov     rax, qword ptr [rsp + 8]
0x1e3ff:    mov     byte ptr [rax + rbx], 0x5c

```

```

0x1e403:    lea     rax, [rbx + 1]
0x1e407:    cmp     r10, rax
0x1e40a:    jbe     0x1e416
0x1e40c:    mov     rax, qword ptr [rsp + 8]
0x1e411:    mov     byte ptr [rax + rbx + 1], 0x30
0x1e416:    lea     rax, [rbx + 2]
0x1e41a:    cmp     r10, rax
0x1e41d:    jbe     0x1e429
0x1e41f:    mov     rax, qword ptr [rsp + 8]
0x1e424:    mov     byte ptr [rax + rbx + 2], 0x30
0x1e429:    add     rbx, 3
0x1e42d:    mov     eax, 0x10000
0x1e432:    mov     edx, 1
0x1e437:    mov     ebp, 0x30
0x1e43c:    jmp     0x1e1c4
0x1e1c4
0x1e441:    mov     ebp, 0x61
0x1e446:    jmp     0x1e34d
0x1e34d
0x1e44b:    mov     ebp, 0x6e
0x1e450:    jmp     0x1e346
0x1e346
0x1e455:    nop     dword ptr [rax]
0x1e458:    cmp     dl, 0x7a
0x1e45b:    jg      0x1e220
0x1e461:    cmp     dl, 0x40
0x1e464:    je      0x1e240
0x1e46a:    lea     ecx, [rbp - 0x41]
0x1e46d:    mov     eax, 1
0x1e472:    movabs  rdi, 0x3fffffff53ffffff
0x1e47c:    shl     rax, cl
0x1e47f:    test    rax, rdi
0x1e482:    jne     0x1e1b0
0x1e488:    test    eax, 0xa4000000
0x1e48d:    jne     0x1e20a
0x1e493:    cmp     bpl, 0x5c
0x1e497:    jne     0x1e240
0x1e49d:    cmp     dword ptr [rsp + 0x10], 1
0x1e4a2:    je      0x1e380
0x1e4a8:    jmp     0x1e34d
0x1e34d
0x1e4ad:    nop     dword ptr [rax]
0x1e4b0:    mov     ebp, edx
0x1e4b2:    cmp     dl, 0x3f
0x1e4b5:    jg      0x1e6d8
0x1e4bb:    test    dl, dl
0x1e4bd:    js      0x1e500
0x1e4bf:    cmp     dl, 0x3f

```

```

0x1e4c2:      ja      0x1e500
0x1e4c4:      lea     rdi, [rip + 0x43c5]
0x1e4cb:      movzx   eax, dl
0x1e4ce:      movsxd  rax, dword ptr [rdi + rax*4]
0x1e4d2:      add     rax, rdi
0x1e4d5:      jmp     rax
rax
->->->                                indirect JMP
0x1e630:      cmp     rcx, 1
0x1e634:      ja      0x1e2a4
0x1e63a:      jmp     0x1e299
0x1e299
0x1e63f:      mov     ebp, 0xc
0x1e644:      mov     eax, 0x66
0x1e649:      nop     dword ptr [rax]
0x1e650:      add     r12, 1
0x1e654:      test    r8b, r8b
0x1e657:      je      0x1e1e1
0x1e65d:      mov     ebp, eax
0x1e65f:      jmp     0x1e1cf
0x1e1cf
0x1e664:      nop     dword ptr [rax]
0x1e668:      mov     ebp, 0xb
0x1e66d:      mov     eax, 0x76
0x1e672:      jmp     0x1e650
0x1e650
0x1e674:      mov     ebp, 9
0x1e679:      mov     eax, 0x74
0x1e67e:      cmp     dword ptr [rsp + 0x10], 1
0x1e683:      jne     0x1e650
0x1e685:      jmp     0x1e380
0x1e380
0x1e68a:      nop     word ptr [rax + rax]
0x1e690:      add     r12, 1
0x1e694:      test    r8b, r8b
0x1e697:      jne     0x1e3f5
0x1e69d:      xor     ebp, ebp
0x1e69f:      jmp     0x1e1e1
0x1e1e1
0x1e6a4:      mov     ebp, 8
0x1e6a9:      mov     eax, 0x62
0x1e6ae:      jmp     0x1e650
0x1e650
0x1e6b0:      mov     ebp, 0xa
0x1e6b5:      mov     eax, 0x6e
0x1e6ba:      jmp     0x1e67e
0x1e67e
0x1e6bc:      mov     ebp, 7

```

```

0x1e6c1:      mov     eax, 0x61
0x1e6c6:      jmp     0x1e650
0x1e650:
0x1e6c8:      mov     ebp, 0xd
0x1e6cd:      mov     eax, 0x72
0x1e6d2:      jmp     0x1e67e
0x1e67e:
0x1e6d4:      nop     dword ptr [rax]
0x1e6d8:      cmp     dl, 0x7a
0x1e6db:      jg      0x1e4e0
0x1e6e1:      cmp     dl, 0x40
0x1e6e4:      je      0x1e500
0x1e6ea:      lea     ecx, [rbp - 0x41]
0x1e6ed:      mov     eax, 1
0x1e6f2:      movabs  rdi, 0x3fffffff53ffffff
0x1e6fc:      shl     rax, cl
0x1e6ff:      test    rax, rdi
0x1e702:      jne     0x1e215
0x1e708:      test    eax, 0xa4000000
0x1e70d:      jne     0x1e20a
0x1e713:      cmp     bpl, 0x5c
0x1e717:      jne     0x1e500
0x1e71d:      mov     eax, 0x5c
0x1e722:      jmp     0x1e67e
0x1e67e:
0x1e727:      cmp     dl, 0x7b
0x1e72a:      jne     0x1e750
0x1e72c:      cmp     r13, 1
0x1e730:      setne   al
0x1e733:      cmp     r13, -1
0x1e737:      jne     0x1e741
0x1e739:      cmp     byte ptr [r11 + 1], 0
0x1e73e:      setne   al
0x1e741:      test    al, al
0x1e743:      jne     0x1e215
0x1e749:      jmp     0x1e365
0x1e365:
0x1e74e:      nop
0x1e750:      cmp     dl, 0x7c
0x1e753:      je      0x1e20a
0x1e759:      jmp     0x1e240
0x1e240:
0x1e75e:      nop
0x1e760:      lea     rax, [r12 + 2]
0x1e765:      lea     r15, [r12 + 1]
0x1e76a:      mov     ebp, 0x3f
0x1e76f:      cmp     rax, r13
0x1e772:      jae     0x1e1a4

```

```

0x1e778:      cmp     byte ptr [r11 + r12 + 1], 0x3f
0x1e77e:      jne     0x1e1a4
0x1e784:      movzx   ebp, byte ptr [r11 + r12 + 2]
0x1e78a:      cmp     bpl, 0x3e
0x1e78e:      ja      0x1e7a4
0x1e790:      movabs   rax, 0x7000a38200000000
0x1e79a:      bt      rax, rbp
0x1e79e:      jb      0x1e8f9
0x1e7a4:      mov     ebp, 0x3f
0x1e7a9:      jmp     0x1e1a4
0x1e1a4:
0x1e7ae:      test    r12, r12
0x1e7b1:      jne     0x1e7be
0x1e7b3:      cmp     dword ptr [rsp + 0x10], 1
0x1e7b8:      je      0x1e380
0x1e7be:      mov     rsi, qword ptr [rsp + 0x20]
0x1e7c3:      mov     rax, rbx
0x1e7c6:      test    rsi, rsi
0x1e7c9:      je      0x1e7f4
0x1e7cb:      movzx   edx, byte ptr [rsi]
0x1e7ce:      test    dl, dl
0x1e7d0:      je      0x1e7f4
0x1e7d2:      sub     rsi, rbx
0x1e7d5:      mov     rcx, rsi
0x1e7d8:      mov     rsi, qword ptr [rsp + 8]
0x1e7dd:      nop     dword ptr [rax]
0x1e7e0:      cmp     r10, rax
0x1e7e3:      jbe     0x1e7e8
0x1e7e5:      mov     byte ptr [rsi + rax], dl
0x1e7e8:      add     rax, 1
0x1e7ec:      movzx   edx, byte ptr [rcx + rax]
0x1e7f0:      test    dl, dl
0x1e7f2:      jne     0x1e7e0
0x1e7f4:      cmp     rax, r10
0x1e7f7:      jae     0x1e802
0x1e7f9:      mov     rsi, qword ptr [rsp + 8]
0x1e7fe:      mov     byte ptr [rsi + rax], 0
0x1e802:      mov     rdx, qword ptr [rsp + 0x78]
0x1e807:      sub     rdx, qword ptr fs:[0x28]
0x1e810:      jne     0x1e9ca
0x1e816:      add     rsp, 0x88
0x1e81d:      pop     rbx
0x1e81e:      pop     rbp
0x1e81f:      pop     r12
0x1e821:      pop     r13
0x1e823:      pop     r14
0x1e825:      pop     r15
0x1e827:      ret

```

```

0x1e828:    mov     eax, r8d
0x1e82b:    xor     ecx, ecx
0x1e82d:    jmp     0x1e299
0x1e299:
0x1e832:    mov     eax, dword ptr [rsp + 0x10]
0x1e836:    jbe     0x1e88a
0x1e838:    sub     eax, 5
0x1e83b:    cmp     eax, 1
0x1e83e:    ja      0x1e8c8
0x1e844:    cmp     dword ptr [rsp + 0x10], 6
0x1e849:    mov     edx, 0x60
0x1e84e:    mov     eax, 0x22
0x1e853:    cmovne  eax, edx
0x1e856:    lea     rcx, [rip + 0x2a3b]
0x1e85d:    lea     rdx, [rip + 0x3ef8]
0x1e864:    cmov     rdx, rcx
0x1e868:    mov     qword ptr [rsp + 0x20], rdx
0x1e86d:    test    r10, r10
0x1e870:    je      0x1e879
0x1e872:    mov     rsi, qword ptr [rsp + 8]
0x1e877:    mov     byte ptr [rsi], al
0x1e879:    mov     r8d, 1
0x1e87f:    mov     r9d, 1
0x1e885:    jmp     0x1e0b0
0x1e0b0:
0x1e88a:    cmp     eax, 2
0x1e88d:    je      0x1e39e
0x1e893:    cmp     eax, 3
0x1e896:    jne     0x1e8c8
0x1e898:    test    r10, r10
0x1e89b:    je      0x1e8dc
0x1e89d:    mov     rax, qword ptr [rsp + 8]
0x1e8a2:    mov     r8d, 1
0x1e8a8:    mov     r9d, 1
0x1e8ae:    mov     byte ptr [rax], 0x22
0x1e8b1:    lea     rax, [rip + 0x29e0]
0x1e8b8:    mov     qword ptr [rsp + 0x20], rax
0x1e8bd:    jmp     0x1e0b0
0x1e0b0:
0x1e8c2:    nop     word ptr [rax + rax]
0x1e8c8:    mov     qword ptr [rsp + 0x20], 0
0x1e8d1:    xor     r8d, r8d
0x1e8d4:    xor     r9d, r9d
0x1e8d7:    jmp     0x1e0b0
0x1e0b0:
0x1e8dc:    lea     rax, [rip + 0x29b5]
0x1e8e3:    mov     r8d, 1
0x1e8e9:    mov     r9d, 1

```



```

0x1e8ef:      mov     qword ptr [rsp + 0x20], rax
0x1e8f4:      jmp     0x1e0b0
0x1e0b0:
0x1e8f9:      cmp     r10, rbx
0x1e8fc:      jbe     0x1e907
0x1e8fe:      mov     rax, qword ptr [rsp + 8]
0x1e903:      mov     byte ptr [rax + rbx], 0x3f
0x1e907:      lea     rax, [rbx + 1]
0x1e90b:      cmp     r10, rax
0x1e90e:      jbe     0x1e91a
0x1e910:      mov     rax, qword ptr [rsp + 8]
0x1e915:      mov     byte ptr [rax + rbx + 1], 0x5c
0x1e91a:      lea     rax, [rbx + 2]
0x1e91e:      cmp     r10, rax
0x1e921:      jbe     0x1e92d
0x1e923:      mov     rax, qword ptr [rsp + 8]
0x1e928:      mov     byte ptr [rax + rbx + 2], 0x3f
0x1e92d:      add     rbx, 3
0x1e931:      lea     r15, [r12 + 3]
0x1e936:      jmp     0x1e1a4
0x1e1a4:
0x1e93b:      xor     ecx, ecx
0x1e93d:      cmp     r13, r12
0x1e940:      ja      0x1e951
0x1e942:      jmp     0x1e9c2
0x1e9c2:
0x1e944:      nop     dword ptr [rax]
0x1e948:      add     rcx, 1
0x1e94c:      cmp     rdx, rcx
0x1e94f:      je      0x1e957
0x1e951:      cmp     byte ptr [rsi + rcx], 0
0x1e955:      jne     0x1e948
0x1e957:      mov     eax, r8d
0x1e95a:      jmp     0x1e630
0x1e630:
0x1e95f:      cmp     rax, 1
0x1e963:      je      0x1e5ec
0x1e969:      lea     rdx, [r11 + r14]
0x1e96d:      add     rsi, rax
0x1e970:      jmp     0x1e985
0x1e985:
0x1e972:      nop     word ptr [rax + rax]
0x1e978:      add     rdx, 1
0x1e97c:      cmp     rsi, rdx
0x1e97f:      je      0x1e5ec
0x1e985:      movzx   eax, byte ptr [rdx]
0x1e988:      sub     eax, 0x5b
0x1e98b:      cmp     al, 0x21

```

```

0x1e98d:    ja      0x1e978
0x1e98f:    movabs  rdi, 0x20000002b
0x1e999:    bt      rdi, rax
0x1e99d:    jae     0x1e978
0x1e99f:    mov     qword ptr [rsp + 0x20], r10
0x1e9a4:    mov     qword ptr [rsp + 0x10], r11
0x1e9a9:    call    0x49a0
0x1e9ae:    mov     r11, qword ptr [rsp + 0x10]
0x1e9b3:    mov     r10, qword ptr [rsp + 0x20]
0x1e9b8:    mov     qword ptr [rsp + 0x28], rax
0x1e9bd:    jmp     0x1e39e
0x1e39e:
0x1e9c2:    mov     eax, r8d
0x1e9c5:    jmp     0x1e299
0x1e299:
0x1e9ca:    call    0x49d0
0x1e9cf:    mov     r12, r15
0x1e9d2:    jmp     0x1e1e1
0x1e1e1:
0x1e9d7:    nop     word ptr [rax + rax]
0x1e9e0:    push    r15
0x1e9e2:    mov     r15d, edi
0x1e9e5:    push    r14
0x1e9e7:    push    r13
0x1e9e9:    mov     r13, rdx
0x1e9ec:    push    r12
0x1e9ee:    mov     r12, rsi
0x1e9f1:    push    rbp
0x1e9f2:    mov     rbp, rcx
0x1e9f5:    push    rbx
0x1e9f6:    sub     rsp, 0x28
0x1e9fa:    call    0x4860
0x1e9ff:    mov     r14, qword ptr [rip + 0xbe0a]
0x1ea06:    mov     rbx, rax
0x1ea09:    mov     eax, dword ptr [rax]
0x1ea0b:    mov     dword ptr [rsp + 0xc], eax
0x1ea0f:    test    r15d, r15d
0x1ea12:    js      0x4f2a
0x1ea18:    cmp     dword ptr [rip + 0xbde5], r15d
0x1ea1f:    mov     r10d, dword ptr [rsp + 0xc]
0x1ea24:    ja      0x1ea89
0x1ea26:    lea     edx, [r15 + 1]
0x1ea2a:    lea     rax, [rip + 0xbdef]
0x1ea31:    mov     rsi, rdx
0x1ea34:    mov     dword ptr [rsp + 0x10], edx
0x1ea38:    shl     rsi, 4
0x1ea3c:    cmp     r14, rax
0x1ea3f:    je      0x1eb40

```

```

0x1ea45:    mov     rdi, r14
0x1ea48:    mov     qword ptr [rsp + 0x18], rdx
0x1ea4d:    call    0x1d690
0x1ea52:    mov     rdx, qword ptr [rsp + 0x18]
0x1ea57:    mov     qword ptr [rip + 0xbdb2], rax
0x1ea5e:    mov     r14, rax
0x1ea61:    mov     edi, dword ptr [rip + 0xbd9d]
0x1ea67:    xor     esi, esi
0x1ea69:    sub     rdx, rdi
0x1ea6c:    shl     rdi, 4
0x1ea70:    shl     rdx, 4
0x1ea74:    add     rdi, r14
0x1ea77:    call    0x4a50
0x1ea7c:    mov     eax, dword ptr [rsp + 0x10]
0x1ea80:    mov     r10d, dword ptr [rbx]
0x1ea83:    mov     dword ptr [rip + 0xbd7b], eax
0x1ea89:    movsxd  r8, r15d
0x1ea8c:    mov     r9, rbp
0x1ea8f:    mov     rcx, r13
0x1ea92:    mov     rdx, r12
0x1ea95:    shl     r8, 4
0x1ea99:    mov     dword ptr [rsp + 0x18], r10d
0x1ea9e:    lea     r15, [r14 + r8]
0x1eaa2:    mov     r8d, dword ptr [rbp]
0x1eaa6:    mov     r14, qword ptr [r15 + 8]
0x1eaaa:    mov     rsi, qword ptr [r15]
0x1eaad:    mov     rdi, r14
0x1eab0:    mov     qword ptr [rsp + 0x10], rsi
0x1eab5:    call    0x1e040
0x1eaba:    mov     r10d, dword ptr [rsp + 0x18]
0x1eabf:    mov     rsi, qword ptr [rsp + 0x10]
0x1eac4:    mov     dword ptr [rbx], r10d
0x1eac7:    cmp     rax, rsi
0x1eaca:    jb      0x1eb1f
0x1eacc:    lea     rsi, [rax + 1]
0x1ead0:    lea     rax, [rip + 0xd089]
0x1ead7:    mov     qword ptr [r15], rsi
0x1eada:    cmp     r14, rax
0x1eadd:    je      0x1eaf1
0x1eadf:    mov     rdi, r14
0x1eae2:    mov     qword ptr [rsp + 0x10], rsi
0x1eae7:    call    0x4800
0x1eaec:    mov     rsi, qword ptr [rsp + 0x10]
0x1eaf1:    mov     rdi, rsi
0x1eaf4:    mov     qword ptr [rsp + 0x10], rsi
0x1eaf9:    call    0x1d670
0x1eafe:    mov     r8d, dword ptr [rbp]
0x1eb02:    mov     r9, rbp

```

```

0x1eb05:      mov     rcx, r13
0x1eb08:      mov     qword ptr [r15 + 8], rax
0x1eb0c:      mov     rsi, qword ptr [rsp + 0x10]
0x1eb11:      mov     rdx, r12
0x1eb14:      mov     rdi, rax
0x1eb17:      mov     r14, rax
0x1eb1a:      call    0x1e040
0x1eb1f:      mov     eax, dword ptr [rsp + 0xc]
0x1eb23:      mov     dword ptr [rbx], eax
0x1eb25:      add     rsp, 0x28
0x1eb29:      mov     rax, r14
0x1eb2c:      pop     rbx
0x1eb2d:      pop     rbp
0x1eb2e:      pop     r12
0x1eb30:      pop     r13
0x1eb32:      pop     r14
0x1eb34:      pop     r15
0x1eb36:      ret
0x1eb37:      nop     word ptr [rax + rax]
0x1eb40:      xor     edi, edi
0x1eb42:      mov     qword ptr [rsp + 0x18], rdx
0x1eb47:      call    0x1d690
0x1eb4c:      movdqa  xmm0, xmmword ptr [rip + 0xbccc]
0x1eb54:      mov     rdx, qword ptr [rsp + 0x18]
0x1eb59:      mov     qword ptr [rip + 0xbcb0], rax
0x1eb60:      mov     r14, rax
0x1eb63:      movups  xmmword ptr [rax], xmm0
0x1eb66:      jmp     0x1ea61
0x1ea61:
0x1eb6b:      nop     dword ptr [rax + rax]
0x1eb70:      endbr64
0x1eb74:      push    r12
0x1eb76:      push    rbp
0x1eb77:      mov     rbp, rdi
0x1eb7a:      push    rbx
0x1eb7b:      call    0x4860
0x1eb80:      test    rbp, rbp
0x1eb83:      mov     esi, 0x24
0x1eb88:      mov     r12d, dword ptr [rax]
0x1eb8b:      mov     rbx, rax
0x1eb8e:      lea     rax, [rip + 0xd0cb]
0x1eb95:      cmov     rbp, rax
0x1eb99:      mov     rdi, rbp
0x1eb9c:      call    0x1d760
0x1eba1:      mov     dword ptr [rbx], r12d
0x1eba4:      pop     rbx
0x1eba5:      pop     rbp
0x1eba6:      pop     r12

```

```

0x1eba8:      ret
0x1eba9:      nop        dword ptr [rax]
0x1ebb0:      endbr64
0x1ebb4:      test     rdi, rdi
0x1ebb7:      lea      rax, [rip + 0xd0a2]
0x1ebbe:      cmov     rdi, rax
0x1ebc2:      mov      eax, dword ptr [rdi]
0x1ebc4:      ret
0x1ebc5:      nop      word ptr cs:[rax + rax]
0x1ebd0:      endbr64
0x1ebd4:      test     rdi, rdi
0x1ebd7:      lea      rax, [rip + 0xd082]
0x1ebde:      cmov     rdi, rax
0x1ebe2:      mov      dword ptr [rdi], esi
0x1ebe4:      ret
0x1ebe5:      nop      word ptr cs:[rax + rax]
0x1ebf0:      endbr64
0x1ebf4:      test     rdi, rdi
0x1ebf7:      lea      rax, [rip + 0xd062]
0x1ebfe:      mov      ecx, esi
0x1ec00:      cmov     rdi, rax
0x1ec04:      mov      eax, esi
0x1ec06:      and      ecx, 0x1f
0x1ec09:      shr      al, 5
0x1ec0c:      movzx    eax, al
0x1ec0f:      lea      rsi, [rdi + rax*4 + 4]
0x1ec14:      mov      edi, dword ptr [rsi]
0x1ec16:      mov      eax, edi
0x1ec18:      shr      eax, cl
0x1ec1a:      xor      edx, eax
0x1ec1c:      and      eax, 1
0x1ec1f:      and      edx, 1
0x1ec22:      shl      edx, cl
0x1ec24:      xor      edx, edi
0x1ec26:      mov      dword ptr [rsi], edx
0x1ec28:      ret
0x1ec29:      nop      dword ptr [rax]
0x1ec30:      endbr64
0x1ec34:      push     r15
0x1ec36:      lea      rax, [rip + 0xd023]
0x1ec3d:      push     r14
0x1ec3f:      mov      r14, rdx
0x1ec42:      push     r13
0x1ec44:      mov      r13, rsi
0x1ec47:      push     r12
0x1ec49:      mov      r12, rdi
0x1ec4c:      push     rbp
0x1ec4d:      push     rbx

```

```

0x1ec4e:    mov     rbx, r8
0x1ec51:    sub     rsp, 0x18
0x1ec55:    test    r8, r8
0x1ec58:    cmov     rbx, rax
0x1ec5c:    mov     qword ptr [rsp + 8], rcx
0x1ec61:    call    0x4860
0x1ec66:    mov     rcx, qword ptr [rsp + 8]
0x1ec6b:    mov     rdx, r14
0x1ec6e:    mov     rsi, r13
0x1ec71:    mov     r15d, dword ptr [rax]
0x1ec74:    mov     r8d, dword ptr [rbx]
0x1ec77:    mov     rbp, rax
0x1ec7a:    mov     r9, rbx
0x1ec7d:    mov     rdi, r12
0x1ec80:    call    0x1e040
0x1ec85:    mov     dword ptr [rbp], r15d
0x1ec89:    add     rsp, 0x18
0x1ec8d:    pop     rbx
0x1ec8e:    pop     rbp
0x1ec8f:    pop     r12
0x1ec91:    pop     r13
0x1ec93:    pop     r14
0x1ec95:    pop     r15
0x1ec97:    ret
0x1ec98:    nop     dword ptr [rax + rax]
0x1eca0:    endbr64
0x1eca4:    push    r15
0x1eca6:    mov     r15, rdi
0x1eca9:    push    r14
0x1ecab:    mov     r14, rsi
0x1ecae:    push    r13
0x1ecb0:    push    r12
0x1ecb2:    push    rbp
0x1ecb3:    push    rbx
0x1ecb4:    mov     rbx, rdx
0x1ecb7:    sub     rsp, 0x18
0x1ecbb:    call    0x4860
0x1ecc0:    test    rbx, rbx
0x1ecc3:    mov     rcx, r14
0x1ecc6:    mov     rdx, r15
0x1ecc9:    mov     r12d, dword ptr [rax]
0x1eccc:    mov     rbp, rax
0x1eccf:    lea     rax, [rip + 0xcf8a]
0x1ecd6:    cmov     rbx, rax
0x1ecda:    xor     esi, esi
0x1ecdc:    xor     edi, edi
0x1ecde:    mov     r8d, dword ptr [rbx]
0x1ece1:    mov     r9, rbx

```

```

0x1ece4:      call    0x1e040
0x1ece9:      mov     dword ptr [rbp], r12d
0x1eced:      lea     r13, [rax + 1]
0x1ecf1:      mov     dword ptr [rsp + 0xc], r12d
0x1ecf6:      mov     rdi, r13
0x1ecf9:      call    0x1d670
0x1ecfe:      mov     r8d, dword ptr [rbx]
0x1ed01:      mov     r9, rbx
0x1ed04:      mov     rcx, r14
0x1ed07:      mov     rdx, r15
0x1ed0a:      mov     rsi, r13
0x1ed0d:      mov     rdi, rax
0x1ed10:      mov     r12, rax
0x1ed13:      call    0x1e040
0x1ed18:      mov     eax, dword ptr [rsp + 0xc]
0x1ed1c:      mov     dword ptr [rbp], eax
0x1ed1f:      add     rsp, 0x18
0x1ed23:      mov     rax, r12
0x1ed26:      pop     rbx
0x1ed27:      pop     rbp
0x1ed28:      pop     r12
0x1ed2a:      pop     r13
0x1ed2c:      pop     r14
0x1ed2e:      pop     r15
0x1ed30:      ret
0x1ed31:      nop     word ptr cs:[rax + rax]
0x1ed3c:      nop     dword ptr [rax]
0x1ed40:      endbr64
0x1ed44:      mov     eax, dword ptr [rip + 0xbaba]
0x1ed4a:      push    r12
0x1ed4c:      mov     r12, qword ptr [rip + 0xbabd]
0x1ed53:      push    rbp
0x1ed54:      push    rbx
0x1ed55:      cmp     eax, 1
0x1ed58:      jbe     0x1ed81
0x1ed5a:      sub     eax, 2
0x1ed5d:      lea     rbx, [r12 + 0x18]
0x1ed62:      shl     rax, 4
0x1ed66:      lea     rbp, [r12 + rax + 0x28]
0x1ed6b:      nop     dword ptr [rax + rax]
0x1ed70:      mov     rdi, qword ptr [rbx]
0x1ed73:      add     rbx, 0x10
0x1ed77:      call    0x4800
0x1ed7c:      cmp     rbx, rbp
0x1ed7f:      jne     0x1ed70
0x1ed81:      mov     rdi, qword ptr [r12 + 8]
0x1ed86:      lea     rbx, [rip + 0xcdd3]
0x1ed8d:      cmp     rdi, rbx

```

```

0x1ed90:    je      0x1eda9
0x1ed92:    call   0x4800
0x1ed97:    mov     qword ptr [rip + 0xba8a], rbx
0x1ed9e:    mov     qword ptr [rip + 0xba77], 0x100
0x1eda9:    lea     rbx, [rip + 0xba70]
0x1edb0:    cmp     r12, rbx
0x1edb3:    je      0x1edc4
0x1edb5:    mov     rdi, r12
0x1edb8:    call   0x4800
0x1edbd:    mov     qword ptr [rip + 0xba4c], rbx
0x1edc4:    mov     dword ptr [rip + 0xba36], 1
0x1edce:    pop     rbx
0x1edcf:    pop     rbp
0x1edd0:    pop     r12
0x1edd2:    ret
0x1edd3:    nop     word ptr cs:[rax + rax]
0x1edde:    nop
0x1ede0:    endbr64
0x1ede4:    lea     rcx, [rip + 0xce75]
0x1edeb:    mov     rdx, -1
0x1edf2:    jmp     0x1e9e0
0x1e9e0:
0x1edf7:    nop     word ptr [rax + rax]
0x1ee00:    endbr64
0x1ee04:    mov     rsi, rdi
0x1ee07:    lea     rcx, [rip + 0xce52]
0x1ee0e:    mov     rdx, -1
0x1ee15:    xor     edi, edi
0x1ee17:    jmp     0x1e9e0
0x1e9e0:
0x1ee1c:    nop     dword ptr [rax]
0x1ee20:    endbr64
0x1ee24:    sub     rsp, 0x68
0x1ee28:    mov     r8d, esi
0x1ee2b:    pxor    xmm0, xmm0
0x1ee2f:    mov     rsi, rdx
0x1ee32:    mov     rax, qword ptr fs:[0x28]
0x1ee3b:    mov     qword ptr [rsp + 0x58], rax
0x1ee40:    xor     eax, eax
0x1ee42:    movups  xmmword ptr [rsp + 0x34], xmm0
0x1ee47:    mov     rdx, -1
0x1ee4e:    mov     rcx, rsp
0x1ee51:    movups  xmmword ptr [rsp + 0x44], xmm0
0x1ee56:    movdqa  xmm2, xmmword ptr [rsp + 0x40]
0x1ee5c:    mov     eax, dword ptr [rsp + 0x50]
0x1ee60:    mov     dword ptr [rsp + 0x30], r8d
0x1ee65:    movdqa  xmm1, xmmword ptr [rsp + 0x30]
0x1ee6b:    mov     dword ptr [rsp + 0x20], eax

```



```

0x1ee6f:      movaps  xmmword ptr [rsp], xmm1
0x1ee73:      movaps  xmmword ptr [rsp + 0x10], xmm2
0x1ee78:      call   0x1e9e0
0x1ee7d:      mov     rdx, qword ptr [rsp + 0x58]
0x1ee82:      sub     rdx, qword ptr fs:[0x28]
0x1ee8b:      jne     0x1ee92
0x1ee8d:      add     rsp, 0x68
0x1ee91:      ret
0x1ee92:      call   0x49d0
0x1ee97:      nop     word ptr [rax + rax]
0x1eea0:      endbr64
0x1eea4:      sub     rsp, 0x68
0x1eea8:      mov     r8d, esi
0x1eeab:      pxor    xmm0, xmm0
0x1eeaf:      mov     rsi, rdx
0x1eeb2:      mov     rax, qword ptr fs:[0x28]
0x1eebb:      mov     qword ptr [rsp + 0x58], rax
0x1eec0:      xor     eax, eax
0x1eec2:      movups  xmmword ptr [rsp + 0x34], xmm0
0x1eec7:      mov     rdx, rcx
0x1eeca:      mov     rcx, rsp
0x1eecd:      movups  xmmword ptr [rsp + 0x44], xmm0
0x1eed2:      movdqa  xmm2, xmmword ptr [rsp + 0x40]
0x1eed8:      mov     eax, dword ptr [rsp + 0x50]
0x1eedc:      mov     dword ptr [rsp + 0x30], r8d
0x1eee1:      movdqa  xmm1, xmmword ptr [rsp + 0x30]
0x1eee7:      mov     dword ptr [rsp + 0x20], eax
0x1eeeb:      movaps  xmmword ptr [rsp], xmm1
0x1eeef:      movaps  xmmword ptr [rsp + 0x10], xmm2
0x1eef4:      call   0x1e9e0
0x1eef9:      mov     rdx, qword ptr [rsp + 0x58]
0x1eefe:      sub     rdx, qword ptr fs:[0x28]
0x1ef07:      jne     0x1ef0e
0x1ef09:      add     rsp, 0x68
0x1ef0d:      ret
0x1ef0e:      call   0x49d0
0x1ef13:      nop     word ptr cs:[rax + rax]
0x1ef1e:      nop
0x1ef20:      endbr64
0x1ef24:      sub     rsp, 0x68
0x1ef28:      pxor    xmm0, xmm0
0x1ef2c:      mov     rdx, -1
0x1ef33:      mov     rax, qword ptr fs:[0x28]
0x1ef3c:      mov     qword ptr [rsp + 0x58], rax
0x1ef41:      xor     eax, eax
0x1ef43:      mov     dword ptr [rsp + 0x30], edi
0x1ef47:      mov     rcx, rsp
0x1ef4a:      xor     edi, edi

```

```

0x1ef4c:      movups  xmmword ptr [rsp + 0x34], xmm0
0x1ef51:      movdqa  xmm1, xmmword ptr [rsp + 0x30]
0x1ef57:      movups  xmmword ptr [rsp + 0x44], xmm0
0x1ef5c:      movdqa  xmm2, xmmword ptr [rsp + 0x40]
0x1ef62:      mov     eax, dword ptr [rsp + 0x50]
0x1ef66:      movaps  xmmword ptr [rsp], xmm1
0x1ef6a:      mov     dword ptr [rsp + 0x20], eax
0x1ef6e:      movaps  xmmword ptr [rsp + 0x10], xmm2
0x1ef73:      call    0x1e9e0
0x1ef78:      mov     rdx, qword ptr [rsp + 0x58]
0x1ef7d:      sub     rdx, qword ptr fs:[0x28]
0x1ef86:      jne     0x1ef8d
0x1ef88:      add     rsp, 0x68
0x1ef8c:      ret
0x1ef8d:      call    0x49d0
0x1ef92:      nop     word ptr cs:[rax + rax]
0x1ef9d:      nop     dword ptr [rax]
0x1efa0:      endbr64
0x1efa4:      sub     rsp, 0x38
0x1efa8:      movdqa  xmm0, xmmword ptr [rip + 0xccb0]
0x1efb0:      movdqa  xmm1, xmmword ptr [rip + 0xccb8]
0x1efb8:      mov     ecx, esi
0x1efba:      mov     rax, qword ptr fs:[0x28]
0x1efc3:      mov     qword ptr [rsp + 0x28], rax
0x1efc8:      xor     eax, eax
0x1efca:      mov     eax, dword ptr [rip + 0xccb0]
0x1efd0:      and     ecx, 0x1f
0x1efd3:      mov     r9, rsp
0x1efd6:      movaps  xmmword ptr [rsp], xmm0
0x1efda:      mov     dword ptr [rsp + 0x20], eax
0x1efde:      mov     eax, esi
0x1efe0:      shr     al, 5
0x1efe3:      movaps  xmmword ptr [rsp + 0x10], xmm1
0x1efe8:      movzx   eax, al
0x1efeb:      lea     rdx, [rsp + rax*4 + 4]
0x1eff0:      mov     esi, dword ptr [rdx]
0x1eff2:      mov     eax, esi
0x1eff4:      shr     eax, cl
0x1eff6:      not     eax
0x1eff8:      and     eax, 1
0x1effb:      shl     eax, cl
0x1effd:      mov     rcx, r9
0x1f000:      xor     eax, esi
0x1f002:      mov     rsi, rdi
0x1f005:      xor     edi, edi
0x1f007:      mov     dword ptr [rdx], eax
0x1f009:      mov     rdx, -1
0x1f010:      call    0x1e9e0

```

```

0x1f015:      mov     rdx, qword ptr [rsp + 0x28]
0x1f01a:      sub     rdx, qword ptr fs:[0x28]
0x1f023:      jne     0x1f02a
0x1f025:      add     rsp, 0x38
0x1f029:      ret
0x1f02a:      call    0x49d0
0x1f02f:      nop
0x1f030:      endbr64
0x1f034:      mov     esi, 0x3a
0x1f039:      jmp     0x1efa0
0x1efa0
0x1e5ec:      mov     edi, dword ptr [rsp + 0x6c]
0x1e5f0:      mov     qword ptr [rsp + 0x50], r11
0x1e5f5:      mov     qword ptr [rsp + 0x48], r10
0x1e5fa:      mov     byte ptr [rsp + 0x40], r8b
0x1e5ff:      mov     qword ptr [rsp + 0x38], r9
0x1e604:      mov     qword ptr [rsp + 0x30], rcx
0x1e609:      call    0x4e80
0x1e60e:      movzx   r8d, byte ptr [rsp + 0x40]
0x1e614:      mov     rcx, qword ptr [rsp + 0x30]
0x1e619:      test    eax, eax
0x1e61b:      mov     r9, qword ptr [rsp + 0x38]
0x1e620:      mov     r10, qword ptr [rsp + 0x48]
0x1e625:      sete    al
0x1e628:      mov     r11, qword ptr [rsp + 0x50]
0x1e62d:      and     eax, r8d
0x1e630:      cmp     rcx, 1
0x1e634:      ja      0x1e2a4
0x1e63a:      jmp     0x1e299
0x1e299
0x1e63f:      mov     ebp, 0xc
0x1e644:      mov     eax, 0x66
0x1e649:      nop     dword ptr [rax]
0x1e650:      add     r12, 1
0x1e654:      test    r8b, r8b
0x1e657:      je      0x1e1e1
0x1e65d:      mov     ebp, eax
0x1e65f:      jmp     0x1e1cf
0x1e1cf
0x1e664:      nop     dword ptr [rax]
0x1e668:      mov     ebp, 0xb
0x1e66d:      mov     eax, 0x76
0x1e672:      jmp     0x1e650
0x1e650
0x1e674:      mov     ebp, 9
0x1e679:      mov     eax, 0x74
0x1e67e:      cmp     dword ptr [rsp + 0x10], 1
0x1e683:      jne     0x1e650

```

```

0x1e685:      jmp      0x1e380
0x1e380
0x1e68a:      nop      word ptr [rax + rax]
0x1e690:      add      r12, 1
0x1e694:      test     r8b, r8b
0x1e697:      jne      0x1e3f5
0x1e69d:      xor      ebp, ebp
0x1e69f:      jmp      0x1e1e1
0x1e1e1
0x1e6a4:      mov      ebp, 8
0x1e6a9:      mov      eax, 0x62
0x1e6ae:      jmp      0x1e650
0x1e650
0x1e6b0:      mov      ebp, 0xa
0x1e6b5:      mov      eax, 0x6e
0x1e6ba:      jmp      0x1e67e
0x1e67e
0x1e6bc:      mov      ebp, 7
0x1e6c1:      mov      eax, 0x61
0x1e6c6:      jmp      0x1e650
0x1e650
0x1e6c8:      mov      ebp, 0xd
0x1e6cd:      mov      eax, 0x72
0x1e6d2:      jmp      0x1e67e
0x1e67e
0x1e6d4:      nop      dword ptr [rax]
0x1e6d8:      cmp      dl, 0x7a
0x1e6db:      jg       0x1e4e0
0x1e6e1:      cmp      dl, 0x40
0x1e6e4:      je       0x1e500
0x1e6ea:      lea      ecx, [rbp - 0x41]
0x1e6ed:      mov      eax, 1
0x1e6f2:      movabs   rdi, 0x3fffffff53ffffff
0x1e6fc:      shl      rax, cl
0x1e6ff:      test     rax, rdi
0x1e702:      jne      0x1e215
0x1e708:      test     eax, 0xa4000000
0x1e70d:      jne      0x1e20a
0x1e713:      cmp      bpl, 0x5c
0x1e717:      jne      0x1e500
0x1e71d:      mov      eax, 0x5c
0x1e722:      jmp      0x1e67e
0x1e67e
0x1e727:      cmp      dl, 0x7b
0x1e72a:      jne      0x1e750
0x1e72c:      cmp      r13, 1
0x1e730:      setne    al
0x1e733:      cmp      r13, -1

```

```

0x1e737:      jne      0x1e741
0x1e739:      cmp      byte ptr [r11 + 1], 0
0x1e73e:      setne    al
0x1e741:      test     al, al
0x1e743:      jne      0x1e215
0x1e749:      jmp      0x1e365
0x1e365:
0x1e74e:      nop
0x1e750:      cmp      dl, 0x7c
0x1e753:      je       0x1e20a
0x1e759:      jmp      0x1e240
0x1e240:
0x1e75e:      nop
0x1e760:      lea      rax, [r12 + 2]
0x1e765:      lea      r15, [r12 + 1]
0x1e76a:      mov      ebp, 0x3f
0x1e76f:      cmp      rax, r13
0x1e772:      jae      0x1e1a4
0x1e778:      cmp      byte ptr [r11 + r12 + 1], 0x3f
0x1e77e:      jne      0x1e1a4
0x1e784:      movzx    ebp, byte ptr [r11 + r12 + 2]
0x1e78a:      cmp      bpl, 0x3e
0x1e78e:      ja       0x1e7a4
0x1e790:      movabs   rax, 0x7000a38200000000
0x1e79a:      bt       rax, rbp
0x1e79e:      jb       0x1e8f9
0x1e7a4:      mov      ebp, 0x3f
0x1e7a9:      jmp      0x1e1a4
0x1e1a4:
0x1e7ae:      test     r12, r12
0x1e7b1:      jne      0x1e7be
0x1e7b3:      cmp      dword ptr [rsp + 0x10], 1
0x1e7b8:      je       0x1e380
0x1e7be:      mov      rsi, qword ptr [rsp + 0x20]
0x1e7c3:      mov      rax, rbx
0x1e7c6:      test     rsi, rsi
0x1e7c9:      je       0x1e7f4
0x1e7cb:      movzx    edx, byte ptr [rsi]
0x1e7ce:      test     dl, dl
0x1e7d0:      je       0x1e7f4
0x1e7d2:      sub      rsi, rbx
0x1e7d5:      mov      rcx, rsi
0x1e7d8:      mov      rsi, qword ptr [rsp + 8]
0x1e7dd:      nop      dword ptr [rax]
0x1e7e0:      cmp      r10, rax
0x1e7e3:      jbe      0x1e7e8
0x1e7e5:      mov      byte ptr [rsi + rax], dl
0x1e7e8:      add      rax, 1

```

```

0x1e7ec:      movzx    edx, byte ptr [rcx + rax]
0x1e7f0:      test     dl, dl
0x1e7f2:      jne       0x1e7e0
0x1e7f4:      cmp       rax, r10
0x1e7f7:      jae       0x1e802
0x1e7f9:      mov       rsi, qword ptr [rsp + 8]
0x1e7fe:      mov       byte ptr [rsi + rax], 0
0x1e802:      mov       rdx, qword ptr [rsp + 0x78]
0x1e807:      sub       rdx, qword ptr fs:[0x28]
0x1e810:      jne       0x1e9ca
0x1e816:      add       rsp, 0x88
0x1e81d:      pop       rbx
0x1e81e:      pop       rbp
0x1e81f:      pop       r12
0x1e821:      pop       r13
0x1e823:      pop       r14
0x1e825:      pop       r15
0x1e827:      ret
0x1e828:      mov       eax, r8d
0x1e82b:      xor       ecx, ecx
0x1e82d:      jmp       0x1e299
0x1e299:
0x1e832:      mov       eax, dword ptr [rsp + 0x10]
0x1e836:      jbe       0x1e88a
0x1e838:      sub       eax, 5
0x1e83b:      cmp       eax, 1
0x1e83e:      ja        0x1e8c8
0x1e844:      cmp       dword ptr [rsp + 0x10], 6
0x1e849:      mov       edx, 0x60
0x1e84e:      mov       eax, 0x22
0x1e853:      cmovne    eax, edx
0x1e856:      lea       rcx, [rip + 0x2a3b]
0x1e85d:      lea       rdx, [rip + 0x3ef8]
0x1e864:      cmov     rdx, rcx
0x1e868:      mov       qword ptr [rsp + 0x20], rdx
0x1e86d:      test      r10, r10
0x1e870:      je        0x1e879
0x1e872:      mov       rsi, qword ptr [rsp + 8]
0x1e877:      mov       byte ptr [rsi], al
0x1e879:      mov       r8d, 1
0x1e87f:      mov       r9d, 1
0x1e885:      jmp       0x1e0b0
0x1e0b0:
0x1e88a:      cmp       eax, 2
0x1e88d:      je        0x1e39e
0x1e893:      cmp       eax, 3
0x1e896:      jne       0x1e8c8
0x1e898:      test      r10, r10

```

```

0x1e89b:      je      0x1e8dc
0x1e89d:      mov     rax, qword ptr [rsp + 8]
0x1e8a2:      mov     r8d, 1
0x1e8a8:      mov     r9d, 1
0x1e8ae:      mov     byte ptr [rax], 0x22
0x1e8b1:      lea     rax, [rip + 0x29e0]
0x1e8b8:      mov     qword ptr [rsp + 0x20], rax
0x1e8bd:      jmp     0x1e0b0
0x1e0b0
0x1e8c2:      nop     word ptr [rax + rax]
0x1e8c8:      mov     qword ptr [rsp + 0x20], 0
0x1e8d1:      xor     r8d, r8d
0x1e8d4:      xor     r9d, r9d
0x1e8d7:      jmp     0x1e0b0
0x1e0b0
0x1e8dc:      lea     rax, [rip + 0x29b5]
0x1e8e3:      mov     r8d, 1
0x1e8e9:      mov     r9d, 1
0x1e8ef:      mov     qword ptr [rsp + 0x20], rax
0x1e8f4:      jmp     0x1e0b0
0x1e0b0
0x1e8f9:      cmp     r10, rbx
0x1e8fc:      jbe     0x1e907
0x1e8fe:      mov     rax, qword ptr [rsp + 8]
0x1e903:      mov     byte ptr [rax + rbx], 0x3f
0x1e907:      lea     rax, [rbx + 1]
0x1e90b:      cmp     r10, rax
0x1e90e:      jbe     0x1e91a
0x1e910:      mov     rax, qword ptr [rsp + 8]
0x1e915:      mov     byte ptr [rax + rbx + 1], 0x5c
0x1e91a:      lea     rax, [rbx + 2]
0x1e91e:      cmp     r10, rax
0x1e921:      jbe     0x1e92d
0x1e923:      mov     rax, qword ptr [rsp + 8]
0x1e928:      mov     byte ptr [rax + rbx + 2], 0x3f
0x1e92d:      add     rbx, 3
0x1e931:      lea     r15, [r12 + 3]
0x1e936:      jmp     0x1e1a4
0x1e1a4
0x1e93b:      xor     ecx, ecx
0x1e93d:      cmp     r13, r12
0x1e940:      ja      0x1e951
0x1e942:      jmp     0x1e9c2
0x1e9c2
0x1e944:      nop     dword ptr [rax]
0x1e948:      add     rcx, 1
0x1e94c:      cmp     rdx, rcx
0x1e94f:      je      0x1e957

```

```

0x1e951:      cmp     byte ptr [rsi + rcx], 0
0x1e955:      jne     0x1e948
0x1e957:      mov     eax, r8d
0x1e95a:      jmp     0x1e630
0x1e630
0x1e95f:      cmp     rax, 1
0x1e963:      je      0x1e5ec
0x1e969:      lea     rdx, [r11 + r14]
0x1e96d:      add     rsi, rax
0x1e970:      jmp     0x1e985
0x1e985
0x1e972:      nop     word ptr [rax + rax]
0x1e978:      add     rdx, 1
0x1e97c:      cmp     rsi, rdx
0x1e97f:      je      0x1e5ec
0x1e985:      movzx   eax, byte ptr [rdx]
0x1e988:      sub     eax, 0x5b
0x1e98b:      cmp     al, 0x21
0x1e98d:      ja      0x1e978
0x1e98f:      movabs  rdi, 0x20000002b
0x1e999:      bt      rdi, rax
0x1e99d:      jae     0x1e978
0x1e99f:      mov     qword ptr [rsp + 0x20], r10
0x1e9a4:      mov     qword ptr [rsp + 0x10], r11
0x1e9a9:      call    0x49a0
0x1e9ae:      mov     r11, qword ptr [rsp + 0x10]
0x1e9b3:      mov     r10, qword ptr [rsp + 0x20]
0x1e9b8:      mov     qword ptr [rsp + 0x28], rax
0x1e9bd:      jmp     0x1e39e
0x1e39e
0x1e9c2:      mov     eax, r8d
0x1e9c5:      jmp     0x1e299
0x1e299
0x1e9ca:      call    0x49d0
0x1e9cf:      mov     r12, r15
0x1e9d2:      jmp     0x1e1e1
0x1e1e1
0x1e9d7:      nop     word ptr [rax + rax]
0x1e9e0:      push    r15
0x1e9e2:      mov     r15d, edi
0x1e9e5:      push    r14
0x1e9e7:      push    r13
0x1e9e9:      mov     r13, rdx
0x1e9ec:      push    r12
0x1e9ee:      mov     r12, rsi
0x1e9f1:      push    rbp
0x1e9f2:      mov     rbp, rcx
0x1e9f5:      push    rbx

```



```

0x1e9f6:      sub     rsp, 0x28
0x1e9fa:      call    0x4860
0x1e9ff:      mov     r14, qword ptr [rip + 0xbe0a]
0x1ea06:      mov     rbx, rax
0x1ea09:      mov     eax, dword ptr [rax]
0x1ea0b:      mov     dword ptr [rsp + 0xc], eax
0x1ea0f:      test    r15d, r15d
0x1ea12:      js       0x4f2a
0x1ea18:      cmp     dword ptr [rip + 0xbde5], r15d
0x1ea1f:      mov     r10d, dword ptr [rsp + 0xc]
0x1ea24:      ja      0x1ea89
0x1ea26:      lea     edx, [r15 + 1]
0x1ea2a:      lea     rax, [rip + 0xbdef]
0x1ea31:      mov     rsi, rdx
0x1ea34:      mov     dword ptr [rsp + 0x10], edx
0x1ea38:      shl     rsi, 4
0x1ea3c:      cmp     r14, rax
0x1ea3f:      je      0x1eb40
0x1ea45:      mov     rdi, r14
0x1ea48:      mov     qword ptr [rsp + 0x18], rdx
0x1ea4d:      call    0x1d690
0x1ea52:      mov     rdx, qword ptr [rsp + 0x18]
0x1ea57:      mov     qword ptr [rip + 0xbdb2], rax
0x1ea5e:      mov     r14, rax
0x1ea61:      mov     edi, dword ptr [rip + 0xbd9d]
0x1ea67:      xor     esi, esi
0x1ea69:      sub     rdx, rdi
0x1ea6c:      shl     rdi, 4
0x1ea70:      shl     rdx, 4
0x1ea74:      add     rdi, r14
0x1ea77:      call    0x4a50
0x1ea7c:      mov     eax, dword ptr [rsp + 0x10]
0x1ea80:      mov     r10d, dword ptr [rbx]
0x1ea83:      mov     dword ptr [rip + 0xbd7b], eax
0x1ea89:      movsxd  r8, r15d
0x1ea8c:      mov     r9, rbp
0x1ea8f:      mov     rcx, r13
0x1ea92:      mov     rdx, r12
0x1ea95:      shl     r8, 4
0x1ea99:      mov     dword ptr [rsp + 0x18], r10d
0x1ea9e:      lea     r15, [r14 + r8]
0x1eaa2:      mov     r8d, dword ptr [rbp]
0x1eaa6:      mov     r14, qword ptr [r15 + 8]
0x1eaaa:      mov     rsi, qword ptr [r15]
0x1eaad:      mov     rdi, r14
0x1eab0:      mov     qword ptr [rsp + 0x10], rsi
0x1eab5:      call    0x1e040
0x1eaba:      mov     r10d, dword ptr [rsp + 0x18]

```

```

0x1eabf:    mov     rsi, qword ptr [rsp + 0x10]
0x1eac4:    mov     dword ptr [rbx], r10d
0x1eac7:    cmp     rax, rsi
0x1eaca:    jb      0x1eb1f
0x1eacc:    lea     rsi, [rax + 1]
0x1ead0:    lea     rax, [rip + 0xd089]
0x1ead7:    mov     qword ptr [r15], rsi
0x1eada:    cmp     r14, rax
0x1eadd:    je      0x1eaf1
0x1eadf:    mov     rdi, r14
0x1eae2:    mov     qword ptr [rsp + 0x10], rsi
0x1eae7:    call    0x4800
0x1eae8:    mov     rsi, qword ptr [rsp + 0x10]
0x1eaf1:    mov     rdi, rsi
0x1eaf4:    mov     qword ptr [rsp + 0x10], rsi
0x1eaf9:    call    0x1d670
0x1eafe:    mov     r8d, dword ptr [rbp]
0x1eb02:    mov     r9, rbp
0x1eb05:    mov     rcx, r13
0x1eb08:    mov     qword ptr [r15 + 8], rax
0x1eb0c:    mov     rsi, qword ptr [rsp + 0x10]
0x1eb11:    mov     rdx, r12
0x1eb14:    mov     rdi, rax
0x1eb17:    mov     r14, rax
0x1eb1a:    call    0x1e040
0x1eb1f:    mov     eax, dword ptr [rsp + 0xc]
0x1eb23:    mov     dword ptr [rbx], eax
0x1eb25:    add     rsp, 0x28
0x1eb29:    mov     rax, r14
0x1eb2c:    pop     rbx
0x1eb2d:    pop     rbp
0x1eb2e:    pop     r12
0x1eb30:    pop     r13
0x1eb32:    pop     r14
0x1eb34:    pop     r15
0x1eb36:    ret
0x1eb37:    nop     word ptr [rax + rax]
0x1eb40:    xor     edi, edi
0x1eb42:    mov     qword ptr [rsp + 0x18], rdx
0x1eb47:    call    0x1d690
0x1eb4c:    movdqa  xmm0, xmmword ptr [rip + 0xbccc]
0x1eb54:    mov     rdx, qword ptr [rsp + 0x18]
0x1eb59:    mov     qword ptr [rip + 0xbcb0], rax
0x1eb60:    mov     r14, rax
0x1eb63:    movups  xmmword ptr [rax], xmm0
0x1eb66:    jmp     0x1ea61
0x1ea61:
0x1eb6b:    nop     dword ptr [rax + rax]

```

```

0x1eb70:      endbr64
0x1eb74:      push    r12
0x1eb76:      push    rbp
0x1eb77:      mov     rbp, rdi
0x1eb7a:      push    rbx
0x1eb7b:      call    0x4860
0x1eb80:      test    rbp, rbp
0x1eb83:      mov     esi, 0x24
0x1eb88:      mov     r12d, dword ptr [rax]
0x1eb8b:      mov     rbx, rax
0x1eb8e:      lea     rax, [rip + 0xd0cb]
0x1eb95:      cmove    rbp, rax
0x1eb99:      mov     rdi, rbp
0x1eb9c:      call    0x1d760
0x1eba1:      mov     dword ptr [rbx], r12d
0x1eba4:      pop     rbx
0x1eba5:      pop     rbp
0x1eba6:      pop     r12
0x1eba8:      ret
0x1eba9:      nop     dword ptr [rax]
0x1ebb0:      endbr64
0x1ebb4:      test    rdi, rdi
0x1ebb7:      lea     rax, [rip + 0xd0a2]
0x1ebbe:      cmove    rdi, rax
0x1ebc2:      mov     eax, dword ptr [rdi]
0x1ebc4:      ret
0x1ebc5:      nop     word ptr cs:[rax + rax]
0x1ebd0:      endbr64
0x1ebd4:      test    rdi, rdi
0x1ebd7:      lea     rax, [rip + 0xd082]
0x1ebde:      cmove    rdi, rax
0x1ebe2:      mov     dword ptr [rdi], esi
0x1ebe4:      ret
0x1ebe5:      nop     word ptr cs:[rax + rax]
0x1ebf0:      endbr64
0x1ebf4:      test    rdi, rdi
0x1ebf7:      lea     rax, [rip + 0xd062]
0x1ebfe:      mov     ecx, esi
0x1ec00:      cmove    rdi, rax
0x1ec04:      mov     eax, esi
0x1ec06:      and     ecx, 0x1f
0x1ec09:      shr     al, 5
0x1ec0c:      movzx   eax, al
0x1ec0f:      lea     rsi, [rdi + rax*4 + 4]
0x1ec14:      mov     edi, dword ptr [rsi]
0x1ec16:      mov     eax, edi
0x1ec18:      shr     eax, cl
0x1ec1a:      xor     edx, eax

```

```

0x1ec1c:    and     eax, 1
0x1ec1f:    and     edx, 1
0x1ec22:    shl     edx, cl
0x1ec24:    xor     edx, edi
0x1ec26:    mov     dword ptr [rsi], edx
0x1ec28:    ret
0x1ec29:    nop     dword ptr [rax]
0x1ec30:    endbr64
0x1ec34:    push    r15
0x1ec36:    lea     rax, [rip + 0xd023]
0x1ec3d:    push    r14
0x1ec3f:    mov     r14, rdx
0x1ec42:    push    r13
0x1ec44:    mov     r13, rsi
0x1ec47:    push    r12
0x1ec49:    mov     r12, rdi
0x1ec4c:    push    rbp
0x1ec4d:    push    rbx
0x1ec4e:    mov     rbx, r8
0x1ec51:    sub     rsp, 0x18
0x1ec55:    test    r8, r8
0x1ec58:    cmovbe  rbx, rax
0x1ec5c:    mov     qword ptr [rsp + 8], rcx
0x1ec61:    call    0x4860
0x1ec66:    mov     rcx, qword ptr [rsp + 8]
0x1ec6b:    mov     rdx, r14
0x1ec6e:    mov     rsi, r13
0x1ec71:    mov     r15d, dword ptr [rax]
0x1ec74:    mov     r8d, dword ptr [rbx]
0x1ec77:    mov     rbp, rax
0x1ec7a:    mov     r9, rbx
0x1ec7d:    mov     rdi, r12
0x1ec80:    call    0x1e040
0x1ec85:    mov     dword ptr [rbp], r15d
0x1ec89:    add     rsp, 0x18
0x1ec8d:    pop     rbx
0x1ec8e:    pop     rbp
0x1ec8f:    pop     r12
0x1ec91:    pop     r13
0x1ec93:    pop     r14
0x1ec95:    pop     r15
0x1ec97:    ret
0x1ec98:    nop     dword ptr [rax + rax]
0x1eca0:    endbr64
0x1eca4:    push    r15
0x1eca6:    mov     r15, rdi
0x1eca9:    push    r14
0x1ecab:    mov     r14, rsi

```

```

0x1ecae:    push    r13
0x1ecb0:    push    r12
0x1ecb2:    push    rbp
0x1ecb3:    push    rbx
0x1ecb4:    mov     rbx, rdx
0x1ecb7:    sub     rsp, 0x18
0x1ecbb:    call    0x4860
0x1ecc0:    test    rbx, rbx
0x1ecc3:    mov     rcx, r14
0x1ecc6:    mov     rdx, r15
0x1ecc9:    mov     r12d, dword ptr [rax]
0x1eccc:    mov     rbp, rax
0x1eccf:    lea     rax, [rip + 0xcf8a]
0x1ecd6:    cmov     rbx, rax
0x1ecda:    xor     esi, esi
0x1ecdc:    xor     edi, edi
0x1ecde:    mov     r8d, dword ptr [rbx]
0x1ece1:    mov     r9, rbx
0x1ece4:    call    0x1e040
0x1ece9:    mov     dword ptr [rbp], r12d
0x1eced:    lea     r13, [rax + 1]
0x1ecf1:    mov     dword ptr [rsp + 0xc], r12d
0x1ecf6:    mov     rdi, r13
0x1ecf9:    call    0x1d670
0x1ecfe:    mov     r8d, dword ptr [rbx]
0x1ed01:    mov     r9, rbx
0x1ed04:    mov     rcx, r14
0x1ed07:    mov     rdx, r15
0x1ed0a:    mov     rsi, r13
0x1ed0d:    mov     rdi, rax
0x1ed10:    mov     r12, rax
0x1ed13:    call    0x1e040
0x1ed18:    mov     eax, dword ptr [rsp + 0xc]
0x1ed1c:    mov     dword ptr [rbp], eax
0x1ed1f:    add     rsp, 0x18
0x1ed23:    mov     rax, r12
0x1ed26:    pop     rbx
0x1ed27:    pop     rbp
0x1ed28:    pop     r12
0x1ed2a:    pop     r13
0x1ed2c:    pop     r14
0x1ed2e:    pop     r15
0x1ed30:    ret
0x1ed31:    nop     word ptr cs:[rax + rax]
0x1ed3c:    nop     dword ptr [rax]
0x1ed40:    endbr64
0x1ed44:    mov     eax, dword ptr [rip + 0xbaba]
0x1ed4a:    push    r12

```

```

0x1ed4c:      mov     r12, qword ptr [rip + 0xbabd]
0x1ed53:      push    rbp
0x1ed54:      push    rbx
0x1ed55:      cmp     eax, 1
0x1ed58:      jbe     0x1ed81
0x1ed5a:      sub     eax, 2
0x1ed5d:      lea     rbx, [r12 + 0x18]
0x1ed62:      shl     rax, 4
0x1ed66:      lea     rbp, [r12 + rax + 0x28]
0x1ed6b:      nop     dword ptr [rax + rax]
0x1ed70:      mov     rdi, qword ptr [rbx]
0x1ed73:      add     rbx, 0x10
0x1ed77:      call    0x4800
0x1ed7c:      cmp     rbx, rbp
0x1ed7f:      jne     0x1ed70
0x1ed81:      mov     rdi, qword ptr [r12 + 8]
0x1ed86:      lea     rbx, [rip + 0xcdd3]
0x1ed8d:      cmp     rdi, rbx
0x1ed90:      je      0x1eda9
0x1ed92:      call    0x4800
0x1ed97:      mov     qword ptr [rip + 0xba8a], rbx
0x1ed9e:      mov     qword ptr [rip + 0xba77], 0x100
0x1eda9:      lea     rbx, [rip + 0xba70]
0x1edb0:      cmp     r12, rbx
0x1edb3:      je      0x1edc4
0x1edb5:      mov     rdi, r12
0x1edb8:      call    0x4800
0x1edbd:      mov     qword ptr [rip + 0xba4c], rbx
0x1edc4:      mov     dword ptr [rip + 0xba36], 1
0x1edce:      pop     rbx
0x1edcf:      pop     rbp
0x1edd0:      pop     r12
0x1edd2:      ret
0x1edd3:      nop     word ptr cs:[rax + rax]
0x1edde:      nop
0x1ede0:      endbr64
0x1ede4:      lea     rcx, [rip + 0xce75]
0x1edeb:      mov     rdx, -1
0x1edf2:      jmp     0x1e9e0
0x1e9e0:
0x1edf7:      nop     word ptr [rax + rax]
0x1ee00:      endbr64
0x1ee04:      mov     rsi, rdi
0x1ee07:      lea     rcx, [rip + 0xce52]
0x1ee0e:      mov     rdx, -1
0x1ee15:      xor     edi, edi
0x1ee17:      jmp     0x1e9e0
0x1e9e0:

```

```

0x1ee1c:      nop      dword ptr [rax]
0x1ee20:      endbr64
0x1ee24:      sub      rsp, 0x68
0x1ee28:      mov      r8d, esi
0x1ee2b:      pxor     xmm0, xmm0
0x1ee2f:      mov      rsi, rdx
0x1ee32:      mov      rax, qword ptr fs:[0x28]
0x1ee3b:      mov      qword ptr [rsp + 0x58], rax
0x1ee40:      xor      eax, eax
0x1ee42:      movups   xmmword ptr [rsp + 0x34], xmm0
0x1ee47:      mov      rdx, -1
0x1ee4e:      mov      rcx, rsp
0x1ee51:      movups   xmmword ptr [rsp + 0x44], xmm0
0x1ee56:      movdqa   xmm2, xmmword ptr [rsp + 0x40]
0x1ee5c:      mov      eax, dword ptr [rsp + 0x50]
0x1ee60:      mov      dword ptr [rsp + 0x30], r8d
0x1ee65:      movdqa   xmm1, xmmword ptr [rsp + 0x30]
0x1ee6b:      mov      dword ptr [rsp + 0x20], eax
0x1ee6f:      movaps   xmmword ptr [rsp], xmm1
0x1ee73:      movaps   xmmword ptr [rsp + 0x10], xmm2
0x1ee78:      call     0x1e9e0
0x1ee7d:      mov      rdx, qword ptr [rsp + 0x58]
0x1ee82:      sub      rdx, qword ptr fs:[0x28]
0x1ee8b:      jne      0x1ee92
0x1ee8d:      add      rsp, 0x68
0x1ee91:      ret
0x1ee92:      call     0x49d0
0x1ee97:      nop      word ptr [rax + rax]
0x1eea0:      endbr64
0x1eea4:      sub      rsp, 0x68
0x1eea8:      mov      r8d, esi
0x1eeab:      pxor     xmm0, xmm0
0x1eeaf:      mov      rsi, rdx
0x1eeb2:      mov      rax, qword ptr fs:[0x28]
0x1eebb:      mov      qword ptr [rsp + 0x58], rax
0x1eec0:      xor      eax, eax
0x1eec2:      movups   xmmword ptr [rsp + 0x34], xmm0
0x1eec7:      mov      rdx, rcx
0x1eeca:      mov      rcx, rsp
0x1eecd:      movups   xmmword ptr [rsp + 0x44], xmm0
0x1eed2:      movdqa   xmm2, xmmword ptr [rsp + 0x40]
0x1eed8:      mov      eax, dword ptr [rsp + 0x50]
0x1eedc:      mov      dword ptr [rsp + 0x30], r8d
0x1eee1:      movdqa   xmm1, xmmword ptr [rsp + 0x30]
0x1eee7:      mov      dword ptr [rsp + 0x20], eax
0x1eeeb:      movaps   xmmword ptr [rsp], xmm1
0x1eeef:      movaps   xmmword ptr [rsp + 0x10], xmm2
0x1eef4:      call     0x1e9e0

```

```

0x1eef9:      mov     rdx, qword ptr [rsp + 0x58]
0x1eefe:      sub     rdx, qword ptr fs:[0x28]
0x1ef07:      jne     0x1ef0e
0x1ef09:      add     rsp, 0x68
0x1ef0d:      ret
0x1ef0e:      call    0x49d0
0x1ef13:      nop     word ptr cs:[rax + rax]
0x1ef1e:      nop
0x1ef20:      endbr64
0x1ef24:      sub     rsp, 0x68
0x1ef28:      pxor    xmm0, xmm0
0x1ef2c:      mov     rdx, -1
0x1ef33:      mov     rax, qword ptr fs:[0x28]
0x1ef3c:      mov     qword ptr [rsp + 0x58], rax
0x1ef41:      xor     eax, eax
0x1ef43:      mov     dword ptr [rsp + 0x30], edi
0x1ef47:      mov     rcx, rsp
0x1ef4a:      xor     edi, edi
0x1ef4c:      movups  xmmword ptr [rsp + 0x34], xmm0
0x1ef51:      movdqa  xmm1, xmmword ptr [rsp + 0x30]
0x1ef57:      movups  xmmword ptr [rsp + 0x44], xmm0
0x1ef5c:      movdqa  xmm2, xmmword ptr [rsp + 0x40]
0x1ef62:      mov     eax, dword ptr [rsp + 0x50]
0x1ef66:      movaps  xmmword ptr [rsp], xmm1
0x1ef6a:      mov     dword ptr [rsp + 0x20], eax
0x1ef6e:      movaps  xmmword ptr [rsp + 0x10], xmm2
0x1ef73:      call    0x1e9e0
0x1ef78:      mov     rdx, qword ptr [rsp + 0x58]
0x1ef7d:      sub     rdx, qword ptr fs:[0x28]
0x1ef86:      jne     0x1ef8d
0x1ef88:      add     rsp, 0x68
0x1ef8c:      ret
0x1ef8d:      call    0x49d0
0x1ef92:      nop     word ptr cs:[rax + rax]
0x1ef9d:      nop     dword ptr [rax]
0x1efa0:      endbr64
0x1efa4:      sub     rsp, 0x38
0x1efa8:      movdqa  xmm0, xmmword ptr [rip + 0xccb0]
0x1efb0:      movdqa  xmm1, xmmword ptr [rip + 0xccb8]
0x1efb8:      mov     ecx, esi
0x1efba:      mov     rax, qword ptr fs:[0x28]
0x1efc3:      mov     qword ptr [rsp + 0x28], rax
0x1efc8:      xor     eax, eax
0x1efca:      mov     eax, dword ptr [rip + 0xccb0]
0x1efd0:      and     ecx, 0x1f
0x1efd3:      mov     r9, rsp
0x1efd6:      movaps  xmmword ptr [rsp], xmm0
0x1efda:      mov     dword ptr [rsp + 0x20], eax

```



```

0x1efde:    mov     eax, esi
0x1efe0:    shr     al, 5
0x1efe3:    movaps  xmmword ptr [rsp + 0x10], xmm1
0x1efe8:    movzx   eax, al
0x1efeb:    lea     rdx, [rsp + rax*4 + 4]
0x1eff0:    mov     esi, dword ptr [rdx]
0x1eff2:    mov     eax, esi
0x1eff4:    shr     eax, cl
0x1eff6:    not     eax
0x1eff8:    and     eax, 1
0x1effb:    shl     eax, cl
0x1effd:    mov     rcx, r9
0x1f000:    xor     eax, esi
0x1f002:    mov     rsi, rdi
0x1f005:    xor     edi, edi
0x1f007:    mov     dword ptr [rdx], eax
0x1f009:    mov     rdx, -1
0x1f010:    call    0x1e9e0
0x1f015:    mov     rdx, qword ptr [rsp + 0x28]
0x1f01a:    sub     rdx, qword ptr fs:[0x28]
0x1f023:    jne     0x1f02a
0x1f025:    add     rsp, 0x38
0x1f029:    ret
0x1f02a:    call    0x49d0
0x1f02f:    nop
0x1f030:    endbr64
0x1f034:    mov     esi, 0x3a
0x1f039:    jmp     0x1efa0
0x1efa0:
0x1e500:    lea     r14, [r12 + 1]
0x1e505:    cmp     qword ptr [rsp + 0x28], 1
0x1e50b:    mov     byte ptr [rsp + 0x30], dl
0x1e50f:    mov     r15, r14
0x1e512:    je      0x1e250
0x1e518:    mov     qword ptr [rsp + 0x70], 0
0x1e521:    lea     rcx, [rsp + 0x70]
0x1e526:    cmp     r13, -1
0x1e52a:    jne     0x1e574
0x1e52c:    mov     rdi, r11
0x1e52f:    mov     qword ptr [rsp + 0x58], r10
0x1e534:    mov     byte ptr [rsp + 0x50], r8b
0x1e539:    mov     qword ptr [rsp + 0x48], r9
0x1e53e:    mov     qword ptr [rsp + 0x40], rsi
0x1e543:    mov     qword ptr [rsp + 0x38], rcx
0x1e548:    mov     qword ptr [rsp + 0x30], r11
0x1e54d:    call    0x49b0
0x1e552:    mov     r10, qword ptr [rsp + 0x58]
0x1e557:    movzx   r8d, byte ptr [rsp + 0x50]

```

```

0x1e55d:    mov     r9, qword ptr [rsp + 0x48]
0x1e562:    mov     rsi, qword ptr [rsp + 0x40]
0x1e567:    mov     r13, rax
0x1e56a:    mov     rcx, qword ptr [rsp + 0x38]
0x1e56f:    mov     r11, qword ptr [rsp + 0x30]
0x1e574:    mov     rdx, r13
0x1e577:    lea     rdi, [rsp + 0x6c]
0x1e57c:    mov     qword ptr [rsp + 0x48], r11
0x1e581:    sub     rdx, r12
0x1e584:    mov     qword ptr [rsp + 0x40], r10
0x1e589:    mov     byte ptr [rsp + 0x38], r8b
0x1e58e:    mov     qword ptr [rsp + 0x30], r9
0x1e593:    mov     qword ptr [rsp + 0x58], rdx
0x1e598:    mov     qword ptr [rsp + 0x50], rsi
0x1e59d:    call    0x49f0
0x1e5a2:    mov     r9, qword ptr [rsp + 0x30]
0x1e5a7:    movzx   r8d, byte ptr [rsp + 0x38]
0x1e5ad:    test    rax, rax
0x1e5b0:    mov     r10, qword ptr [rsp + 0x40]
0x1e5b5:    mov     r11, qword ptr [rsp + 0x48]
0x1e5ba:    mov     rcx, rax
0x1e5bd:    je      0x1e1a4
0x1e5c3:    cmp     rax, -1
0x1e5c7:    je      0x1e828
0x1e5cd:    cmp     rax, -2
0x1e5d1:    mov     rsi, qword ptr [rsp + 0x50]
0x1e5d6:    mov     rdx, qword ptr [rsp + 0x58]
0x1e5db:    je      0x1e93b
0x1e5e1:    cmp     dword ptr [rsp + 0x10], 1
0x1e5e6:    je      0x1e95f
0x1e5ec:    mov     edi, dword ptr [rsp + 0x6c]
0x1e5f0:    mov     qword ptr [rsp + 0x50], r11
0x1e5f5:    mov     qword ptr [rsp + 0x48], r10
0x1e5fa:    mov     byte ptr [rsp + 0x40], r8b
0x1e5ff:    mov     qword ptr [rsp + 0x38], r9
0x1e604:    mov     qword ptr [rsp + 0x30], rcx
0x1e609:    call    0x4e80
0x1e60e:    movzx   r8d, byte ptr [rsp + 0x40]
0x1e614:    mov     rcx, qword ptr [rsp + 0x30]
0x1e619:    test    eax, eax
0x1e61b:    mov     r9, qword ptr [rsp + 0x38]
0x1e620:    mov     r10, qword ptr [rsp + 0x48]
0x1e625:    sete    al
0x1e628:    mov     r11, qword ptr [rsp + 0x50]
0x1e62d:    and     eax, r8d
0x1e630:    cmp     rcx, 1
0x1e634:    ja      0x1e2a4
0x1e63a:    jmp     0x1e299

```

```

0x1e299
0x1e63f:      mov     ebp, 0xc
0x1e644:      mov     eax, 0x66
0x1e649:      nop     dword ptr [rax]
0x1e650:      add     r12, 1
0x1e654:      test    r8b, r8b
0x1e657:      je      0x1e1e1
0x1e65d:      mov     ebp, eax
0x1e65f:      jmp     0x1e1cf
0x1e1cf
0x1e664:      nop     dword ptr [rax]
0x1e668:      mov     ebp, 0xb
0x1e66d:      mov     eax, 0x76
0x1e672:      jmp     0x1e650
0x1e650
0x1e674:      mov     ebp, 9
0x1e679:      mov     eax, 0x74
0x1e67e:      cmp     dword ptr [rsp + 0x10], 1
0x1e683:      jne     0x1e650
0x1e685:      jmp     0x1e380
0x1e380
0x1e68a:      nop     word ptr [rax + rax]
0x1e690:      add     r12, 1
0x1e694:      test    r8b, r8b
0x1e697:      jne     0x1e3f5
0x1e69d:      xor     ebp, ebp
0x1e69f:      jmp     0x1e1e1
0x1e1e1
0x1e6a4:      mov     ebp, 8
0x1e6a9:      mov     eax, 0x62
0x1e6ae:      jmp     0x1e650
0x1e650
0x1e6b0:      mov     ebp, 0xa
0x1e6b5:      mov     eax, 0x6e
0x1e6ba:      jmp     0x1e67e
0x1e67e
0x1e6bc:      mov     ebp, 7
0x1e6c1:      mov     eax, 0x61
0x1e6c6:      jmp     0x1e650
0x1e650
0x1e6c8:      mov     ebp, 0xd
0x1e6cd:      mov     eax, 0x72
0x1e6d2:      jmp     0x1e67e
0x1e67e
0x1e6d4:      nop     dword ptr [rax]
0x1e6d8:      cmp     dl, 0x7a
0x1e6db:      jg      0x1e4e0
0x1e6e1:      cmp     dl, 0x40

```

```

0x1e6e4:      je      0x1e500
0x1e6ea:      lea     ecx, [rbp - 0x41]
0x1e6ed:      mov     eax, 1
0x1e6f2:      movabs   rdi, 0x3fffffff53ffffff
0x1e6fc:      shl     rax, cl
0x1e6ff:      test    rax, rdi
0x1e702:      jne     0x1e215
0x1e708:      test    eax, 0xa4000000
0x1e70d:      jne     0x1e20a
0x1e713:      cmp     bpl, 0x5c
0x1e717:      jne     0x1e500
0x1e71d:      mov     eax, 0x5c
0x1e722:      jmp     0x1e67e
0x1e67e
0x1e727:      cmp     dl, 0x7b
0x1e72a:      jne     0x1e750
0x1e72c:      cmp     r13, 1
0x1e730:      setne   al
0x1e733:      cmp     r13, -1
0x1e737:      jne     0x1e741
0x1e739:      cmp     byte ptr [r11 + 1], 0
0x1e73e:      setne   al
0x1e741:      test    al, al
0x1e743:      jne     0x1e215
0x1e749:      jmp     0x1e365
0x1e365
0x1e74e:      nop
0x1e750:      cmp     dl, 0x7c
0x1e753:      je      0x1e20a
0x1e759:      jmp     0x1e240
0x1e240
0x1e75e:      nop
0x1e760:      lea     rax, [r12 + 2]
0x1e765:      lea     r15, [r12 + 1]
0x1e76a:      mov     ebp, 0x3f
0x1e76f:      cmp     rax, r13
0x1e772:      jae     0x1e1a4
0x1e778:      cmp     byte ptr [r11 + r12 + 1], 0x3f
0x1e77e:      jne     0x1e1a4
0x1e784:      movzx   ebp, byte ptr [r11 + r12 + 2]
0x1e78a:      cmp     bpl, 0x3e
0x1e78e:      ja      0x1e7a4
0x1e790:      movabs   rax, 0x7000a38200000000
0x1e79a:      bt      rax, rbp
0x1e79e:      jb      0x1e8f9
0x1e7a4:      mov     ebp, 0x3f
0x1e7a9:      jmp     0x1e1a4
0x1e1a4

```

```

0x1e7ae:    test    r12, r12
0x1e7b1:    jne     0x1e7be
0x1e7b3:    cmp     dword ptr [rsp + 0x10], 1
0x1e7b8:    je      0x1e380
0x1e7be:    mov     rsi, qword ptr [rsp + 0x20]
0x1e7c3:    mov     rax, rbx
0x1e7c6:    test    rsi, rsi
0x1e7c9:    je      0x1e7f4
0x1e7cb:    movzx   edx, byte ptr [rsi]
0x1e7ce:    test    dl, dl
0x1e7d0:    je      0x1e7f4
0x1e7d2:    sub     rsi, rbx
0x1e7d5:    mov     rcx, rsi
0x1e7d8:    mov     rsi, qword ptr [rsp + 8]
0x1e7dd:    nop     dword ptr [rax]
0x1e7e0:    cmp     r10, rax
0x1e7e3:    jbe     0x1e7e8
0x1e7e5:    mov     byte ptr [rsi + rax], dl
0x1e7e8:    add     rax, 1
0x1e7ec:    movzx   edx, byte ptr [rcx + rax]
0x1e7f0:    test    dl, dl
0x1e7f2:    jne     0x1e7e0
0x1e7f4:    cmp     rax, r10
0x1e7f7:    jae     0x1e802
0x1e7f9:    mov     rsi, qword ptr [rsp + 8]
0x1e7fe:    mov     byte ptr [rsi + rax], 0
0x1e802:    mov     rdx, qword ptr [rsp + 0x78]
0x1e807:    sub     rdx, qword ptr fs:[0x28]
0x1e810:    jne     0x1e9ca
0x1e816:    add     rsp, 0x88
0x1e81d:    pop     rbx
0x1e81e:    pop     rbp
0x1e81f:    pop     r12
0x1e821:    pop     r13
0x1e823:    pop     r14
0x1e825:    pop     r15
0x1e827:    ret
0x1e828:    mov     eax, r8d
0x1e82b:    xor     ecx, ecx
0x1e82d:    jmp     0x1e299
0x1e299:
0x1e832:    mov     eax, dword ptr [rsp + 0x10]
0x1e836:    jbe     0x1e88a
0x1e838:    sub     eax, 5
0x1e83b:    cmp     eax, 1
0x1e83e:    ja      0x1e8c8
0x1e844:    cmp     dword ptr [rsp + 0x10], 6
0x1e849:    mov     edx, 0x60

```

```

0x1e84e:      mov     eax, 0x22
0x1e853:      cmovne  eax, edx
0x1e856:      lea     rcx, [rip + 0x2a3b]
0x1e85d:      lea     rdx, [rip + 0x3ef8]
0x1e864:      cmov     rdx, rcx
0x1e868:      mov     qword ptr [rsp + 0x20], rdx
0x1e86d:      test    r10, r10
0x1e870:      je      0x1e879
0x1e872:      mov     rsi, qword ptr [rsp + 8]
0x1e877:      mov     byte ptr [rsi], al
0x1e879:      mov     r8d, 1
0x1e87f:      mov     r9d, 1
0x1e885:      jmp     0x1e0b0
0x1e0b0:
0x1e88a:      cmp     eax, 2
0x1e88d:      je      0x1e39e
0x1e893:      cmp     eax, 3
0x1e896:      jne     0x1e8c8
0x1e898:      test    r10, r10
0x1e89b:      je      0x1e8dc
0x1e89d:      mov     rax, qword ptr [rsp + 8]
0x1e8a2:      mov     r8d, 1
0x1e8a8:      mov     r9d, 1
0x1e8ae:      mov     byte ptr [rax], 0x22
0x1e8b1:      lea     rax, [rip + 0x29e0]
0x1e8b8:      mov     qword ptr [rsp + 0x20], rax
0x1e8bd:      jmp     0x1e0b0
0x1e0b0:
0x1e8c2:      nop     word ptr [rax + rax]
0x1e8c8:      mov     qword ptr [rsp + 0x20], 0
0x1e8d1:      xor     r8d, r8d
0x1e8d4:      xor     r9d, r9d
0x1e8d7:      jmp     0x1e0b0
0x1e0b0:
0x1e8dc:      lea     rax, [rip + 0x29b5]
0x1e8e3:      mov     r8d, 1
0x1e8e9:      mov     r9d, 1
0x1e8ef:      mov     qword ptr [rsp + 0x20], rax
0x1e8f4:      jmp     0x1e0b0
0x1e0b0:
0x1e8f9:      cmp     r10, rbx
0x1e8fc:      jbe     0x1e907
0x1e8fe:      mov     rax, qword ptr [rsp + 8]
0x1e903:      mov     byte ptr [rax + rbx], 0x3f
0x1e907:      lea     rax, [rbx + 1]
0x1e90b:      cmp     r10, rax
0x1e90e:      jbe     0x1e91a
0x1e910:      mov     rax, qword ptr [rsp + 8]

```

```

0x1e915:      mov     byte ptr [rax + rbx + 1], 0x5c
0x1e91a:      lea     rax, [rbx + 2]
0x1e91e:      cmp     r10, rax
0x1e921:      jbe     0x1e92d
0x1e923:      mov     rax, qword ptr [rsp + 8]
0x1e928:      mov     byte ptr [rax + rbx + 2], 0x3f
0x1e92d:      add     rbx, 3
0x1e931:      lea     r15, [r12 + 3]
0x1e936:      jmp     0x1e1a4
0x1e1a4
0x1e93b:      xor     ecx, ecx
0x1e93d:      cmp     r13, r12
0x1e940:      ja      0x1e951
0x1e942:      jmp     0x1e9c2
0x1e9c2
0x1e944:      nop     dword ptr [rax]
0x1e948:      add     rcx, 1
0x1e94c:      cmp     rdx, rcx
0x1e94f:      je      0x1e957
0x1e951:      cmp     byte ptr [rsi + rcx], 0
0x1e955:      jne     0x1e948
0x1e957:      mov     eax, r8d
0x1e95a:      jmp     0x1e630
0x1e630
0x1e95f:      cmp     rax, 1
0x1e963:      je      0x1e5ec
0x1e969:      lea     rdx, [r11 + r14]
0x1e96d:      add     rsi, rax
0x1e970:      jmp     0x1e985
0x1e985
0x1e972:      nop     word ptr [rax + rax]
0x1e978:      add     rdx, 1
0x1e97c:      cmp     rsi, rdx
0x1e97f:      je      0x1e5ec
0x1e985:      movzx   eax, byte ptr [rdx]
0x1e988:      sub     eax, 0x5b
0x1e98b:      cmp     al, 0x21
0x1e98d:      ja      0x1e978
0x1e98f:      movabs  rdi, 0x20000002b
0x1e999:      bt      rdi, rax
0x1e99d:      jae     0x1e978
0x1e99f:      mov     qword ptr [rsp + 0x20], r10
0x1e9a4:      mov     qword ptr [rsp + 0x10], r11
0x1e9a9:      call    0x49a0
0x1e9ae:      mov     r11, qword ptr [rsp + 0x10]
0x1e9b3:      mov     r10, qword ptr [rsp + 0x20]
0x1e9b8:      mov     qword ptr [rsp + 0x28], rax
0x1e9bd:      jmp     0x1e39e

```

```

0x1e39e
0x1e9c2:      mov     eax, r8d
0x1e9c5:      jmp     0x1e299
0x1e299
0x1e9ca:      call    0x49d0
0x1e9cf:      mov     r12, r15
0x1e9d2:      jmp     0x1e1e1
0x1e1e1
0x1e9d7:      nop     word ptr [rax + rax]
0x1e9e0:      push    r15
0x1e9e2:      mov     r15d, edi
0x1e9e5:      push    r14
0x1e9e7:      push    r13
0x1e9e9:      mov     r13, rdx
0x1e9ec:      push    r12
0x1e9ee:      mov     r12, rsi
0x1e9f1:      push    rbp
0x1e9f2:      mov     rbp, rcx
0x1e9f5:      push    rbx
0x1e9f6:      sub     rsp, 0x28
0x1e9fa:      call    0x4860
0x1e9ff:      mov     r14, qword ptr [rip + 0xbe0a]
0x1ea06:      mov     rbx, rax
0x1ea09:      mov     eax, dword ptr [rax]
0x1ea0b:      mov     dword ptr [rsp + 0xc], eax
0x1ea0f:      test    r15d, r15d
0x1ea12:      js      0x4f2a
0x1ea18:      cmp     dword ptr [rip + 0xbde5], r15d
0x1ea1f:      mov     r10d, dword ptr [rsp + 0xc]
0x1ea24:      ja      0x1ea89
0x1ea26:      lea     edx, [r15 + 1]
0x1ea2a:      lea     rax, [rip + 0xbdef]
0x1ea31:      mov     rsi, rdx
0x1ea34:      mov     dword ptr [rsp + 0x10], edx
0x1ea38:      shl     rsi, 4
0x1ea3c:      cmp     r14, rax
0x1ea3f:      je      0x1eb40
0x1ea45:      mov     rdi, r14
0x1ea48:      mov     qword ptr [rsp + 0x18], rdx
0x1ea4d:      call    0x1d690
0x1ea52:      mov     rdx, qword ptr [rsp + 0x18]
0x1ea57:      mov     qword ptr [rip + 0xbdb2], rax
0x1ea5e:      mov     r14, rax
0x1ea61:      mov     edi, dword ptr [rip + 0xbd9d]
0x1ea67:      xor     esi, esi
0x1ea69:      sub     rdx, rdi
0x1ea6c:      shl     rdi, 4
0x1ea70:      shl     rdx, 4

```



```

0x1ea74:      add     rdi, r14
0x1ea77:      call    0x4a50
0x1ea7c:      mov     eax, dword ptr [rsp + 0x10]
0x1ea80:      mov     r10d, dword ptr [rbx]
0x1ea83:      mov     dword ptr [rip + 0xbd7b], eax
0x1ea89:      movsxd   r8, r15d
0x1ea8c:      mov     r9, rbp
0x1ea8f:      mov     rcx, r13
0x1ea92:      mov     rdx, r12
0x1ea95:      shl     r8, 4
0x1ea99:      mov     dword ptr [rsp + 0x18], r10d
0x1ea9e:      lea     r15, [r14 + r8]
0x1eaa2:      mov     r8d, dword ptr [rbp]
0x1eaa6:      mov     r14, qword ptr [r15 + 8]
0x1eaaa:      mov     rsi, qword ptr [r15]
0x1eaad:      mov     rdi, r14
0x1eab0:      mov     qword ptr [rsp + 0x10], rsi
0x1eab5:      call    0x1e040
0x1eaba:      mov     r10d, dword ptr [rsp + 0x18]
0x1eabf:      mov     rsi, qword ptr [rsp + 0x10]
0x1eac4:      mov     dword ptr [rbx], r10d
0x1eac7:      cmp     rax, rsi
0x1eaca:      jb      0x1eb1f
0x1eacc:      lea     rsi, [rax + 1]
0x1ead0:      lea     rax, [rip + 0xd089]
0x1ead7:      mov     qword ptr [r15], rsi
0x1eada:      cmp     r14, rax
0x1eadd:      je      0x1eaf1
0x1eadf:      mov     rdi, r14
0x1eae2:      mov     qword ptr [rsp + 0x10], rsi
0x1eae7:      call    0x4800
0x1eaec:      mov     rsi, qword ptr [rsp + 0x10]
0x1eaf1:      mov     rdi, rsi
0x1eaf4:      mov     qword ptr [rsp + 0x10], rsi
0x1eaf9:      call    0x1d670
0x1eafe:      mov     r8d, dword ptr [rbp]
0x1eb02:      mov     r9, rbp
0x1eb05:      mov     rcx, r13
0x1eb08:      mov     qword ptr [r15 + 8], rax
0x1eb0c:      mov     rsi, qword ptr [rsp + 0x10]
0x1eb11:      mov     rdx, r12
0x1eb14:      mov     rdi, rax
0x1eb17:      mov     r14, rax
0x1eb1a:      call    0x1e040
0x1eb1f:      mov     eax, dword ptr [rsp + 0xc]
0x1eb23:      mov     dword ptr [rbx], eax
0x1eb25:      add     rsp, 0x28
0x1eb29:      mov     rax, r14

```

```

0x1eb2c:    pop     rbx
0x1eb2d:    pop     rbp
0x1eb2e:    pop     r12
0x1eb30:    pop     r13
0x1eb32:    pop     r14
0x1eb34:    pop     r15
0x1eb36:    ret
0x1eb37:    nop     word ptr [rax + rax]
0x1eb40:    xor     edi, edi
0x1eb42:    mov     qword ptr [rsp + 0x18], rdx
0x1eb47:    call    0x1d690
0x1eb4c:    movdqa  xmm0, xmmword ptr [rip + 0xbccc]
0x1eb54:    mov     rdx, qword ptr [rsp + 0x18]
0x1eb59:    mov     qword ptr [rip + 0xbcb0], rax
0x1eb60:    mov     r14, rax
0x1eb63:    movups  xmmword ptr [rax], xmm0
0x1eb66:    jmp     0x1ea61
0x1ea61:
0x1eb6b:    nop     dword ptr [rax + rax]
0x1eb70:    endbr64
0x1eb74:    push    r12
0x1eb76:    push    rbp
0x1eb77:    mov     rbp, rdi
0x1eb7a:    push    rbx
0x1eb7b:    call    0x4860
0x1eb80:    test    rbp, rbp
0x1eb83:    mov     esi, 0x24
0x1eb88:    mov     r12d, dword ptr [rax]
0x1eb8b:    mov     rbx, rax
0x1eb8e:    lea     rax, [rip + 0xd0cb]
0x1eb95:    cmove   rbp, rax
0x1eb99:    mov     rdi, rbp
0x1eb9c:    call    0x1d760
0x1eba1:    mov     dword ptr [rbx], r12d
0x1eba4:    pop     rbx
0x1eba5:    pop     rbp
0x1eba6:    pop     r12
0x1eba8:    ret
0x1eba9:    nop     dword ptr [rax]
0x1ebb0:    endbr64
0x1ebb4:    test    rdi, rdi
0x1ebb7:    lea     rax, [rip + 0xd0a2]
0x1ebbe:    cmove   rdi, rax
0x1ebc2:    mov     eax, dword ptr [rdi]
0x1ebc4:    ret
0x1ebc5:    nop     word ptr cs:[rax + rax]
0x1ebd0:    endbr64
0x1ebd4:    test    rdi, rdi

```

```

0x1ebd7:    lea     rax, [rip + 0xd082]
0x1ebde:    cmov    rdi, rax
0x1ebe2:    mov     dword ptr [rdi], esi
0x1ebe4:    ret
0x1ebe5:    nop     word ptr cs:[rax + rax]
0x1ebf0:    endbr64
0x1ebf4:    test    rdi, rdi
0x1ebf7:    lea     rax, [rip + 0xd062]
0x1ebfe:    mov     ecx, esi
0x1ec00:    cmov    rdi, rax
0x1ec04:    mov     eax, esi
0x1ec06:    and     ecx, 0x1f
0x1ec09:    shr     al, 5
0x1ec0c:    movzx   eax, al
0x1ec0f:    lea     rsi, [rdi + rax*4 + 4]
0x1ec14:    mov     edi, dword ptr [rsi]
0x1ec16:    mov     eax, edi
0x1ec18:    shr     eax, cl
0x1ec1a:    xor     edx, eax
0x1ec1c:    and     eax, 1
0x1ec1f:    and     edx, 1
0x1ec22:    shl     edx, cl
0x1ec24:    xor     edx, edi
0x1ec26:    mov     dword ptr [rsi], edx
0x1ec28:    ret
0x1ec29:    nop     dword ptr [rax]
0x1ec30:    endbr64
0x1ec34:    push    r15
0x1ec36:    lea     rax, [rip + 0xd023]
0x1ec3d:    push    r14
0x1ec3f:    mov     r14, rdx
0x1ec42:    push    r13
0x1ec44:    mov     r13, rsi
0x1ec47:    push    r12
0x1ec49:    mov     r12, rdi
0x1ec4c:    push    rbp
0x1ec4d:    push    rbx
0x1ec4e:    mov     rbx, r8
0x1ec51:    sub     rsp, 0x18
0x1ec55:    test    r8, r8
0x1ec58:    cmov    rbx, rax
0x1ec5c:    mov     qword ptr [rsp + 8], rcx
0x1ec61:    call    0x4860
0x1ec66:    mov     rcx, qword ptr [rsp + 8]
0x1ec6b:    mov     rdx, r14
0x1ec6e:    mov     rsi, r13
0x1ec71:    mov     r15d, dword ptr [rax]
0x1ec74:    mov     r8d, dword ptr [rbx]

```

```

0x1ec77:    mov     rbp, rax
0x1ec7a:    mov     r9, rbx
0x1ec7d:    mov     rdi, r12
0x1ec80:    call    0x1e040
0x1ec85:    mov     dword ptr [rbp], r15d
0x1ec89:    add     rsp, 0x18
0x1ec8d:    pop     rbx
0x1ec8e:    pop     rbp
0x1ec8f:    pop     r12
0x1ec91:    pop     r13
0x1ec93:    pop     r14
0x1ec95:    pop     r15
0x1ec97:    ret
0x1ec98:    nop     dword ptr [rax + rax]
0x1eca0:    endbr64
0x1eca4:    push    r15
0x1eca6:    mov     r15, rdi
0x1eca9:    push    r14
0x1ecab:    mov     r14, rsi
0x1ecae:    push    r13
0x1ecb0:    push    r12
0x1ecb2:    push    rbp
0x1ecb3:    push    rbx
0x1ecb4:    mov     rbx, rdx
0x1ecb7:    sub     rsp, 0x18
0x1ecbb:    call    0x4860
0x1ecc0:    test    rbx, rbx
0x1ecc3:    mov     rcx, r14
0x1ecc6:    mov     rdx, r15
0x1ecc9:    mov     r12d, dword ptr [rax]
0x1eccc:    mov     rbp, rax
0x1eccf:    lea     rax, [rip + 0xcf8a]
0x1ecd6:    cmov     rbx, rax
0x1ecda:    xor     esi, esi
0x1ecdc:    xor     edi, edi
0x1ecde:    mov     r8d, dword ptr [rbx]
0x1ece1:    mov     r9, rbx
0x1ece4:    call    0x1e040
0x1ece9:    mov     dword ptr [rbp], r12d
0x1eced:    lea     r13, [rax + 1]
0x1ecf1:    mov     dword ptr [rsp + 0xc], r12d
0x1ecf6:    mov     rdi, r13
0x1ecf9:    call    0x1d670
0x1ecfe:    mov     r8d, dword ptr [rbx]
0x1ed01:    mov     r9, rbx
0x1ed04:    mov     rcx, r14
0x1ed07:    mov     rdx, r15
0x1ed0a:    mov     rsi, r13

```

```

0x1ed0d:    mov     rdi, rax
0x1ed10:    mov     r12, rax
0x1ed13:    call    0x1e040
0x1ed18:    mov     eax, dword ptr [rsp + 0xc]
0x1ed1c:    mov     dword ptr [rbp], eax
0x1ed1f:    add     rsp, 0x18
0x1ed23:    mov     rax, r12
0x1ed26:    pop     rbx
0x1ed27:    pop     rbp
0x1ed28:    pop     r12
0x1ed2a:    pop     r13
0x1ed2c:    pop     r14
0x1ed2e:    pop     r15
0x1ed30:    ret
0x1ed31:    nop     word ptr cs:[rax + rax]
0x1ed3c:    nop     dword ptr [rax]
0x1ed40:    endbr64
0x1ed44:    mov     eax, dword ptr [rip + 0xbaba]
0x1ed4a:    push    r12
0x1ed4c:    mov     r12, qword ptr [rip + 0xbabd]
0x1ed53:    push    rbp
0x1ed54:    push    rbx
0x1ed55:    cmp     eax, 1
0x1ed58:    jbe     0x1ed81
0x1ed5a:    sub     eax, 2
0x1ed5d:    lea     rbx, [r12 + 0x18]
0x1ed62:    shl     rax, 4
0x1ed66:    lea     rbp, [r12 + rax + 0x28]
0x1ed6b:    nop     dword ptr [rax + rax]
0x1ed70:    mov     rdi, qword ptr [rbx]
0x1ed73:    add     rbx, 0x10
0x1ed77:    call    0x4800
0x1ed7c:    cmp     rbx, rbp
0x1ed7f:    jne     0x1ed70
0x1ed81:    mov     rdi, qword ptr [r12 + 8]
0x1ed86:    lea     rbx, [rip + 0xcdd3]
0x1ed8d:    cmp     rdi, rbx
0x1ed90:    je      0x1eda9
0x1ed92:    call    0x4800
0x1ed97:    mov     qword ptr [rip + 0xba8a], rbx
0x1ed9e:    mov     qword ptr [rip + 0xba77], 0x100
0x1eda9:    lea     rbx, [rip + 0xba70]
0x1edb0:    cmp     r12, rbx
0x1edb3:    je      0x1edc4
0x1edb5:    mov     rdi, r12
0x1edb8:    call    0x4800
0x1edbd:    mov     qword ptr [rip + 0xba4c], rbx
0x1edc4:    mov     dword ptr [rip + 0xba36], 1

```

```

0x1edce:      pop     rbx
0x1edcf:      pop     rbp
0x1edd0:      pop     r12
0x1edd2:      ret
0x1edd3:      nop     word ptr cs:[rax + rax]
0x1edde:      nop
0x1ede0:      endbr64
0x1ede4:      lea     rcx, [rip + 0xce75]
0x1edeb:      mov     rdx, -1
0x1edf2:      jmp     0x1e9e0
0x1e9e0
0x1edf7:      nop     word ptr [rax + rax]
0x1ee00:      endbr64
0x1ee04:      mov     rsi, rdi
0x1ee07:      lea     rcx, [rip + 0xce52]
0x1ee0e:      mov     rdx, -1
0x1ee15:      xor     edi, edi
0x1ee17:      jmp     0x1e9e0
0x1e9e0
0x1ee1c:      nop     dword ptr [rax]
0x1ee20:      endbr64
0x1ee24:      sub     rsp, 0x68
0x1ee28:      mov     r8d, esi
0x1ee2b:      pxor    xmm0, xmm0
0x1ee2f:      mov     rsi, rdx
0x1ee32:      mov     rax, qword ptr fs:[0x28]
0x1ee3b:      mov     qword ptr [rsp + 0x58], rax
0x1ee40:      xor     eax, eax
0x1ee42:      movups  xmmword ptr [rsp + 0x34], xmm0
0x1ee47:      mov     rdx, -1
0x1ee4e:      mov     rcx, rsp
0x1ee51:      movups  xmmword ptr [rsp + 0x44], xmm0
0x1ee56:      movdqa  xmm2, xmmword ptr [rsp + 0x40]
0x1ee5c:      mov     eax, dword ptr [rsp + 0x50]
0x1ee60:      mov     dword ptr [rsp + 0x30], r8d
0x1ee65:      movdqa  xmm1, xmmword ptr [rsp + 0x30]
0x1ee6b:      mov     dword ptr [rsp + 0x20], eax
0x1ee6f:      movaps  xmmword ptr [rsp], xmm1
0x1ee73:      movaps  xmmword ptr [rsp + 0x10], xmm2
0x1ee78:      call    0x1e9e0
0x1ee7d:      mov     rdx, qword ptr [rsp + 0x58]
0x1ee82:      sub     rdx, qword ptr fs:[0x28]
0x1ee8b:      jne     0x1ee92
0x1ee8d:      add     rsp, 0x68
0x1ee91:      ret
0x1ee92:      call    0x49d0
0x1ee97:      nop     word ptr [rax + rax]
0x1eea0:      endbr64

```

```

0x1eea4:    sub     rsp, 0x68
0x1eea8:    mov     r8d, esi
0x1eeab:    pxor    xmm0, xmm0
0x1eeaf:    mov     rsi, rdx
0x1eeb2:    mov     rax, qword ptr fs:[0x28]
0x1eebb:    mov     qword ptr [rsp + 0x58], rax
0x1eec0:    xor     eax, eax
0x1eec2:    movups  xmmword ptr [rsp + 0x34], xmm0
0x1eec7:    mov     rdx, rcx
0x1eeca:    mov     rcx, rsp
0x1eecd:    movups  xmmword ptr [rsp + 0x44], xmm0
0x1eed2:    movdqa  xmm2, xmmword ptr [rsp + 0x40]
0x1eed8:    mov     eax, dword ptr [rsp + 0x50]
0x1eedc:    mov     dword ptr [rsp + 0x30], r8d
0x1eee1:    movdqa  xmm1, xmmword ptr [rsp + 0x30]
0x1eee7:    mov     dword ptr [rsp + 0x20], eax
0x1eeeb:    movaps  xmmword ptr [rsp], xmm1
0x1eeef:    movaps  xmmword ptr [rsp + 0x10], xmm2
0x1eef4:    call    0x1e9e0
0x1eef9:    mov     rdx, qword ptr [rsp + 0x58]
0x1eefe:    sub     rdx, qword ptr fs:[0x28]
0x1ef07:    jne     0x1ef0e
0x1ef09:    add     rsp, 0x68
0x1ef0d:    ret
0x1ef0e:    call    0x49d0
0x1ef13:    nop     word ptr cs:[rax + rax]
0x1ef1e:    nop
0x1ef20:    endbr64
0x1ef24:    sub     rsp, 0x68
0x1ef28:    pxor    xmm0, xmm0
0x1ef2c:    mov     rdx, -1
0x1ef33:    mov     rax, qword ptr fs:[0x28]
0x1ef3c:    mov     qword ptr [rsp + 0x58], rax
0x1ef41:    xor     eax, eax
0x1ef43:    mov     dword ptr [rsp + 0x30], edi
0x1ef47:    mov     rcx, rsp
0x1ef4a:    xor     edi, edi
0x1ef4c:    movups  xmmword ptr [rsp + 0x34], xmm0
0x1ef51:    movdqa  xmm1, xmmword ptr [rsp + 0x30]
0x1ef57:    movups  xmmword ptr [rsp + 0x44], xmm0
0x1ef5c:    movdqa  xmm2, xmmword ptr [rsp + 0x40]
0x1ef62:    mov     eax, dword ptr [rsp + 0x50]
0x1ef66:    movaps  xmmword ptr [rsp], xmm1
0x1ef6a:    mov     dword ptr [rsp + 0x20], eax
0x1ef6e:    movaps  xmmword ptr [rsp + 0x10], xmm2
0x1ef73:    call    0x1e9e0
0x1ef78:    mov     rdx, qword ptr [rsp + 0x58]
0x1ef7d:    sub     rdx, qword ptr fs:[0x28]

```

```

0x1ef86:    jne     0x1ef8d
0x1ef88:    add     rsp, 0x68
0x1ef8c:    ret
0x1ef8d:    call    0x49d0
0x1ef92:    nop     word ptr cs:[rax + rax]
0x1ef9d:    nop     dword ptr [rax]
0x1efa0:    endbr64
0x1efa4:    sub     rsp, 0x38
0x1efa8:    movdqa  xmm0, xmmword ptr [rip + 0xccb0]
0x1efb0:    movdqa  xmm1, xmmword ptr [rip + 0xccb8]
0x1efb8:    mov     ecx, esi
0x1efba:    mov     rax, qword ptr fs:[0x28]
0x1efc3:    mov     qword ptr [rsp + 0x28], rax
0x1efc8:    xor     eax, eax
0x1efca:    mov     eax, dword ptr [rip + 0xccb0]
0x1efd0:    and     ecx, 0x1f
0x1efd3:    mov     r9, rsp
0x1efd6:    movaps  xmmword ptr [rsp], xmm0
0x1efda:    mov     dword ptr [rsp + 0x20], eax
0x1efde:    mov     eax, esi
0x1efe0:    shr     al, 5
0x1efe3:    movaps  xmmword ptr [rsp + 0x10], xmm1
0x1efe8:    movzx   eax, al
0x1efeb:    lea     rdx, [rsp + rax*4 + 4]
0x1eff0:    mov     esi, dword ptr [rdx]
0x1eff2:    mov     eax, esi
0x1eff4:    shr     eax, cl
0x1eff6:    not     eax
0x1eff8:    and     eax, 1
0x1effb:    shl     eax, cl
0x1effd:    mov     rcx, r9
0x1f000:    xor     eax, esi
0x1f002:    mov     rsi, rdi
0x1f005:    xor     edi, edi
0x1f007:    mov     dword ptr [rdx], eax
0x1f009:    mov     rdx, -1
0x1f010:    call    0x1e9e0
0x1f015:    mov     rdx, qword ptr [rsp + 0x28]
0x1f01a:    sub     rdx, qword ptr fs:[0x28]
0x1f023:    jne     0x1f02a
0x1f025:    add     rsp, 0x38
0x1f029:    ret
0x1f02a:    call    0x49d0
0x1f02f:    nop
0x1f030:    endbr64
0x1f034:    mov     esi, 0x3a
0x1f039:    jmp     0x1efa0
0x1efa0:

```



```

0x1e4e0:    cmp     dl, 0x7d
0x1e4e3:    je      0x1e72c
0x1e4e9:    jle     0x1e1f8
0x1e4ef:    cmp     dl, 0x7e
0x1e4f2:    je      0x1e365
0x1e4f8:    nop     dword ptr [rax + rax]
0x1e500:    lea     r14, [r12 + 1]
0x1e505:    cmp     qword ptr [rsp + 0x28], 1
0x1e50b:    mov     byte ptr [rsp + 0x30], dl
0x1e50f:    mov     r15, r14
0x1e512:    je      0x1e250
0x1e518:    mov     qword ptr [rsp + 0x70], 0
0x1e521:    lea     rcx, [rsp + 0x70]
0x1e526:    cmp     r13, -1
0x1e52a:    jne     0x1e574
0x1e52c:    mov     rdi, r11
0x1e52f:    mov     qword ptr [rsp + 0x58], r10
0x1e534:    mov     byte ptr [rsp + 0x50], r8b
0x1e539:    mov     qword ptr [rsp + 0x48], r9
0x1e53e:    mov     qword ptr [rsp + 0x40], rsi
0x1e543:    mov     qword ptr [rsp + 0x38], rcx
0x1e548:    mov     qword ptr [rsp + 0x30], r11
0x1e54d:    call    0x49b0
0x1e552:    mov     r10, qword ptr [rsp + 0x58]
0x1e557:    movzx   r8d, byte ptr [rsp + 0x50]
0x1e55d:    mov     r9, qword ptr [rsp + 0x48]
0x1e562:    mov     rsi, qword ptr [rsp + 0x40]
0x1e567:    mov     r13, rax
0x1e56a:    mov     rcx, qword ptr [rsp + 0x38]
0x1e56f:    mov     r11, qword ptr [rsp + 0x30]
0x1e574:    mov     rdx, r13
0x1e577:    lea     rdi, [rsp + 0x6c]
0x1e57c:    mov     qword ptr [rsp + 0x48], r11
0x1e581:    sub     rdx, r12
0x1e584:    mov     qword ptr [rsp + 0x40], r10
0x1e589:    mov     byte ptr [rsp + 0x38], r8b
0x1e58e:    mov     qword ptr [rsp + 0x30], r9
0x1e593:    mov     qword ptr [rsp + 0x58], rdx
0x1e598:    mov     qword ptr [rsp + 0x50], rsi
0x1e59d:    call    0x49f0
0x1e5a2:    mov     r9, qword ptr [rsp + 0x30]
0x1e5a7:    movzx   r8d, byte ptr [rsp + 0x38]
0x1e5ad:    test    rax, rax
0x1e5b0:    mov     r10, qword ptr [rsp + 0x40]
0x1e5b5:    mov     r11, qword ptr [rsp + 0x48]
0x1e5ba:    mov     rcx, rax
0x1e5bd:    je      0x1e1a4
0x1e5c3:    cmp     rax, -1

```

```

0x1e5c7:      je      0x1e828
0x1e5cd:      cmp     rax, -2
0x1e5d1:      mov     rsi, qword ptr [rsp + 0x50]
0x1e5d6:      mov     rdx, qword ptr [rsp + 0x58]
0x1e5db:      je      0x1e93b
0x1e5e1:      cmp     dword ptr [rsp + 0x10], 1
0x1e5e6:      je      0x1e95f
0x1e5ec:      mov     edi, dword ptr [rsp + 0x6c]
0x1e5f0:      mov     qword ptr [rsp + 0x50], r11
0x1e5f5:      mov     qword ptr [rsp + 0x48], r10
0x1e5fa:      mov     byte ptr [rsp + 0x40], r8b
0x1e5ff:      mov     qword ptr [rsp + 0x38], r9
0x1e604:      mov     qword ptr [rsp + 0x30], rcx
0x1e609:      call    0x4e80
0x1e60e:      movzx   r8d, byte ptr [rsp + 0x40]
0x1e614:      mov     rcx, qword ptr [rsp + 0x30]
0x1e619:      test    eax, eax
0x1e61b:      mov     r9, qword ptr [rsp + 0x38]
0x1e620:      mov     r10, qword ptr [rsp + 0x48]
0x1e625:      sete    al
0x1e628:      mov     r11, qword ptr [rsp + 0x50]
0x1e62d:      and     eax, r8d
0x1e630:      cmp     rcx, 1
0x1e634:      ja      0x1e2a4
0x1e63a:      jmp     0x1e299
0x1e299
0x1e63f:      mov     ebp, 0xc
0x1e644:      mov     eax, 0x66
0x1e649:      nop     dword ptr [rax]
0x1e650:      add     r12, 1
0x1e654:      test    r8b, r8b
0x1e657:      je      0x1e1e1
0x1e65d:      mov     ebp, eax
0x1e65f:      jmp     0x1e1cf
0x1e1cf
0x1e664:      nop     dword ptr [rax]
0x1e668:      mov     ebp, 0xb
0x1e66d:      mov     eax, 0x76
0x1e672:      jmp     0x1e650
0x1e650
0x1e674:      mov     ebp, 9
0x1e679:      mov     eax, 0x74
0x1e67e:      cmp     dword ptr [rsp + 0x10], 1
0x1e683:      jne     0x1e650
0x1e685:      jmp     0x1e380
0x1e380
0x1e68a:      nop     word ptr [rax + rax]
0x1e690:      add     r12, 1

```

```

0x1e694:      test    r8b, r8b
0x1e697:      jne     0x1e3f5
0x1e69d:      xor     ebp, ebp
0x1e69f:      jmp     0x1e1e1
0x1e1e1
0x1e6a4:      mov     ebp, 8
0x1e6a9:      mov     eax, 0x62
0x1e6ae:      jmp     0x1e650
0x1e650
0x1e6b0:      mov     ebp, 0xa
0x1e6b5:      mov     eax, 0x6e
0x1e6ba:      jmp     0x1e67e
0x1e67e
0x1e6bc:      mov     ebp, 7
0x1e6c1:      mov     eax, 0x61
0x1e6c6:      jmp     0x1e650
0x1e650
0x1e6c8:      mov     ebp, 0xd
0x1e6cd:      mov     eax, 0x72
0x1e6d2:      jmp     0x1e67e
0x1e67e
0x1e6d4:      nop     dword ptr [rax]
0x1e6d8:      cmp     dl, 0x7a
0x1e6db:      jg      0x1e4e0
0x1e6e1:      cmp     dl, 0x40
0x1e6e4:      je      0x1e500
0x1e6ea:      lea     ecx, [rbp - 0x41]
0x1e6ed:      mov     eax, 1
0x1e6f2:      movabs  rdi, 0x3fffffff53ffffff
0x1e6fc:      shl     rax, cl
0x1e6ff:      test    rax, rdi
0x1e702:      jne     0x1e215
0x1e708:      test    eax, 0xa4000000
0x1e70d:      jne     0x1e20a
0x1e713:      cmp     bpl, 0x5c
0x1e717:      jne     0x1e500
0x1e71d:      mov     eax, 0x5c
0x1e722:      jmp     0x1e67e
0x1e67e
0x1e727:      cmp     dl, 0x7b
0x1e72a:      jne     0x1e750
0x1e72c:      cmp     r13, 1
0x1e730:      setne   al
0x1e733:      cmp     r13, -1
0x1e737:      jne     0x1e741
0x1e739:      cmp     byte ptr [r11 + 1], 0
0x1e73e:      setne   al
0x1e741:      test    al, al

```

```

0x1e743:      jne      0x1e215
0x1e749:      jmp      0x1e365
0x1e365
0x1e74e:      nop
0x1e750:      cmp      dl, 0x7c
0x1e753:      je       0x1e20a
0x1e759:      jmp      0x1e240
0x1e240
0x1e75e:      nop
0x1e760:      lea      rax, [r12 + 2]
0x1e765:      lea      r15, [r12 + 1]
0x1e76a:      mov      ebp, 0x3f
0x1e76f:      cmp      rax, r13
0x1e772:      jae      0x1e1a4
0x1e778:      cmp      byte ptr [r11 + r12 + 1], 0x3f
0x1e77e:      jne      0x1e1a4
0x1e784:      movzx    ebp, byte ptr [r11 + r12 + 2]
0x1e78a:      cmp      bpl, 0x3e
0x1e78e:      ja       0x1e7a4
0x1e790:      movabs   rax, 0x7000a38200000000
0x1e79a:      bt       rax, rbp
0x1e79e:      jb       0x1e8f9
0x1e7a4:      mov      ebp, 0x3f
0x1e7a9:      jmp      0x1e1a4
0x1e1a4
0x1e7ae:      test     r12, r12
0x1e7b1:      jne      0x1e7be
0x1e7b3:      cmp      dword ptr [rsp + 0x10], 1
0x1e7b8:      je       0x1e380
0x1e7be:      mov      rsi, qword ptr [rsp + 0x20]
0x1e7c3:      mov      rax, rbx
0x1e7c6:      test     rsi, rsi
0x1e7c9:      je       0x1e7f4
0x1e7cb:      movzx    edx, byte ptr [rsi]
0x1e7ce:      test     dl, dl
0x1e7d0:      je       0x1e7f4
0x1e7d2:      sub      rsi, rbx
0x1e7d5:      mov      rcx, rsi
0x1e7d8:      mov      rsi, qword ptr [rsp + 8]
0x1e7dd:      nop      dword ptr [rax]
0x1e7e0:      cmp      r10, rax
0x1e7e3:      jbe      0x1e7e8
0x1e7e5:      mov      byte ptr [rsi + rax], dl
0x1e7e8:      add      rax, 1
0x1e7ec:      movzx    edx, byte ptr [rcx + rax]
0x1e7f0:      test     dl, dl
0x1e7f2:      jne      0x1e7e0
0x1e7f4:      cmp      rax, r10

```

```

0x1e7f7:    jae     0x1e802
0x1e7f9:    mov     rsi, qword ptr [rsp + 8]
0x1e7fe:    mov     byte ptr [rsi + rax], 0
0x1e802:    mov     rdx, qword ptr [rsp + 0x78]
0x1e807:    sub     rdx, qword ptr fs:[0x28]
0x1e810:    jne     0x1e9ca
0x1e816:    add     rsp, 0x88
0x1e81d:    pop     rbx
0x1e81e:    pop     rbp
0x1e81f:    pop     r12
0x1e821:    pop     r13
0x1e823:    pop     r14
0x1e825:    pop     r15
0x1e827:    ret
0x1e828:    mov     eax, r8d
0x1e82b:    xor     ecx, ecx
0x1e82d:    jmp     0x1e299
0x1e299
0x1e832:    mov     eax, dword ptr [rsp + 0x10]
0x1e836:    jbe     0x1e88a
0x1e838:    sub     eax, 5
0x1e83b:    cmp     eax, 1
0x1e83e:    ja      0x1e8c8
0x1e844:    cmp     dword ptr [rsp + 0x10], 6
0x1e849:    mov     edx, 0x60
0x1e84e:    mov     eax, 0x22
0x1e853:    cmovne  eax, edx
0x1e856:    lea     rcx, [rip + 0x2a3b]
0x1e85d:    lea     rdx, [rip + 0x3ef8]
0x1e864:    cmov     rdx, rcx
0x1e868:    mov     qword ptr [rsp + 0x20], rdx
0x1e86d:    test    r10, r10
0x1e870:    je      0x1e879
0x1e872:    mov     rsi, qword ptr [rsp + 8]
0x1e877:    mov     byte ptr [rsi], al
0x1e879:    mov     r8d, 1
0x1e87f:    mov     r9d, 1
0x1e885:    jmp     0x1e0b0
0x1e0b0
0x1e88a:    cmp     eax, 2
0x1e88d:    je      0x1e39e
0x1e893:    cmp     eax, 3
0x1e896:    jne     0x1e8c8
0x1e898:    test    r10, r10
0x1e89b:    je      0x1e8dc
0x1e89d:    mov     rax, qword ptr [rsp + 8]
0x1e8a2:    mov     r8d, 1
0x1e8a8:    mov     r9d, 1

```

```

0x1e8ae:    mov     byte ptr [rax], 0x22
0x1e8b1:    lea     rax, [rip + 0x29e0]
0x1e8b8:    mov     qword ptr [rsp + 0x20], rax
0x1e8bd:    jmp     0x1e0b0
0x1e0b0:
0x1e8c2:    nop     word ptr [rax + rax]
0x1e8c8:    mov     qword ptr [rsp + 0x20], 0
0x1e8d1:    xor     r8d, r8d
0x1e8d4:    xor     r9d, r9d
0x1e8d7:    jmp     0x1e0b0
0x1e0b0:
0x1e8dc:    lea     rax, [rip + 0x29b5]
0x1e8e3:    mov     r8d, 1
0x1e8e9:    mov     r9d, 1
0x1e8ef:    mov     qword ptr [rsp + 0x20], rax
0x1e8f4:    jmp     0x1e0b0
0x1e0b0:
0x1e8f9:    cmp     r10, rbx
0x1e8fc:    jbe     0x1e907
0x1e8fe:    mov     rax, qword ptr [rsp + 8]
0x1e903:    mov     byte ptr [rax + rbx], 0x3f
0x1e907:    lea     rax, [rbx + 1]
0x1e90b:    cmp     r10, rax
0x1e90e:    jbe     0x1e91a
0x1e910:    mov     rax, qword ptr [rsp + 8]
0x1e915:    mov     byte ptr [rax + rbx + 1], 0x5c
0x1e91a:    lea     rax, [rbx + 2]
0x1e91e:    cmp     r10, rax
0x1e921:    jbe     0x1e92d
0x1e923:    mov     rax, qword ptr [rsp + 8]
0x1e928:    mov     byte ptr [rax + rbx + 2], 0x3f
0x1e92d:    add     rbx, 3
0x1e931:    lea     r15, [r12 + 3]
0x1e936:    jmp     0x1e1a4
0x1e1a4:
0x1e93b:    xor     ecx, ecx
0x1e93d:    cmp     r13, r12
0x1e940:    ja      0x1e951
0x1e942:    jmp     0x1e9c2
0x1e9c2:
0x1e944:    nop     dword ptr [rax]
0x1e948:    add     rcx, 1
0x1e94c:    cmp     rdx, rcx
0x1e94f:    je      0x1e957
0x1e951:    cmp     byte ptr [rsi + rcx], 0
0x1e955:    jne     0x1e948
0x1e957:    mov     eax, r8d
0x1e95a:    jmp     0x1e630

```

```

0x1e630
0x1e95f:      cmp     rax, 1
0x1e963:      je      0x1e5ec
0x1e969:      lea     rdx, [r11 + r14]
0x1e96d:      add     rsi, rax
0x1e970:      jmp     0x1e985
0x1e985
0x1e972:      nop     word ptr [rax + rax]
0x1e978:      add     rdx, 1
0x1e97c:      cmp     rsi, rdx
0x1e97f:      je      0x1e5ec
0x1e985:      movzx   eax, byte ptr [rdx]
0x1e988:      sub     eax, 0x5b
0x1e98b:      cmp     al, 0x21
0x1e98d:      ja      0x1e978
0x1e98f:      movabs   rdi, 0x20000002b
0x1e999:      bt      rdi, rax
0x1e99d:      jae     0x1e978
0x1e99f:      mov     qword ptr [rsp + 0x20], r10
0x1e9a4:      mov     qword ptr [rsp + 0x10], r11
0x1e9a9:      call    0x49a0
0x1e9ae:      mov     r11, qword ptr [rsp + 0x10]
0x1e9b3:      mov     r10, qword ptr [rsp + 0x20]
0x1e9b8:      mov     qword ptr [rsp + 0x28], rax
0x1e9bd:      jmp     0x1e39e
0x1e39e
0x1e9c2:      mov     eax, r8d
0x1e9c5:      jmp     0x1e299
0x1e299
0x1e9ca:      call    0x49d0
0x1e9cf:      mov     r12, r15
0x1e9d2:      jmp     0x1e1e1
0x1e1e1
0x1e9d7:      nop     word ptr [rax + rax]
0x1e9e0:      push    r15
0x1e9e2:      mov     r15d, edi
0x1e9e5:      push    r14
0x1e9e7:      push    r13
0x1e9e9:      mov     r13, rdx
0x1e9ec:      push    r12
0x1e9ee:      mov     r12, rsi
0x1e9f1:      push    rbp
0x1e9f2:      mov     rbp, rcx
0x1e9f5:      push    rbx
0x1e9f6:      sub     rsp, 0x28
0x1e9fa:      call    0x4860
0x1e9ff:      mov     r14, qword ptr [rip + 0xbe0a]
0x1ea06:      mov     rbx, rax

```

```

0x1ea09:    mov     eax, dword ptr [rax]
0x1ea0b:    mov     dword ptr [rsp + 0xc], eax
0x1ea0f:    test    r15d, r15d
0x1ea12:    js      0x4f2a
0x1ea18:    cmp     dword ptr [rip + 0xbde5], r15d
0x1ea1f:    mov     r10d, dword ptr [rsp + 0xc]
0x1ea24:    ja      0x1ea89
0x1ea26:    lea     edx, [r15 + 1]
0x1ea2a:    lea     rax, [rip + 0xbdef]
0x1ea31:    mov     rsi, rdx
0x1ea34:    mov     dword ptr [rsp + 0x10], edx
0x1ea38:    shl     rsi, 4
0x1ea3c:    cmp     r14, rax
0x1ea3f:    je      0x1eb40
0x1ea45:    mov     rdi, r14
0x1ea48:    mov     qword ptr [rsp + 0x18], rdx
0x1ea4d:    call    0x1d690
0x1ea52:    mov     rdx, qword ptr [rsp + 0x18]
0x1ea57:    mov     qword ptr [rip + 0xbdb2], rax
0x1ea5e:    mov     r14, rax
0x1ea61:    mov     edi, dword ptr [rip + 0xbd9d]
0x1ea67:    xor     esi, esi
0x1ea69:    sub     rdx, rdi
0x1ea6c:    shl     rdi, 4
0x1ea70:    shl     rdx, 4
0x1ea74:    add     rdi, r14
0x1ea77:    call    0x4a50
0x1ea7c:    mov     eax, dword ptr [rsp + 0x10]
0x1ea80:    mov     r10d, dword ptr [rbx]
0x1ea83:    mov     dword ptr [rip + 0xbd7b], eax
0x1ea89:    movsxd  r8, r15d
0x1ea8c:    mov     r9, rbp
0x1ea8f:    mov     rcx, r13
0x1ea92:    mov     rdx, r12
0x1ea95:    shl     r8, 4
0x1ea99:    mov     dword ptr [rsp + 0x18], r10d
0x1ea9e:    lea     r15, [r14 + r8]
0x1eaa2:    mov     r8d, dword ptr [rbp]
0x1eaa6:    mov     r14, qword ptr [r15 + 8]
0x1eaaa:    mov     rsi, qword ptr [r15]
0x1eaad:    mov     rdi, r14
0x1eab0:    mov     qword ptr [rsp + 0x10], rsi
0x1eab5:    call    0x1e040
0x1eaba:    mov     r10d, dword ptr [rsp + 0x18]
0x1eabf:    mov     rsi, qword ptr [rsp + 0x10]
0x1eac4:    mov     dword ptr [rbx], r10d
0x1eac7:    cmp     rax, rsi
0x1eaca:    jb      0x1eb1f

```



```

0x1eacc:    lea     rsi, [rax + 1]
0x1ead0:    lea     rax, [rip + 0xd089]
0x1ead7:    mov     qword ptr [r15], rsi
0x1eada:    cmp     r14, rax
0x1eadd:    je      0x1eaf1
0x1eadf:    mov     rdi, r14
0x1eae2:    mov     qword ptr [rsp + 0x10], rsi
0x1eae7:    call    0x4800
0x1eaec:    mov     rsi, qword ptr [rsp + 0x10]
0x1eaf1:    mov     rdi, rsi
0x1eaf4:    mov     qword ptr [rsp + 0x10], rsi
0x1eaf9:    call    0x1d670
0x1eafe:    mov     r8d, dword ptr [rbp]
0x1eb02:    mov     r9, rbp
0x1eb05:    mov     rcx, r13
0x1eb08:    mov     qword ptr [r15 + 8], rax
0x1eb0c:    mov     rsi, qword ptr [rsp + 0x10]
0x1eb11:    mov     rdx, r12
0x1eb14:    mov     rdi, rax
0x1eb17:    mov     r14, rax
0x1eb1a:    call    0x1e040
0x1eb1f:    mov     eax, dword ptr [rsp + 0xc]
0x1eb23:    mov     dword ptr [rbx], eax
0x1eb25:    add     rsp, 0x28
0x1eb29:    mov     rax, r14
0x1eb2c:    pop     rbx
0x1eb2d:    pop     rbp
0x1eb2e:    pop     r12
0x1eb30:    pop     r13
0x1eb32:    pop     r14
0x1eb34:    pop     r15
0x1eb36:    ret
0x1eb37:    nop     word ptr [rax + rax]
0x1eb40:    xor     edi, edi
0x1eb42:    mov     qword ptr [rsp + 0x18], rdx
0x1eb47:    call    0x1d690
0x1eb4c:    movdqa  xmm0, xmmword ptr [rip + 0xbccc]
0x1eb54:    mov     rdx, qword ptr [rsp + 0x18]
0x1eb59:    mov     qword ptr [rip + 0xbcb0], rax
0x1eb60:    mov     r14, rax
0x1eb63:    movups  xmmword ptr [rax], xmm0
0x1eb66:    jmp     0x1ea61
0x1ea61:
0x1eb6b:    nop     dword ptr [rax + rax]
0x1eb70:    endbr64
0x1eb74:    push    r12
0x1eb76:    push    rbp
0x1eb77:    mov     rbp, rdi

```

```

0x1eb7a:    push    rbx
0x1eb7b:    call    0x4860
0x1eb80:    test    rbp, rbp
0x1eb83:    mov     esi, 0x24
0x1eb88:    mov     r12d, dword ptr [rax]
0x1eb8b:    mov     rbx, rax
0x1eb8e:    lea     rax, [rip + 0xd0cb]
0x1eb95:    cmov     rbp, rax
0x1eb99:    mov     rdi, rbp
0x1eb9c:    call    0x1d760
0x1eba1:    mov     dword ptr [rbx], r12d
0x1eba4:    pop     rbx
0x1eba5:    pop     rbp
0x1eba6:    pop     r12
0x1eba8:    ret
0x1eba9:    nop     dword ptr [rax]
0x1ebb0:    endbr64
0x1ebb4:    test    rdi, rdi
0x1ebb7:    lea     rax, [rip + 0xd0a2]
0x1ebbe:    cmov     rdi, rax
0x1ebc2:    mov     eax, dword ptr [rdi]
0x1ebc4:    ret
0x1ebc5:    nop     word ptr cs:[rax + rax]
0x1ebd0:    endbr64
0x1ebd4:    test    rdi, rdi
0x1ebd7:    lea     rax, [rip + 0xd082]
0x1ebde:    cmov     rdi, rax
0x1ebe2:    mov     dword ptr [rdi], esi
0x1ebe4:    ret
0x1ebe5:    nop     word ptr cs:[rax + rax]
0x1ebf0:    endbr64
0x1ebf4:    test    rdi, rdi
0x1ebf7:    lea     rax, [rip + 0xd062]
0x1ebfe:    mov     ecx, esi
0x1ec00:    cmov     rdi, rax
0x1ec04:    mov     eax, esi
0x1ec06:    and     ecx, 0x1f
0x1ec09:    shr     al, 5
0x1ec0c:    movzx   eax, al
0x1ec0f:    lea     rsi, [rdi + rax*4 + 4]
0x1ec14:    mov     edi, dword ptr [rsi]
0x1ec16:    mov     eax, edi
0x1ec18:    shr     eax, cl
0x1ec1a:    xor     edx, eax
0x1ec1c:    and     eax, 1
0x1ec1f:    and     edx, 1
0x1ec22:    shl     edx, cl
0x1ec24:    xor     edx, edi

```

```

0x1ec26:    mov     dword ptr [rsi], edx
0x1ec28:    ret
0x1ec29:    nop     dword ptr [rax]
0x1ec30:    endbr64
0x1ec34:    push    r15
0x1ec36:    lea     rax, [rip + 0xd023]
0x1ec3d:    push    r14
0x1ec3f:    mov     r14, rdx
0x1ec42:    push    r13
0x1ec44:    mov     r13, rsi
0x1ec47:    push    r12
0x1ec49:    mov     r12, rdi
0x1ec4c:    push    rbp
0x1ec4d:    push    rbx
0x1ec4e:    mov     rbx, r8
0x1ec51:    sub     rsp, 0x18
0x1ec55:    test    r8, r8
0x1ec58:    cmovbe  rbx, rax
0x1ec5c:    mov     qword ptr [rsp + 8], rcx
0x1ec61:    call    0x4860
0x1ec66:    mov     rcx, qword ptr [rsp + 8]
0x1ec6b:    mov     rdx, r14
0x1ec6e:    mov     rsi, r13
0x1ec71:    mov     r15d, dword ptr [rax]
0x1ec74:    mov     r8d, dword ptr [rbx]
0x1ec77:    mov     rbp, rax
0x1ec7a:    mov     r9, rbx
0x1ec7d:    mov     rdi, r12
0x1ec80:    call    0x1e040
0x1ec85:    mov     dword ptr [rbp], r15d
0x1ec89:    add     rsp, 0x18
0x1ec8d:    pop     rbx
0x1ec8e:    pop     rbp
0x1ec8f:    pop     r12
0x1ec91:    pop     r13
0x1ec93:    pop     r14
0x1ec95:    pop     r15
0x1ec97:    ret
0x1ec98:    nop     dword ptr [rax + rax]
0x1eca0:    endbr64
0x1eca4:    push    r15
0x1eca6:    mov     r15, rdi
0x1eca9:    push    r14
0x1ecab:    mov     r14, rsi
0x1ecae:    push    r13
0x1ecb0:    push    r12
0x1ecb2:    push    rbp
0x1ecb3:    push    rbx

```

```

0x1ecb4:    mov     rbx, rdx
0x1ecb7:    sub     rsp, 0x18
0x1ecbb:    call    0x4860
0x1ecc0:    test    rbx, rbx
0x1ecc3:    mov     rcx, r14
0x1ecc6:    mov     rdx, r15
0x1ecc9:    mov     r12d, dword ptr [rax]
0x1eccc:    mov     rbp, rax
0x1eccf:    lea     rax, [rip + 0xcf8a]
0x1ecd6:    cmove    rbx, rax
0x1ecda:    xor     esi, esi
0x1ecdc:    xor     edi, edi
0x1ecde:    mov     r8d, dword ptr [rbx]
0x1ece1:    mov     r9, rbx
0x1ece4:    call    0x1e040
0x1ece9:    mov     dword ptr [rbp], r12d
0x1eced:    lea     r13, [rax + 1]
0x1ecf1:    mov     dword ptr [rsp + 0xc], r12d
0x1ecf6:    mov     rdi, r13
0x1ecf9:    call    0x1d670
0x1ecfe:    mov     r8d, dword ptr [rbx]
0x1ed01:    mov     r9, rbx
0x1ed04:    mov     rcx, r14
0x1ed07:    mov     rdx, r15
0x1ed0a:    mov     rsi, r13
0x1ed0d:    mov     rdi, rax
0x1ed10:    mov     r12, rax
0x1ed13:    call    0x1e040
0x1ed18:    mov     eax, dword ptr [rsp + 0xc]
0x1ed1c:    mov     dword ptr [rbp], eax
0x1ed1f:    add     rsp, 0x18
0x1ed23:    mov     rax, r12
0x1ed26:    pop     rbx
0x1ed27:    pop     rbp
0x1ed28:    pop     r12
0x1ed2a:    pop     r13
0x1ed2c:    pop     r14
0x1ed2e:    pop     r15
0x1ed30:    ret
0x1ed31:    nop     word ptr cs:[rax + rax]
0x1ed3c:    nop     dword ptr [rax]
0x1ed40:    endbr64
0x1ed44:    mov     eax, dword ptr [rip + 0xbaba]
0x1ed4a:    push    r12
0x1ed4c:    mov     r12, qword ptr [rip + 0xbabd]
0x1ed53:    push    rbp
0x1ed54:    push    rbx
0x1ed55:    cmp     eax, 1

```

```

0x1ed58:      jbe      0x1ed81
0x1ed5a:      sub      eax, 2
0x1ed5d:      lea      rbx, [r12 + 0x18]
0x1ed62:      shl      rax, 4
0x1ed66:      lea      rbp, [r12 + rax + 0x28]
0x1ed6b:      nop      dword ptr [rax + rax]
0x1ed70:      mov      rdi, qword ptr [rbx]
0x1ed73:      add      rbx, 0x10
0x1ed77:      call     0x4800
0x1ed7c:      cmp      rbx, rbp
0x1ed7f:      jne      0x1ed70
0x1ed81:      mov      rdi, qword ptr [r12 + 8]
0x1ed86:      lea      rbx, [rip + 0xcdd3]
0x1ed8d:      cmp      rdi, rbx
0x1ed90:      je       0x1eda9
0x1ed92:      call     0x4800
0x1ed97:      mov      qword ptr [rip + 0xba8a], rbx
0x1ed9e:      mov      qword ptr [rip + 0xba77], 0x100
0x1eda9:      lea      rbx, [rip + 0xba70]
0x1edb0:      cmp      r12, rbx
0x1edb3:      je       0x1edc4
0x1edb5:      mov      rdi, r12
0x1edb8:      call     0x4800
0x1edbd:      mov      qword ptr [rip + 0xba4c], rbx
0x1edc4:      mov      dword ptr [rip + 0xba36], 1
0x1edce:      pop      rbx
0x1edcf:      pop      rbp
0x1edd0:      pop      r12
0x1edd2:      ret
0x1edd3:      nop      word ptr cs:[rax + rax]
0x1edde:      nop
0x1ede0:      endbr64
0x1ede4:      lea      rcx, [rip + 0xce75]
0x1edeb:      mov      rdx, -1
0x1edf2:      jmp      0x1e9e0
0x1e9e0
0x1edf7:      nop      word ptr [rax + rax]
0x1ee00:      endbr64
0x1ee04:      mov      rsi, rdi
0x1ee07:      lea      rcx, [rip + 0xce52]
0x1ee0e:      mov      rdx, -1
0x1ee15:      xor      edi, edi
0x1ee17:      jmp      0x1e9e0
0x1e9e0
0x1ee1c:      nop      dword ptr [rax]
0x1ee20:      endbr64
0x1ee24:      sub      rsp, 0x68
0x1ee28:      mov      r8d, esi

```

```

0x1ee2b:    pxor     xmm0, xmm0
0x1ee2f:    mov     rsi, rdx
0x1ee32:    mov     rax, qword ptr fs:[0x28]
0x1ee3b:    mov     qword ptr [rsp + 0x58], rax
0x1ee40:    xor     eax, eax
0x1ee42:    movups  xmmword ptr [rsp + 0x34], xmm0
0x1ee47:    mov     rdx, -1
0x1ee4e:    mov     rcx, rsp
0x1ee51:    movups  xmmword ptr [rsp + 0x44], xmm0
0x1ee56:    movdqa  xmm2, xmmword ptr [rsp + 0x40]
0x1ee5c:    mov     eax, dword ptr [rsp + 0x50]
0x1ee60:    mov     dword ptr [rsp + 0x30], r8d
0x1ee65:    movdqa  xmm1, xmmword ptr [rsp + 0x30]
0x1ee6b:    mov     dword ptr [rsp + 0x20], eax
0x1ee6f:    movaps  xmmword ptr [rsp], xmm1
0x1ee73:    movaps  xmmword ptr [rsp + 0x10], xmm2
0x1ee78:    call    0x1e9e0
0x1ee7d:    mov     rdx, qword ptr [rsp + 0x58]
0x1ee82:    sub     rdx, qword ptr fs:[0x28]
0x1ee8b:    jne     0x1ee92
0x1ee8d:    add     rsp, 0x68
0x1ee91:    ret
0x1ee92:    call    0x49d0
0x1ee97:    nop     word ptr [rax + rax]
0x1eea0:    endbr64
0x1eea4:    sub     rsp, 0x68
0x1eea8:    mov     r8d, esi
0x1eeab:    pxor     xmm0, xmm0
0x1eeaf:    mov     rsi, rdx
0x1eeb2:    mov     rax, qword ptr fs:[0x28]
0x1eebb:    mov     qword ptr [rsp + 0x58], rax
0x1eec0:    xor     eax, eax
0x1eec2:    movups  xmmword ptr [rsp + 0x34], xmm0
0x1eec7:    mov     rdx, rcx
0x1eeca:    mov     rcx, rsp
0x1eecd:    movups  xmmword ptr [rsp + 0x44], xmm0
0x1eed2:    movdqa  xmm2, xmmword ptr [rsp + 0x40]
0x1eed8:    mov     eax, dword ptr [rsp + 0x50]
0x1eedc:    mov     dword ptr [rsp + 0x30], r8d
0x1eee1:    movdqa  xmm1, xmmword ptr [rsp + 0x30]
0x1eee7:    mov     dword ptr [rsp + 0x20], eax
0x1eeeb:    movaps  xmmword ptr [rsp], xmm1
0x1eeef:    movaps  xmmword ptr [rsp + 0x10], xmm2
0x1eef4:    call    0x1e9e0
0x1eef9:    mov     rdx, qword ptr [rsp + 0x58]
0x1eefe:    sub     rdx, qword ptr fs:[0x28]
0x1ef07:    jne     0x1ef0e
0x1ef09:    add     rsp, 0x68

```

```

0x1ef0d:      ret
0x1ef0e:      call    0x49d0
0x1ef13:      nop     word ptr cs:[rax + rax]
0x1ef1e:      nop
0x1ef20:      endbr64
0x1ef24:      sub     rsp, 0x68
0x1ef28:      pxor    xmm0, xmm0
0x1ef2c:      mov     rdx, -1
0x1ef33:      mov     rax, qword ptr fs:[0x28]
0x1ef3c:      mov     qword ptr [rsp + 0x58], rax
0x1ef41:      xor     eax, eax
0x1ef43:      mov     dword ptr [rsp + 0x30], edi
0x1ef47:      mov     rcx, rsp
0x1ef4a:      xor     edi, edi
0x1ef4c:      movups   xmmword ptr [rsp + 0x34], xmm0
0x1ef51:      movdqa   xmm1, xmmword ptr [rsp + 0x30]
0x1ef57:      movups   xmmword ptr [rsp + 0x44], xmm0
0x1ef5c:      movdqa   xmm2, xmmword ptr [rsp + 0x40]
0x1ef62:      mov     eax, dword ptr [rsp + 0x50]
0x1ef66:      movaps   xmmword ptr [rsp], xmm1
0x1ef6a:      mov     dword ptr [rsp + 0x20], eax
0x1ef6e:      movaps   xmmword ptr [rsp + 0x10], xmm2
0x1ef73:      call    0x1e9e0
0x1ef78:      mov     rdx, qword ptr [rsp + 0x58]
0x1ef7d:      sub     rdx, qword ptr fs:[0x28]
0x1ef86:      jne     0x1ef8d
0x1ef88:      add     rsp, 0x68
0x1ef8c:      ret
0x1ef8d:      call    0x49d0
0x1ef92:      nop     word ptr cs:[rax + rax]
0x1ef9d:      nop     dword ptr [rax]
0x1efa0:      endbr64
0x1efa4:      sub     rsp, 0x38
0x1efa8:      movdqa   xmm0, xmmword ptr [rip + 0xccb0]
0x1efb0:      movdqa   xmm1, xmmword ptr [rip + 0xccb8]
0x1efb8:      mov     ecx, esi
0x1efba:      mov     rax, qword ptr fs:[0x28]
0x1efc3:      mov     qword ptr [rsp + 0x28], rax
0x1efc8:      xor     eax, eax
0x1efca:      mov     eax, dword ptr [rip + 0xccb0]
0x1efd0:      and     ecx, 0x1f
0x1efd3:      mov     r9, rsp
0x1efd6:      movaps   xmmword ptr [rsp], xmm0
0x1efda:      mov     dword ptr [rsp + 0x20], eax
0x1efde:      mov     eax, esi
0x1efe0:      shr     al, 5
0x1efe3:      movaps   xmmword ptr [rsp + 0x10], xmm1
0x1efe8:      movzx    eax, al

```

```

0x1efeb:    lea     rdx, [rsp + rax*4 + 4]
0x1eff0:    mov     esi, dword ptr [rdx]
0x1eff2:    mov     eax, esi
0x1eff4:    shr     eax, cl
0x1eff6:    not     eax
0x1eff8:    and     eax, 1
0x1effb:    shl     eax, cl
0x1effd:    mov     rcx, r9
0x1f000:    xor     eax, esi
0x1f002:    mov     rsi, rdi
0x1f005:    xor     edi, edi
0x1f007:    mov     dword ptr [rdx], eax
0x1f009:    mov     rdx, -1
0x1f010:    call    0x1e9e0
0x1f015:    mov     rdx, qword ptr [rsp + 0x28]
0x1f01a:    sub     rdx, qword ptr fs:[0x28]
0x1f023:    jne     0x1f02a
0x1f025:    add     rsp, 0x38
0x1f029:    ret
0x1f02a:    call    0x49d0
0x1f02f:    nop
0x1f030:    endbr64
0x1f034:    mov     esi, 0x3a
0x1f039:    jmp     0x1efa0
0x1efa0

```

[8]: *# sorting the instruction according to address, for easier printing*

```

address_keys = list(address_instruction_info_dict.keys())
address_keys.sort()
address_instruction_info_dict = {i: address_instruction_info_dict[i] for i in
    ↪address_keys}

```

[9]: *# display and save into file*

```

current_address = code_section.offset

is_new_block = False

with open('instructions.s', 'w') as out_file:
    while current_address < textSectionEnd:

        if current_address in address_instruction_info_dict:

```



```

if is_new_block == True:
    print('\n\n')
    print("0x%x:\t -----" %(current_address-2))
    print("0x%x:\t -----" %(current_address-1))

    out_file.write("0x%x:\t -----" %(current_address-1)+ '\n')

is_new_block = False

info = address_instruction_info_dict[current_address]
print("0x%x:\t%s\t%s" %(info['address'],
                        info['mnemonic'], info['op_str']))
out_file.write("0x%x:\t%s\t%s" %(info['address'],
                        info['mnemonic'], info['op_str'])+ '\n')

current_address+= info['size']

else:

    if is_new_block == False:
        print("0x%x:\t -----" %(current_address))
        out_file.write("0x%x:\t -----" %(current_address)+ '\n')

    current_address+=1
    is_new_block = True

```

```

0x4f20: mov     byte ptr [rsp + 8], 0
0x4f25: jmp     0x107fa
0x4f2a: call    0x4850
0x4f2f: nop
0x4f30: endbr64
0x4f34: push    r15
0x4f36: push    r14
0x4f38: lea     r14, [rip + 0x1b669]
0x4f3f: push    r13
0x4f41: lea     r13, [rip + 0x1b2b4]
0x4f48: push    r12
0x4f4a: push    rbp
0x4f4b: mov     ebp, edi
0x4f4d: push    rbx
0x4f4e: mov     rbx, rsi
0x4f51: sub     rsp, 0x88
0x4f58: mov     rax, qword ptr fs:[0x28]
0x4f61: mov     qword ptr [rsp + 0x78], rax

```

```

0x4f66: xor     eax, eax
0x4f68: mov     dword ptr [rsp + 0x34], 0
0x4f70: mov     qword ptr [rsp + 0x50], 0
0x4f79: mov     qword ptr [rsp + 0x58], 0
0x4f82: call    0x18c50
0x4f87: lea     rsi, [rip + 0x1d457]
0x4f8e: mov     edi, 6
0x4f93: call    0x4d00
0x4f98: mov     rax, qword ptr [rbx]
0x4f9b: lea     rdi, [rip + 0x1b5be]
0x4fa2: mov     qword ptr [rip + 0x2593f], rax
0x4fa9: call    0x47e0
0x4fae: lea     rdx, [rip + 0x1c729]
0x4fb5: test    rax, rax
0x4fb8: cmov     rax, rdx
0x4fbc: mov     qword ptr [rip + 0x2592d], rax
0x4fc3: xor     eax, eax
0x4fc5: call    0x17660
0x4fca: xor     eax, eax
0x4fcc: call    0x18420
0x4fd1: xor     eax, eax
0x4fd3: call    0x18160
0x4fd8: mov     edi, 0x25
0x4fdd: call    0x1d670
0x4fe2: lea     r8, [rip + 0x1b57c]
0x4fe9: mov     esi, 1
0x4fee: lea     rcx, [rip + 0x1b586]
0x4ff5: mov     rdi, rax
0x4ff8: mov     qword ptr [rip + 0x258d1], rax
0x4fff: xor     eax, eax
0x5001: mov     rdx, -1
0x5008: call    0x4f10
0x500d: lea     rdi, [rip + 0x1b571]
0x5014: call    0x47e0
0x5019: mov     dword ptr [rsp], 1
0x5020: cmp     rax, 1
0x5024: sbb     r12d, r12d
0x5027: xor     r15d, r15d
0x502a: or      r12d, 1
0x502e: mov     rdx, r14
0x5031: mov     rsi, rbx
0x5034: mov     edi, ebp
0x5036: call    0x4de0
0x503b: cmp     eax, -1
0x503e: je      0x510e
0x5044: sub     eax, 0x3f
0x5047: cmp     eax, 0x37
0x504a: ja      0x5057

```

```

0x504c: movsxd  rax, dword ptr [r13 + rax*4]
0x5051: add      rax, r13
0x5054: jmp      rax
0x5057: mov      rcx, qword ptr [rip + 0x2588a]
0x505e: mov      rdi, qword ptr [rip + 0x2581b]
0x5065: lea      rdx, [rip + 0x1b529]
0x506c: xor      eax, eax
0x506e: mov      esi, 1
0x5073: mov      r15d, 1
0x5079: call     0x4e60
0x507e: mov      rax, qword ptr [rsp + 0x78]
0x5083: sub      rax, qword ptr fs:[0x28]
0x508c: jne      0x7475
0x5092: add      rsp, 0x88
0x5099: mov      eax, r15d
0x509c: pop      rbx
0x509d: pop      rbp
0x509e: pop      r12
0x50a0: pop      r13
0x50a2: pop      r14
0x50a4: pop      r15
0x50a6: ret
0x50a7: lea      rcx, [rip + 0x1b4e1]
0x50ae: mov      edi, 1
0x50b3: xor      eax, eax
0x50b5: xor      r15d, r15d
0x50b8: lea      rdx, [rip + 0x1b4a6]
0x50bf: lea      rsi, [rip + 0x1b517]
0x50c6: call     0x4d20
0x50cb: jmp      0x507e
0x50cd: mov      r15d, 1
0x50d3: jmp      0x502e
0x50d8: xor      eax, eax
0x50da: xor      r15d, r15d
0x50dd: call     0x9720
0x50e2: jmp      0x507e
0x50e4: mov      r12d, 1
0x50ea: jmp      0x502e
0x50ef: mov      dword ptr [rsp], 0
0x50f6: jmp      0x502e
0x50fb: mov      r15d, 1
0x5101: jmp      0x507e
0x5106: xor      r12d, r12d
0x5109: jmp      0x502e
0x510e: movsxd  rax, dword ptr [rip + 0x25733]
0x5115: lea      r13, [rip + 0x1c5c2]
0x511c: cmp      eax, ebp
0x511e: jl       0x6f96

```

```

0x5124: mov     r14, r13
0x5127: mov     edi, 0xa
0x512c: lea     rbp, [rip + 0x1b51c]
0x5133: call    0x4810
0x5138: lea     rax, [rip + 0x1b4a5]
0x513f: lea     r9, [rip + 0x1b469]
0x5146: mov     edi, 1
0x514b: push    rsi
0x514c: lea     r8, [rip + 0x1b465]
0x5153: lea     rcx, [rip + 0x1b435]
0x515a: push    rax
0x515b: lea     rdx, [rip + 0x1b403]
0x5162: lea     rsi, [rip + 0x1b468]
0x5169: xor     eax, eax
0x516b: call    0x4d20
0x5170: mov     rsi, qword ptr [rip + 0x24b09]
0x5177: mov     edi, 1
0x517c: xor     eax, eax
0x517e: call    0x4d20
0x5183: xor     eax, eax
0x5185: lea     rdx, [rip + 0x1b464]
0x518c: lea     rsi, [rip + 0x1b464]
0x5193: lea     rdi, [rip + 0x1b459]
0x519a: call    0x92b0
0x519f: xor     eax, eax
0x51a1: lea     rdx, [rip + 0x1b456]
0x51a8: lea     rsi, [rip + 0x1b458]
0x51af: lea     rdi, [rip + 0x1b44d]
0x51b6: call    0x92b0
0x51bb: xor     esi, esi
0x51bd: xor     eax, eax
0x51bf: lea     rdx, [rip + 0x1b447]
0x51c6: lea     rdi, [rip + 0x1b445]
0x51cd: call    0x92b0
0x51d2: xor     eax, eax
0x51d4: lea     rdx, [rip + 0x1b443]
0x51db: lea     rsi, [rip + 0x1b448]
0x51e2: lea     rdi, [rip + 0x1b43d]
0x51e9: call    0x92b0
0x51ee: xor     esi, esi
0x51f0: xor     eax, eax
0x51f2: lea     rdx, [rip + 0x1b437]
0x51f9: lea     rdi, [rip + 0x1b435]
0x5200: call    0x92b0
0x5205: lea     rdx, [rip + 0x1b433]
0x520c: xor     esi, esi
0x520e: xor     eax, eax
0x5210: lea     rdi, [rip + 0x1b42d]

```

```

0x5217: call    0x92b0
0x521c: xor      edi, edi
0x521e: call    0xc3d0
0x5223: xor      eax, eax
0x5225: call    0x179a0
0x522a: mov      rdi, rbp
0x522d: call    0x17c50
0x5232: xor      eax, eax
0x5234: call    0x18660
0x5239: lea      rdi, [rip + 0x1b41b]
0x5240: call    0x17c50
0x5245: lea      rax, [rsp + 0x44]
0x524a: xor      edi, edi
0x524c: mov      rsi, rax
0x524f: mov      qword ptr [rsp + 0x20], rax
0x5254: xor      eax, eax
0x5256: call    0x9310
0x525b: mov      dword ptr [rsp + 0x1c], eax
0x525f: mov      eax, dword ptr [rsp + 0x44]
0x5263: add      dword ptr [rip + 0x25673], eax
0x5269: xor      eax, eax
0x526b: call    0x17280
0x5270: xor      eax, eax
0x5272: call    0x17a30
0x5277: pop      rdi
0x5278: lea      rdi, [rip + 0x1b3e9]
0x527f: pop      r8
0x5281: cmp      eax, 1
0x5284: adc      dword ptr [rip + 0x25651], 0
0x528b: call    0x47e0
0x5290: mov      rbx, rax
0x5293: test     rax, rax
0x5296: je       0x7011
0x529c: mov      rdi, rbx
0x529f: call    0x49b0
0x52a4: lea      rdi, [rax + 0x21]
0x52a8: call    0x1d670
0x52ad: mov      rdi, rbx
0x52b0: mov      qword ptr [rip + 0x26819], rax
0x52b7: call    0x49b0
0x52bc: lea      rdi, [rax + 0x21]
0x52c0: call    0x1d670
0x52c5: xor      edi, edi
0x52c7: mov      qword ptr [rip + 0x267fa], rax
0x52ce: call    0x4be0
0x52d3: mov      edi, eax
0x52d5: call    0x4ae0
0x52da: call    0x4ee0

```

```

0x52df: mov     dword ptr [rsp + 0x1c], eax
0x52e3: call    0x4ee0
0x52e8: mov     dword ptr [rsp + 0x18], eax
0x52ec: call    0x4920
0x52f1: pxor    xmm0, xmm0
0x52f5: mov     r8, rbx
0x52f8: mov     rdi, qword ptr [rip + 0x267d1]
0x52ff: mov     r9d, eax
0x5302: push    rax
0x5303: mov     edx, dword ptr [rsp + 0x24]
0x5307: lea     rcx, [rip + 0x1b361]
0x530e: mov     esi, 1
0x5313: xor     eax, eax
0x5315: cvtsi2sd    xmm0, edx
0x5319: mulsd   xmm0, qword ptr [rip + 0x1b89f]
0x5321: mulsd   xmm0, qword ptr [rip + 0x1b89f]
0x5329: cvtt2sd    edx, xmm0
0x532d: add     edx, 1
0x5330: push    rdx
0x5331: or      rdx, 0xffffffffffffffff
0x5335: call    0x4f10
0x533a: call    0x4920
0x533f: mov     r10d, dword ptr [rsp + 0x28]
0x5344: pxor    xmm0, xmm0
0x5348: mov     rdi, qword ptr [rip + 0x26779]
0x534f: mov     r9d, eax
0x5352: mov     r8, rbx
0x5355: mov     esi, 1
0x535a: xor     eax, eax
0x535c: cvtsi2sd    xmm0, r10d
0x5361: mulsd   xmm0, qword ptr [rip + 0x1b857]
0x5369: lea     rcx, [rip + 0x1b30e]
0x5370: mulsd   xmm0, qword ptr [rip + 0x1b850]
0x5378: cvtt2sd    edx, xmm0
0x537c: add     edx, 1
0x537f: mov     dword ptr [rsp], edx
0x5382: or      rdx, 0xffffffffffffffff
0x5386: call    0x4f10
0x538b: lea     rdi, [rip + 0x1b2fb]
0x5392: call    0x17c50
0x5397: pop     rdx
0x5398: pop     rcx
0x5399: cmp     r12d, -1
0x539d: je      0x7092
0x53a3: cmp     dword ptr [rsp], 0
0x53a7: mov     dword ptr [rip + 0x248b2], r12d
0x53ae: jne     0x6f80
0x53b4: mov     eax, dword ptr [rsp]

```

```

0x53b7: mov     edi, eax
0x53b9: mov     dword ptr [rip + 0x25519], eax
0x53bf: call    0x9af0
0x53c4: lea     rdi, [rip + 0x1b2e7]
0x53cb: call    0x17c50
0x53d0: xor     eax, eax
0x53d2: cmp     dword ptr [rip + 0x24887], 1
0x53d9: je      0x7050
0x53df: xor     esi, esi
0x53e1: lea     rdi, [rip + 0x1b2e8]
0x53e8: mov     dword ptr [rip + 0x2486e], eax
0x53ee: call    0x17e50
0x53f3: lea     rsi, [rip + 0x1cfeb]
0x53fa: lea     rdi, [rip + 0x1b2dd]
0x5401: mov     dword ptr [rip + 0x25501], eax
0x5407: call    0x17f10
0x540c: lea     rsi, [rip + 0x1cfd2]
0x5413: lea     rdi, [rip + 0x1b2d3]
0x541a: mov     qword ptr [rip + 0x2484f], rax
0x5421: call    0x17f10
0x5426: cmp     dword ptr [rip + 0x24833], 0
0x542d: lea     rdi, [rip + 0x24814]
0x5434: mov     qword ptr [rip + 0x2482d], rax
0x543b: lea     rax, [rip + 0x247ee]
0x5442: cmove   rdi, rax
0x5446: call    0x17c50
0x544b: mov     edx, 8
0x5450: lea     rsi, [rip + 0x25229]
0x5457: lea     rdi, [rip + 0x251e2]
0x545e: call    0x17f70
0x5463: mov     rdi, rbp
0x5466: call    0x17c50
0x546b: xor     eax, eax
0x546d: call    0x18660
0x5472: lea     rdi, [rip + 0x1b1e2]
0x5479: call    0x17c50
0x547e: mov     rsi, qword ptr [rsp + 0x10]
0x5483: mov     edi, dword ptr [rsp + 0xc]
0x5487: xor     eax, eax
0x5489: call    0x9310
0x548e: mov     edx, dword ptr [rsp + 0x34]
0x5492: add     dword ptr [rip + 0x25444], edx
0x5498: cmp     eax, 0x800
0x549d: je      0x7066
0x54a3: call    0x49e0
0x54a8: test    eax, eax
0x54aa: jne     0x54b3
0x54ac: mov     byte ptr [rip + 0x24753], 0x23

```

```

0x54b3: xor     eax, eax
0x54b5: call    0x18fe0
0x54ba: mov     rbp, rax
0x54bd: test    rax, rax
0x54c0: je      0x73ec
0x54c6: xor     eax, eax
0x54c8: call    0x9e00
0x54cd: xor     eax, eax
0x54cf: call    0x16c10
0x54d4: xor     eax, eax
0x54d6: call    0xd060
0x54db: mov     rdi, qword ptr [rip + 0x2478e]
0x54e2: call    0xddc0
0x54e7: cmp     byte ptr [r13], 0x2f
0x54ec: je      0x7001
0x54f2: mov     rdi, rbp
0x54f5: call    0x49b0
0x54fa: mov     rdi, r13
0x54fd: mov     r12, rax
0x5500: call    0x49b0
0x5505: lea     rdi, [r12 + rax + 2]
0x550a: call    0x1d670
0x550f: mov     r9, r13
0x5512: mov     r8, rbp
0x5515: or      rdx, 0xffffffffffffffff
0x5519: mov     r12, rax
0x551c: mov     rdi, rax
0x551f: mov     esi, 1
0x5524: xor     eax, eax
0x5526: lea     rcx, [rip + 0x1b1d4]
0x552d: call    0x4f10
0x5532: mov     rdi, r12
0x5535: call    0xf160
0x553a: mov     rdi, r12
0x553d: mov     qword ptr [rip + 0x2537c], rax
0x5544: call    0x18d10
0x5549: cmp     byte ptr [r14], 0x2f
0x554d: je      0x6f70
0x5553: mov     rdi, rbp
0x5556: call    0x49b0
0x555b: mov     rdi, r14
0x555e: mov     r12, rax
0x5561: call    0x49b0
0x5566: lea     rdi, [r12 + rax + 2]
0x556b: call    0x1d670
0x5570: mov     r9, r14
0x5573: mov     r8, rbp
0x5576: or      rdx, 0xffffffffffffffff

```



```

0x557a: mov     r12, rax
0x557d: mov     rdi, rax
0x5580: mov     esi, 1
0x5585: xor     eax, eax
0x5587: lea     rcx, [rip + 0x1b173]
0x558e: call    0x4f10
0x5593: mov     rdi, r12
0x5596: xor     r14d, r14d
0x5599: lea     rbx, [rsp + 0x28]
0x559e: call    0xf160
0x55a3: mov     rdi, r12
0x55a6: mov     qword ptr [rip + 0x2530b], rax
0x55ad: call    0x18d10
0x55b2: mov     rdi, rbp
0x55b5: mov     ebp, 1
0x55ba: call    0x18d10
0x55bf: xor     eax, eax
0x55c1: call    0x17280
0x55c6: mov     rax, qword ptr [rip + 0x252f3]
0x55cd: xor     edi, edi
0x55cf: mov     qword ptr [rip + 0x252da], rax
0x55d6: mov     rax, qword ptr [rip + 0x252db]
0x55dd: mov     qword ptr [rip + 0x252c4], rax
0x55e4: xor     eax, eax
0x55e6: call    0x7890
0x55eb: mov     rdi, qword ptr [rip + 0x252d6]
0x55f2: call    0xbb10
0x55f7: mov     edi, 1
0x55fc: call    0x9b00
0x5601: xor     eax, eax
0x5603: call    0xa6c0
0x5608: mov     edi, 1
0x560d: mov     qword ptr [rip + 0x25280], 0
0x5618: mov     dword ptr [rip + 0x25272], 0
0x5622: mov     dword ptr [rip + 0x25264], 0
0x562c: call    0x18b90
0x5631: mov     rdi, qword ptr [rip + 0x25278]
0x5638: call    0x11100
0x563d: mov     rdi, rax
0x5640: call    0xc4b0
0x5645: lea     rax, [rsp + 0x50]
0x564a: mov     dword ptr [rsp + 0x10], r15d
0x564f: mov     qword ptr [rsp], 0
0x5657: mov     r15, rax
0x565a: cmp     dword ptr [rip + 0x2527b], 0
0x5661: jne     0x6d10
0x5667: mov     rdi, qword ptr [rip + 0x25242]
0x566e: call    0x11100

```

```

0x5673: mov     rdi, rax
0x5676: call    0xc4b0
0x567b: xor     eax, eax
0x567d: call    0x18330
0x5682: mov     edi, 0x3c
0x5687: sub     edi, dword ptr [rax]
0x5689: call    0x4aa0
0x568e: cmp     dword ptr [rip + 0x2524b], 0
0x5695: je      0x6ba2
0x569b: mov     rax, qword ptr [rip + 0x25216]
0x56a2: mov     qword ptr [rip + 0x25207], rax
0x56a9: mov     rax, qword ptr [rip + 0x25210]
0x56b0: cmp     dword ptr [rip + 0x252e5], 6
0x56b7: mov     qword ptr [rip + 0x251ea], rax
0x56be: jle     0x6d75
0x56c4: xor     eax, eax
0x56c6: call    0x16d50
0x56cb: xor     eax, eax
0x56cd: call    0xe4e0
0x56d2: mov     rdi, r14
0x56d5: call    0xc970
0x56da: xor     eax, eax
0x56dc: call    0xa820
0x56e1: xor     eax, eax
0x56e3: call    0xc8c0
0x56e8: mov     r12, rax
0x56eb: xor     eax, eax
0x56ed: call    0x96c0
0x56f2: xor     edi, edi
0x56f4: xor     eax, eax
0x56f6: call    0x7bd0
0x56fb: mov     rdi, r12
0x56fe: call    0xc970
0x5703: mov     rdi, qword ptr [rip + 0x251a6]
0x570a: mov     esi, 1
0x570f: call    0x14d90
0x5714: test    ebp, ebp
0x5716: jne     0x6d5a
0x571c: xor     eax, eax
0x571e: call    0x9640
0x5723: xor     eax, eax
0x5725: call    0xc8c0
0x572a: mov     r14, rax
0x572d: nop     dword ptr [rax]
0x5730: mov     rdi, r14
0x5733: call    0xc970
0x5738: xor     eax, eax
0x573a: call    0xc8c0

```

```

0x573f: mov     r14, rax
0x5742: xor     eax, eax
0x5744: call    0xd870
0x5749: xor     eax, eax
0x574b: call    0xd740
0x5750: xor     eax, eax
0x5752: call    0xa820
0x5757: mov     rdi, r15
0x575a: call    0xdb10
0x575f: mov     dword ptr [rip + 0x26347], 0
0x5769: jmp     0x5779
0x576b: nop     dword ptr [rax + rax]
0x5770: xor     edi, edi
0x5772: xor     eax, eax
0x5774: call    0x7d10
0x5779: mov     rdi, rbx
0x577c: call    0xb180
0x5781: test    rax, rax
0x5784: je      0x5770
0x5786: mov     r12, rax
0x5789: xor     eax, eax
0x578b: call    0xdea0
0x5790: mov     rax, qword ptr [r12]
0x5794: mov     r13, qword ptr [r12 + 0x10]
0x5799: movzx   ebp, byte ptr [rax]
0x579c: test    r13, r13
0x579f: je      0x57dc
0x57a1: cmp     byte ptr [r13 + 0x1b], 0
0x57a6: mov     rax, qword ptr [r13]
0x57aa: je      0x67c0
0x57b0: lea     rdx, [rip + 0x23869]
0x57b7: sub     rax, rdx
0x57ba: movabs  rdx, 0xea0ea0ea0ea0ea1
0x57c4: mov     rcx, rax
0x57c7: imul    rdx
0x57ca: mov     rax, rcx
0x57cd: sar     rax, 0x3f
0x57d1: sar     rdx, 1
0x57d4: mov     rbp, rdx
0x57d7: sub     rbp, rax
0x57da: not     ebp
0x57dc: lea     eax, [rbp + 0x57]
0x57df: cmp     eax, 0x5e
0x57e2: ja      0x57f5
0x57e4: lea     rcx, [rip + 0x1aaf1]
0x57eb: movsxd  rax, dword ptr [rcx + rax*4]
0x57ef: add     rax, rcx
0x57f2: jmp     rax

```

```

0x57f5: test    ebp, ebp
0x57f7: je        0x5730
0x57fd: mov     rdi, r14
0x5800: call    0xc7e0
0x5805: jmp     0x5813
0x5807: mov     esi, 3
0x580c: mov     edi, ebp
0x580e: call    0x7570
0x5813: mov     eax, dword ptr [rsp + 0x28]
0x5817: lea     edx, [rax - 1]
0x581a: mov     dword ptr [rsp + 0x28], edx
0x581e: test    eax, eax
0x5820: jne     0x5807
0x5822: jmp     0x5725
0x5827: mov     esi, 0xffffffff
0x582c: mov     rdi, qword ptr [rip + 0x2507d]
0x5833: sub     esi, ebp
0x5835: xor     ecx, ecx
0x5837: xor     edx, edx
0x5839: mov     r8d, 1
0x583f: call    0x13e30
0x5844: jmp     0x5730
0x5849: mov     esi, 0xffffffff
0x584e: jmp     0x582c
0x5850: mov     rdi, qword ptr [rip + 0x25059]
0x5857: xor     esi, esi
0x5859: call    0x14d90
0x585e: mov     rax, qword ptr [rip + 0x2504b]
0x5865: mov     esi, 1
0x586a: mov     rdi, qword ptr [rip + 0x25037]
0x5871: mov     qword ptr [rip + 0x25028], rax
0x5878: mov     qword ptr [rip + 0x25029], rax
0x587f: xor     eax, eax
0x5881: cmp     dword ptr [rip + 0x25058], 0
0x5888: sete    al
0x588b: mov     qword ptr [rip + 0x2501e], rdi
0x5892: mov     dword ptr [rip + 0x25048], eax
0x5898: call    0x14d90
0x589d: mov     rdi, qword ptr [rip + 0x2500c]
0x58a4: call    0xf570
0x58a9: mov     rdi, r14
0x58ac: call    0xc7e0
0x58b1: xor     eax, eax
0x58b3: call    0x9640
0x58b8: xor     eax, eax
0x58ba: call    0xc8c0
0x58bf: mov     rdi, qword ptr [rip + 0x24fea]
0x58c6: mov     r14, rax

```

```

0x58c9: call    0x11100
0x58ce: mov     rdi, rax
0x58d1: call    0xc4b0
0x58d6: mov     rdi, qword ptr [rip + 0x24fd3]
0x58dd: call    0xf480
0x58e2: mov     rdi, qword ptr [rip + 0x24fbf]
0x58e9: call    0xf480
0x58ee: xor     eax, eax
0x58f0: call    0xa480
0x58f5: mov     rdi, qword ptr [rip + 0x24fac]
0x58fc: call    0xf580
0x5901: xor     eax, eax
0x5903: mov     edi, 1
0x5908: mov     dword ptr [rip + 0x24346], 0
0x5912: call    0x7890
0x5917: mov     rdi, qword ptr [rip + 0x24f92]
0x591e: xor     ecx, ecx
0x5920: xor     edx, edx
0x5922: mov     r8d, 1
0x5928: mov     esi, 0xe
0x592d: call    0x13e30
0x5932: mov     rdi, qword ptr [rip + 0x24f6f]
0x5939: xor     ecx, ecx
0x593b: xor     edx, edx
0x593d: mov     r8d, 1
0x5943: mov     esi, 0xe
0x5948: call    0x13e30
0x594d: mov     rdi, qword ptr [rip + 0x24f5c]
0x5954: call    0x11010
0x5959: jmp     0x5730
0x595e: xor     eax, eax
0x5960: call    0x8960
0x5965: mov     rdi, qword ptr [rip + 0x24f44]
0x596c: call    0x11100
0x5971: mov     rdi, rax
0x5974: xor     eax, eax
0x5976: call    0x89d0
0x597b: jmp     0x5730
0x5980: mov     rdi, qword ptr [rip + 0x24f29]
0x5987: call    0xf480
0x598c: mov     rdi, qword ptr [rip + 0x24f15]
0x5993: call    0xf480
0x5998: xor     eax, eax
0x599a: call    0xa480
0x599f: mov     rdi, qword ptr [rip + 0x24f02]
0x59a6: call    0xf570
0x59ab: xor     eax, eax
0x59ad: mov     edi, 1

```

```

0x59b2: mov     dword ptr [rip + 0x2429c], 1
0x59bc: call    0x7890
0x59c1: cmp     dword ptr [rip + 0x24fd0], 0xb
0x59c8: jle     0x703f
0x59ce: mov     rdi, qword ptr [rip + 0x24edb]
0x59d5: xor     ecx, ecx
0x59d7: xor     edx, edx
0x59d9: mov     esi, 0xb
0x59de: mov     r8d, 1
0x59e4: call    0x13e30
0x59e9: mov     rdi, qword ptr [rip + 0x24eb8]
0x59f0: xor     ecx, ecx
0x59f2: xor     edx, edx
0x59f4: mov     r8d, 1
0x59fa: mov     esi, 0xb
0x59ff: call    0x13e30
0x5a04: mov     rdi, qword ptr [rip + 0x24ea5]
0x5a0b: call    0x11010
0x5a10: mov     rdi, qword ptr [rip + 0x24e91]
0x5a17: call    0x11010
0x5a1c: jmp     0x5730
0x5a21: xor     edi, edi
0x5a23: call    0xca20
0x5a28: lea     rsi, [rsp + 0x38]
0x5a2d: xor     ecx, ecx
0x5a2f: xor     edx, edx
0x5a31: mov     qword ptr [rsp + 0x38], 0
0x5a3a: lea     rdi, [rip + 0x1ad44]
0x5a41: call    0x83e0
0x5a46: mov     rax, qword ptr [rsp + 0x38]
0x5a4b: test    rax, rax
0x5a4e: je      0x5a59
0x5a50: cmp     byte ptr [rax], 0
0x5a53: jne     0x7114
0x5a59: mov     edi, 1
0x5a5e: call    0xca20
0x5a63: jmp     0x5730
0x5a68: mov     dword ptr [rsp + 0x30], 0
0x5a70: lea     rcx, [rsp + 0x30]
0x5a75: mov     rdx, qword ptr [rip + 0x24e2c]
0x5a7c: mov     esi, 0x2b
0x5a81: mov     rdi, qword ptr [rip + 0x24e28]
0x5a88: mov     r8d, 1
0x5a8e: call    0x13e30
0x5a93: jmp     0x5730
0x5a98: mov     dword ptr [rsp + 0x2c], 0
0x5aa0: lea     rcx, [rsp + 0x2c]
0x5aa5: mov     rdx, qword ptr [rip + 0x24dfc]

```

```

0x5aac: mov     esi, 0x2c
0x5ab1: mov     rdi, qword ptr [rip + 0x24df8]
0x5ab8: mov     r8d, 1
0x5abe: call    0x13e30
0x5ac3: jmp     0x5730
0x5ac8: mov     dword ptr [rsp + 0x30], 1
0x5ad0: jmp     0x5a70
0x5ad2: mov     dword ptr [rsp + 0x2c], 1
0x5ada: jmp     0x5aa0
0x5adc: mov     rdx, qword ptr [rip + 0x24dc5]
0x5ae3: mov     rdi, qword ptr [rip + 0x24dc6]
0x5aea: xor     ecx, ecx
0x5aec: mov     esi, 0x2d
0x5af1: mov     r8d, 1
0x5af7: call    0x13e30
0x5afc: mov     rdi, r14
0x5aff: call    0xc970
0x5b04: xor     eax, eax
0x5b06: call    0x9640
0x5b0b: xor     eax, eax
0x5b0d: call    0xc8c0
0x5b12: mov     rdi, qword ptr [rip + 0x24d97]
0x5b19: mov     r14, rax
0x5b1c: call    0x11100
0x5b21: mov     rdi, rax
0x5b24: call    0xc4b0
0x5b29: jmp     0x5730
0x5b2e: mov     rdx, qword ptr [rip + 0x24d73]
0x5b35: mov     rdi, qword ptr [rip + 0x24d74]
0x5b3c: xor     ecx, ecx
0x5b3e: mov     esi, 0x2e
0x5b43: mov     r8d, 1
0x5b49: call    0x13e30
0x5b4e: jmp     0x5730
0x5b53: lea     r12, [rsp + 0x58]
0x5b58: mov     rcx, qword ptr [r13 + 0x28]
0x5b5c: lea     rdx, [rip + 0x1b230]
0x5b63: mov     rsi, r12
0x5b66: lea     rdi, [rip + 0x1ac46]
0x5b6d: call    0x83e0
0x5b72: test    rax, rax
0x5b75: je      0x5730
0x5b7b: mov     rcx, qword ptr [rsp + 0x58]
0x5b80: cmp     byte ptr [rcx], 0
0x5b83: je      0x5730
0x5b89: mov     rdx, qword ptr [rip + 0x24d18]
0x5b90: mov     r8d, 1
0x5b96: mov     esi, 0x2f

```

```

0x5b9b: mov     rdi, qword ptr [rip + 0x24d0e]
0x5ba2: call    0x13e30
0x5ba7: mov     rdi, qword ptr [rsp + 0x58]
0x5bac: call    0x18d10
0x5bb1: mov     qword ptr [rsp + 0x58], 0
0x5bba: jmp     0x5730
0x5bbf: mov     r8d, dword ptr [rsp + 0x28]
0x5bc4: mov     rcx, qword ptr [rsp + 0x58]
0x5bc9: mov     esi, 0x30
0x5bce: mov     rdx, qword ptr [rip + 0x24cd3]
0x5bd5: mov     rdi, qword ptr [rip + 0x24cd4]
0x5bdc: call    0x13e30
0x5be1: jmp     0x5730
0x5be6: mov     r8d, dword ptr [rsp + 0x28]
0x5beb: mov     rcx, qword ptr [rsp + 0x58]
0x5bf0: mov     esi, 0x31
0x5bf5: mov     rdx, qword ptr [rip + 0x24cac]
0x5bfc: mov     rdi, qword ptr [rip + 0x24cad]
0x5c03: call    0x13e30
0x5c08: jmp     0x5730
0x5c0d: mov     rdx, qword ptr [rip + 0x24c94]
0x5c14: mov     rdi, qword ptr [rip + 0x24c95]
0x5c1b: xor     ecx, ecx
0x5c1d: mov     esi, 0x32
0x5c22: mov     r8d, 1
0x5c28: call    0x13e30
0x5c2d: jmp     0x5730
0x5c32: mov     rdx, qword ptr [rip + 0x24c6f]
0x5c39: mov     rdi, qword ptr [rip + 0x24c70]
0x5c40: xor     ecx, ecx
0x5c42: mov     esi, 0x33
0x5c47: mov     r8d, 1
0x5c4d: call    0x13e30
0x5c52: jmp     0x5730
0x5c57: mov     r8d, dword ptr [rsp + 0x28]
0x5c5c: mov     rdx, qword ptr [rip + 0x24c45]
0x5c63: xor     ecx, ecx
0x5c65: mov     esi, 0x1b
0x5c6a: mov     rdi, qword ptr [rip + 0x24c3f]
0x5c71: call    0x13e30
0x5c76: jmp     0x5730
0x5c7b: mov     r8d, dword ptr [rsp + 0x28]
0x5c80: mov     rdx, qword ptr [rip + 0x24c21]
0x5c87: xor     ecx, ecx
0x5c89: mov     esi, 0x1c
0x5c8e: mov     rdi, qword ptr [rip + 0x24c1b]
0x5c95: call    0x13e30
0x5c9a: jmp     0x5730

```



```

0x5c9f: cmp     dword ptr [rip + 0x23fb2], 0
0x5ca6: je      0x5850
0x5cac: test    byte ptr [rsp + 0x28], 1
0x5cb1: je      0x5730
0x5cb7: mov     rdi, qword ptr [rip + 0x24bf2]
0x5cbe: xor     esi, esi
0x5cc0: call    0x14d90
0x5cc5: mov     rax, qword ptr [rip + 0x24be4]
0x5ccc: mov     esi, 1
0x5cd1: mov     rdi, qword ptr [rip + 0x24bd0]
0x5cd8: mov     qword ptr [rip + 0x24bc1], rax
0x5cdf: mov     qword ptr [rip + 0x24bc2], rax
0x5ce6: xor     eax, eax
0x5ce8: cmp     dword ptr [rip + 0x24bf1], 0
0x5cef: sete    al
0x5cf2: mov     qword ptr [rip + 0x24bb7], rdi
0x5cf9: mov     dword ptr [rip + 0x24be1], eax
0x5cff: call    0x14d90
0x5d04: mov     rdi, r14
0x5d07: call    0xc7e0
0x5d0c: jmp     0x5b04
0x5d11: lea     r12, [rsp + 0x58]
0x5d16: mov     rcx, qword ptr [r13 + 0x28]
0x5d1a: xor     edx, edx
0x5d1c: lea     rdi, [rip + 0x1aa29]
0x5d23: mov     rsi, r12
0x5d26: call    0x83e0
0x5d2b: test    rax, rax
0x5d2e: je      0x5730
0x5d34: mov     rdi, qword ptr [rsp + 0x58]
0x5d39: cmp     byte ptr [rdi], 0
0x5d3c: je      0x5730
0x5d42: xor     eax, eax
0x5d44: call    0x19940
0x5d49: mov     r8d, 1
0x5d4f: mov     esi, 0x21
0x5d54: mov     rdx, qword ptr [rip + 0x24b4d]
0x5d5b: mov     rdi, qword ptr [rip + 0x24b4e]
0x5d62: mov     rcx, rax
0x5d65: mov     rbp, rax
0x5d68: call    0x13e30
0x5d6d: mov     rdi, qword ptr [rip + 0x24b3c]
0x5d74: call    0x11100
0x5d79: mov     rdi, rax
0x5d7c: xor     eax, eax
0x5d7e: call    0x89d0
0x5d83: mov     rdi, rbp
0x5d86: call    0x18d10

```

```

0x5d8b: mov     rdi, qword ptr [rsp + 0x58]
0x5d90: call    0x18d10
0x5d95: mov     qword ptr [rsp + 0x58], 0
0x5d9e: jmp     0x5afc
0x5da3: lea     r12, [rsp + 0x58]
0x5da8: mov     rcx, qword ptr [r13 + 0x28]
0x5dac: xor     edx, edx
0x5dae: lea     rdi, [rip + 0x1ac3b]
0x5db5: mov     rsi, r12
0x5db8: call    0x83e0
0x5dbd: test    rax, rax
0x5dc0: je      0x5730
0x5dc6: mov     rcx, qword ptr [rsp + 0x58]
0x5dcb: cmp     byte ptr [rcx], 0
0x5dce: je      0x5730
0x5dd4: mov     rdx, qword ptr [rip + 0x24acd]
0x5ddb: mov     r8d, 1
0x5de1: mov     esi, 0x24
0x5de6: jmp     0x5b9b
0x5deb: lea     r12, [rsp + 0x58]
0x5df0: mov     rcx, qword ptr [r13 + 0x28]
0x5df4: xor     edx, edx
0x5df6: lea     rdi, [rip + 0x1ac23]
0x5dfd: mov     rsi, r12
0x5e00: call    0x83e0
0x5e05: test    rax, rax
0x5e08: je      0x5730
0x5e0e: mov     rcx, qword ptr [rsp + 0x58]
0x5e13: cmp     byte ptr [rcx], 0
0x5e16: je      0x5730
0x5e1c: mov     rdx, qword ptr [rip + 0x24a85]
0x5e23: mov     r8d, 1
0x5e29: mov     esi, 0x25
0x5e2e: jmp     0x5b9b
0x5e33: mov     rdx, qword ptr [rip + 0x24a6e]
0x5e3a: mov     r8d, 1
0x5e40: mov     esi, 0x21
0x5e45: mov     rdi, qword ptr [rip + 0x24a64]
0x5e4c: mov     rcx, qword ptr [rdx + 0x18]
0x5e50: call    0x13e30
0x5e55: mov     rdi, qword ptr [rip + 0x24a54]
0x5e5c: call    0x11100
0x5e61: mov     rdi, rax
0x5e64: xor     eax, eax
0x5e66: call    0x89d0
0x5e6b: mov     rdi, r14
0x5e6e: call    0xc7e0
0x5e73: jmp     0x5b04

```

```

0x5e78: mov     rdx, qword ptr [rip + 0x24a31]
0x5e7f: mov     r8d, 1
0x5e85: mov     esi, 0x21
0x5e8a: mov     rdi, qword ptr [rip + 0x24a17]
0x5e91: mov     rcx, qword ptr [rdx + 0x18]
0x5e95: call    0x13e30
0x5e9a: mov     rdi, qword ptr [rip + 0x24a07]
0x5ea1: call    0x11100
0x5ea6: mov     rdi, rax
0x5ea9: xor     eax, eax
0x5eab: call    0x89d0
0x5eb0: jmp     0x5730
0x5eb5: mov     rax, qword ptr [rip + 0x249ec]
0x5ebc: mov     rdi, qword ptr [rax + 0x78]
0x5ec0: mov     r12, qword ptr [rax + 0x18]
0x5ec4: add     rdi, 5
0x5ec8: call    0x1d670
0x5ecd: mov     rbp, rax
0x5ed0: mov     rdi, qword ptr [rsp + 0x50]
0x5ed5: call    0x49b0
0x5eda: mov     rdi, r14
0x5edd: mov     r13, rax
0x5ee0: call    0xc7e0
0x5ee5: test    r13, r13
0x5ee8: je      0x5f1a
0x5eea: mov     r13, qword ptr [rsp + 0x50]
0x5eef: xor     eax, eax
0x5ef1: call    0xca00
0x5ef6: cmp     byte ptr [r13 + rax - 1], 0x2f
0x5efc: je      0x6f46
0x5f02: mov     r13, qword ptr [rsp + 0x50]
0x5f07: xor     eax, eax
0x5f09: call    0xca00
0x5f0e: cmp     byte ptr [r13 + rax - 1], 0x20
0x5f14: je      0x6f46
0x5f1a: mov     rdi, r12
0x5f1d: call    0x49b0
0x5f22: mov     rdi, r12
0x5f25: mov     rsi, rax
0x5f28: call    0x18530
0x5f2d: mov     r8, r12
0x5f30: lea     rcx, [rip + 0x1a809]
0x5f37: test    eax, eax
0x5f39: jne     0x5f42
0x5f3b: lea     rcx, [rip + 0x1a805]
0x5f42: or      rdx, 0xffffffffffffffff
0x5f46: mov     esi, 1
0x5f4b: mov     rdi, rbp

```

```

0x5f4e: xor     eax, eax
0x5f50: call    0x4f10
0x5f55: mov     rdi, rbp
0x5f58: call    0x49b0
0x5f5d: mov     rdi, rbp
0x5f60: mov     rsi, rax
0x5f63: call    0x184d0
0x5f68: mov     rdi, rbp
0x5f6b: call    0xd610
0x5f70: mov     rdi, rbp
0x5f73: call    0x18d10
0x5f78: xor     eax, eax
0x5f7a: call    0xc8c0
0x5f7f: mov     r14, rax
0x5f82: jmp     0x5730
0x5f87: mov     rdi, r14
0x5f8a: lea     r12, [rip + 0x1b488]
0x5f91: lea     rbp, [rip + 0x1b2fc]
0x5f98: call    0xc7e0
0x5f9d: mov     rdi, qword ptr [rip + 0x2490c]
0x5fa4: call    0x100c0
0x5fa9: jmp     0x6029
0x5fab: mov     rdx, qword ptr [rip + 0x248fe]
0x5fb2: cdqe
0x5fb4: imul    rax, rax, 0x50
0x5fb8: add     rax, qword ptr [rdx + 0x38]
0x5fbc: mov     r14, qword ptr [rax]
0x5fbf: mov     rdi, r14
0x5fc2: call    0x49b0
0x5fc7: lea     rdi, [rax + 5]
0x5fcb: call    0x1d670
0x5fd0: mov     rdi, r14
0x5fd3: mov     r13, rax
0x5fd6: call    0x49b0
0x5fdb: mov     rdi, r14
0x5fde: mov     rsi, rax
0x5fe1: call    0x18530
0x5fe6: mov     r8, r14
0x5fe9: mov     rcx, rbp
0x5fec: test    eax, eax
0x5fee: jne     0x5ff3
0x5ff0: mov     rcx, r12
0x5ff3: or      rdx, 0xffffffffffffffff
0x5ff7: mov     esi, 1
0x5ffc: mov     rdi, r13
0x5fff: xor     eax, eax
0x6001: call    0x4f10
0x6006: mov     rdi, r13

```

```

0x6009: call    0x49b0
0x600e: mov     rdi, r13
0x6011: mov     rsi, rax
0x6014: call    0x184d0
0x6019: mov     rdi, r13
0x601c: call    0xd610
0x6021: mov     rdi, r13
0x6024: call    0x18d10
0x6029: mov     rdi, qword ptr [rip + 0x24880]
0x6030: call    0x100e0
0x6035: cmp     eax, -1
0x6038: jne     0x5fab
0x603e: lea     rdi, [rip + 0x1a700]
0x6045: call    0xd610
0x604a: xor     eax, eax
0x604c: call    0xc8c0
0x6051: mov     r14, rax
0x6054: jmp     0x5730
0x6059: mov     rdi, r14
0x605c: call    0xc7e0
0x6061: mov     rdi, r15
0x6064: call    0xdb10
0x6069: mov     rbp, qword ptr [rsp + 0x50]
0x606e: mov     rdi, rbp
0x6071: call    0x49b0
0x6076: sub     eax, 1
0x6079: movsxd  rdx, eax
0x607c: jmp     0x6097
0x607e: lea     rcx, [rbp + rdx]
0x6083: sub     rdx, 1
0x6087: cmp     byte ptr [rcx], 0x20
0x608a: jne     0x609b
0x608c: mov     byte ptr [rcx], 0
0x608f: mov     rbp, qword ptr [rsp + 0x50]
0x6094: sub     eax, 1
0x6097: test    eax, eax
0x6099: jns     0x607e
0x609b: movzx   eax, byte ptr [rbp]
0x609f: cmp     al, 0x2b
0x60a1: je      0x71c7
0x60a7: jg      0x6edf
0x60ad: test    al, al
0x60af: je      0x71f6
0x60b5: cmp     al, 0x2a
0x60b7: jne     0x60c3
0x60b9: cmp     byte ptr [rbp + 1], 0
0x60bd: je      0x72fd
0x60c3: xor     eax, eax

```

```

0x60c5: lea     rsi, [rsp + 0x48]
0x60ca: mov     rdi, rbp
0x60cd: call    0x1b5b0
0x60d2: test    eax, eax
0x60d4: js      0x7258
0x60da: mov     rdi, qword ptr [rsp + 0x48]
0x60df: call    0x18620
0x60e4: test    eax, eax
0x60e6: je      0x72c0
0x60ec: xor     eax, eax
0x60ee: call    0xc8c0
0x60f3: mov     edx, 0x1c
0x60f8: xor     esi, esi
0x60fa: lea     rdi, [rip + 0x1a631]
0x6101: call    0x8070
0x6106: jmp     0x5723
0x610b: lea     r12, [rsp + 0x58]
0x6110: mov     rcx, qword ptr [r13 + 0x28]
0x6114: xor     edx, edx
0x6116: lea     rdi, [rip + 0x1a63b]
0x611d: mov     rsi, r12
0x6120: call    0x83e0
0x6125: test    rax, rax
0x6128: je      0x5730
0x612e: mov     rcx, qword ptr [rsp + 0x58]
0x6133: cmp     byte ptr [rcx], 0
0x6136: je      0x5730
0x613c: mov     rdx, qword ptr [rip + 0x24765]
0x6143: mov     r8d, 1
0x6149: mov     esi, 0x26
0x614e: jmp     0x5b9b
0x6153: xor     ecx, ecx
0x6155: xor     edx, edx
0x6157: xor     esi, esi
0x6159: xor     r13d, r13d
0x615c: lea     rdi, [rip + 0x1a603]
0x6163: lea     rbp, [rsp + 0x60]
0x6168: call    0x86c0
0x616d: mov     rdx, qword ptr [rip + 0x24734]
0x6174: mov     rdi, qword ptr [rip + 0x24735]
0x617b: xor     ecx, ecx
0x617d: mov     r8d, 1
0x6183: mov     esi, 0x27
0x6188: lea     r12, [rsp + 0x58]
0x618d: call    0x13e30
0x6192: mov     dword ptr [rsp + 0xc], 0
0x619a: xor     edi, edi
0x619c: mov     rcx, rbp

```

```

0x619f: mov     edx, 1
0x61a4: mov     rsi, r12
0x61a7: call    0x86c0
0x61ac: test    rax, rax
0x61af: je      0x6c72
0x61b5: mov     rdi, qword ptr [rsp + 0x58]
0x61ba: cmp     byte ptr [rdi], 0
0x61bd: jne     0x61d1
0x61bf: mov     rax, qword ptr [rsp]
0x61c3: test    rax, rax
0x61c6: je      0x61d1
0x61c8: cmp     byte ptr [rax], 0
0x61cb: jne     0x6b0b
0x61d1: cmp     dword ptr [rsp + 0x60], 2
0x61d6: je      0x6bf1
0x61dc: mov     qword ptr [rsp + 0x68], rdi
0x61e1: mov     rdx, qword ptr [rip + 0x246c0]
0x61e8: mov     r8d, 1
0x61ee: mov     rcx, rbp
0x61f1: mov     rdi, qword ptr [rip + 0x246b8]
0x61f8: mov     esi, 0x28
0x61fd: call    0x13e30
0x6202: cmp     dword ptr [rsp + 0x60], -1
0x6207: je      0x6d8e
0x620d: test    r13d, r13d
0x6210: jne     0x6b64
0x6216: mov     rdi, qword ptr [rsp + 0x58]
0x621b: call    0x49b0
0x6220: cmp     rax, qword ptr [rsp + 0x70]
0x6225: jbe     0x619a
0x622b: xor     eax, eax
0x622d: call    0xce70
0x6232: xor     eax, eax
0x6234: call    0xd870
0x6239: xor     eax, eax
0x623b: call    0xd740
0x6240: xor     eax, eax
0x6242: call    0xa820
0x6247: jmp     0x619a
0x624c: xor     ecx, ecx
0x624e: xor     edx, edx
0x6250: xor     esi, esi
0x6252: xor     r13d, r13d
0x6255: lea     rdi, [rip + 0x1a51e]
0x625c: lea     rbp, [rsp + 0x60]
0x6261: call    0x86c0
0x6266: mov     rdx, qword ptr [rip + 0x2463b]
0x626d: mov     rdi, qword ptr [rip + 0x2463c]

```

```

0x6274: xor     ecx, ecx
0x6276: mov     r8d, 1
0x627c: mov     esi, 0x27
0x6281: lea     r12, [rsp + 0x58]
0x6286: call    0x13e30
0x628b: mov     dword ptr [rsp + 0xc], 0
0x6293: xor     edi, edi
0x6295: mov     rcx, rbp
0x6298: mov     edx, 2
0x629d: mov     rsi, r12
0x62a0: call    0x86c0
0x62a5: test    rax, rax
0x62a8: je      0x6c72
0x62ae: mov     rdi, qword ptr [rsp + 0x58]
0x62b3: cmp     byte ptr [rdi], 0
0x62b6: jne     0x62ca
0x62b8: mov     rax, qword ptr [rsp]
0x62bc: test    rax, rax
0x62bf: je      0x62ca
0x62c1: cmp     byte ptr [rax], 0
0x62c4: jne     0x6a74
0x62ca: cmp     dword ptr [rsp + 0x60], 2
0x62cf: je      0x6bbc
0x62d5: mov     qword ptr [rsp + 0x68], rdi
0x62da: mov     rdx, qword ptr [rip + 0x245c7]
0x62e1: mov     r8d, 1
0x62e7: mov     rcx, rbp
0x62ea: mov     rdi, qword ptr [rip + 0x245bf]
0x62f1: mov     esi, 0x29
0x62f6: call    0x13e30
0x62fb: cmp     dword ptr [rsp + 0x60], -1
0x6300: je      0x6da8
0x6306: test    r13d, r13d
0x6309: jne     0x6acd
0x630f: mov     rdi, qword ptr [rsp + 0x58]
0x6314: call    0x49b0
0x6319: cmp     rax, qword ptr [rsp + 0x70]
0x631e: jbe     0x6293
0x6324: xor     eax, eax
0x6326: call    0xce70
0x632b: xor     eax, eax
0x632d: call    0xd870
0x6332: xor     eax, eax
0x6334: call    0xd740
0x6339: xor     eax, eax
0x633b: call    0xa820
0x6340: jmp     0x6293
0x6345: mov     rsi, qword ptr [rip + 0x2455c]

```



```

0x634c: mov     rdi, qword ptr [rip + 0x2455d]
0x6353: xor     eax, eax
0x6355: call    0x8a80
0x635a: jmp     0x5afc
0x635f: mov     rsi, qword ptr [rip + 0x24542]
0x6366: mov     rdi, qword ptr [rip + 0x24543]
0x636d: xor     eax, eax
0x636f: call    0x8a30
0x6374: jmp     0x5afc
0x6379: mov     rdi, qword ptr [rip + 0x24530]
0x6380: call    0xf490
0x6385: mov     rdi, rax
0x6388: mov     r12, rax
0x638b: call    0x49b0
0x6390: lea     rdi, [rax + 5]
0x6394: call    0x1d670
0x6399: mov     rbp, rax
0x639c: jmp     0x5ed0
0x63a1: mov     rdx, qword ptr [rip + 0x24500]
0x63a8: mov     rdi, qword ptr [rip + 0x24501]
0x63af: xor     ecx, ecx
0x63b1: mov     esi, 0x1f
0x63b6: mov     r8d, 1
0x63bc: call    0x13e30
0x63c1: jmp     0x5730
0x63c6: mov     rdx, qword ptr [rip + 0x244db]
0x63cd: mov     rdi, qword ptr [rip + 0x244dc]
0x63d4: xor     ecx, ecx
0x63d6: mov     esi, 0x20
0x63db: mov     r8d, 1
0x63e1: call    0x13e30
0x63e6: jmp     0x5730
0x63eb: mov     rdx, qword ptr [rip + 0x244b6]
0x63f2: mov     rdi, qword ptr [rip + 0x244b7]
0x63f9: xor     ecx, ecx
0x63fb: mov     esi, 0x19
0x6400: mov     r8d, 1
0x6406: call    0x13e30
0x640b: jmp     0x5730
0x6410: mov     rdx, qword ptr [rip + 0x24491]
0x6417: mov     rdi, qword ptr [rip + 0x24492]
0x641e: xor     ecx, ecx
0x6420: mov     esi, 3
0x6425: mov     r8d, 1
0x642b: call    0x13e30
0x6430: jmp     0x5730
0x6435: cmp     dword ptr [rip + 0x244cc], 0
0x643c: jne     0x701d

```

```

0x6442: mov     r15d, dword ptr [rsp + 0x10]
0x6447: mov     rdi, qword ptr [rip + 0x24462]
0x644e: call    0x11100
0x6453: mov     rdi, qword ptr [rip + 0x24466]
0x645a: mov     rbp, rax
0x645d: call    0xf3e0
0x6462: mov     rdi, qword ptr [rip + 0x2444f]
0x6469: call    0xf3e0
0x646e: xor     edi, edi
0x6470: call    0x9b00
0x6475: xor     eax, eax
0x6477: call    0xa6c0
0x647c: cmp     qword ptr [rip + 0x24f84], 0
0x6484: je      0x648d
0x6486: xor     eax, eax
0x6488: call    0xc820
0x648d: xor     eax, eax
0x648f: call    0xde40
0x6494: xor     eax, eax
0x6496: call    0x19580
0x649b: mov     rdi, qword ptr [rip + 0x24426]
0x64a2: call    0xbbc0
0x64a7: test    r15d, r15d
0x64aa: je      0x507e
0x64b0: mov     rdi, rbp
0x64b3: xor     r15d, r15d
0x64b6: call    0x49b0
0x64bb: mov     rsi, rbp
0x64be: mov     edi, 3
0x64c3: mov     rdx, rax
0x64c6: call    0x4790
0x64cb: jmp     0x507e
0x64d0: mov     rdi, r14
0x64d3: xor     ebp, ebp
0x64d5: call    0xc7e0
0x64da: jmp     0x64f0
0x64dc: mov     edi, 1
0x64e1: add     ebp, 1
0x64e4: call    0xdbb0
0x64e9: xor     eax, eax
0x64eb: call    0xa820
0x64f0: cmp     dword ptr [rsp + 0x28], ebp
0x64f4: jg      0x64dc
0x64f6: jmp     0x5723
0x64fb: test    byte ptr [rsp + 0x28], 1
0x6500: je      0x5730
0x6506: xor     edi, edi
0x6508: lea     rbp, [rip + 0x19f49]

```

```

0x650f: lea      r13, [rip + 0x22b0a]
0x6516: call     0x4aa0
0x651b: mov      rdi, qword ptr [rip + 0x243a6]
0x6522: call     0xbb40
0x6527: mov      rdi, qword ptr [rip + 0x2373a]
0x652e: xor      edx, edx
0x6530: xor      esi, esi
0x6532: call     0xe480
0x6537: xor      eax, eax
0x6539: call     0xa820
0x653e: mov      rdi, r14
0x6541: call     0xc970
0x6546: xor      eax, eax
0x6548: call     0xc8c0
0x654d: mov      r14, rax
0x6550: xor      eax, eax
0x6552: call     0xd870
0x6557: xor      eax, eax
0x6559: call     0xd740
0x655e: xor      eax, eax
0x6560: call     0xa820
0x6565: mov      rdi, r15
0x6568: call     0xdb10
0x656d: mov      dword ptr [rip + 0x2436d], 1
0x6577: jmp      0x658e
0x6579: nop      dword ptr [rax]
0x6580: mov      rdi, qword ptr [rip + 0x236e1]
0x6587: xor      eax, eax
0x6589: call     0x7d10
0x658e: mov      rdi, rbx
0x6591: call     0xb180
0x6596: test     rax, rax
0x6599: je       0x6580
0x659b: mov      rdx, qword ptr [rax]
0x659e: mov      rax, qword ptr [rax + 0x10]
0x65a2: movzx    r12d, byte ptr [rdx]
0x65a6: test     rax, rax
0x65a9: je       0x696f
0x65af: cmp      byte ptr [rax + 0x1b], 0
0x65b3: je       0x696f
0x65b9: mov      rax, qword ptr [rax]
0x65bc: mov      esi, 0x23
0x65c1: sub      rax, r13
0x65c4: cqo
0x65c6: idiv     rsi
0x65c9: mov      r12d, eax
0x65cc: not      r12d
0x65cf: cmp      eax, 6

```

```

0x65d2: jne      0x6e1f
0x65d8: test     byte ptr [rsp + 0x28], 1
0x65dd: jne      0x7383
0x65e3: mov      rdi, r14
0x65e6: call     0xc7e0
0x65eb: jmp      0x65fa
0x65ed: mov      esi, 3
0x65f2: mov      edi, r12d
0x65f5: call     0x7570
0x65fa: mov      eax, dword ptr [rsp + 0x28]
0x65fe: lea      edx, [rax - 1]
0x6601: mov      dword ptr [rsp + 0x28], edx
0x6605: test     eax, eax
0x6607: jne      0x65ed
0x6609: call     0xc8c0
0x660e: mov      r14, rax
0x6611: xor      eax, eax
0x6613: call     0xdea0
0x6618: jmp      0x653e
0x661d: xor      eax, eax
0x661f: call     0x96c0
0x6624: mov      rdi, qword ptr [rip + 0x24285]
0x662b: call     0x11100
0x6630: mov      rdi, rax
0x6633: call     0xc4b0
0x6638: xor      edi, edi
0x663a: xor      eax, eax
0x663c: call     0x7bd0
0x6641: jmp      0x5730
0x6646: test     byte ptr [rsp + 0x28], 1
0x664b: je       0x5730
0x6651: cmp      dword ptr [rip + 0x23600], 0
0x6658: je       0x5730
0x665e: mov      rdi, qword ptr [rip + 0x2424b]
0x6665: call     0xf480
0x666a: mov      rdi, qword ptr [rip + 0x24237]
0x6671: call     0xf480
0x6676: mov      rdx, qword ptr [rip + 0x2422b]
0x667d: mov      rdi, qword ptr [rip + 0x2422c]
0x6684: xor      ecx, ecx
0x6686: mov      r8d, 1
0x668c: mov      esi, 0x23
0x6691: jmp      0x59ff
0x6696: mov      rdi, r14
0x6699: xor      ebp, ebp
0x669b: call     0xc7e0
0x66a0: jmp      0x66b6
0x66a2: mov      edi, 2

```

```

0x66a7: add     ebp, 1
0x66aa: call    0xdbb0
0x66af: xor     eax, eax
0x66b1: call    0xa820
0x66b6: cmp     dword ptr [rsp + 0x28], ebp
0x66ba: jg      0x66a2
0x66bc: jmp     0x5723
0x66c1: mov     rdi, r14
0x66c4: call    0xc7e0
0x66c9: xor     edi, edi
0x66cb: call    0xd1a0
0x66d0: xor     eax, eax
0x66d2: call    0xc8c0
0x66d7: mov     r14, rax
0x66da: jmp     0x5730
0x66df: mov     rdx, qword ptr [rip + 0x241c2]
0x66e6: mov     rdi, qword ptr [rip + 0x241c3]
0x66ed: xor     ecx, ecx
0x66ef: mov     esi, 2
0x66f4: mov     r8d, 1
0x66fa: call    0x13e30
0x66ff: jmp     0x5730
0x6704: mov     rdx, qword ptr [rip + 0x2419d]
0x670b: mov     rdi, qword ptr [rip + 0x2419e]
0x6712: xor     ecx, ecx
0x6714: mov     esi, 0x1a
0x6719: mov     r8d, 1
0x671f: call    0x13e30
0x6724: jmp     0x5730
0x6729: xor     ebp, ebp
0x672b: jmp     0x6750
0x672d: mov     r8d, 1
0x6733: xor     ecx, ecx
0x6735: mov     esi, 4
0x673a: add     ebp, 1
0x673d: mov     rdx, qword ptr [rip + 0x24164]
0x6744: mov     rdi, qword ptr [rip + 0x24165]
0x674b: call    0x13e30
0x6750: cmp     dword ptr [rsp + 0x28], ebp
0x6754: jg      0x672d
0x6756: jmp     0x5730
0x675b: xor     ebp, ebp
0x675d: jmp     0x6782
0x675f: mov     r8d, 1
0x6765: xor     ecx, ecx
0x6767: mov     esi, 0x1d
0x676c: add     ebp, 1
0x676f: mov     rdx, qword ptr [rip + 0x24132]

```

```

0x6776: mov     rdi, qword ptr [rip + 0x24133]
0x677d: call    0x13e30
0x6782: cmp     dword ptr [rsp + 0x28], ebp
0x6786: jg      0x675f
0x6788: jmp     0x5730
0x678d: xor     ebp, ebp
0x678f: jmp     0x67b4
0x6791: mov     r8d, 1
0x6797: xor     ecx, ecx
0x6799: mov     esi, 0x1e
0x679e: add     ebp, 1
0x67a1: mov     rdx, qword ptr [rip + 0x24100]
0x67a8: mov     rdi, qword ptr [rip + 0x24101]
0x67af: call    0x13e30
0x67b4: cmp     dword ptr [rsp + 0x28], ebp
0x67b8: jg      0x6791
0x67ba: jmp     0x5730
0x67bf: nop
0x67c0: test    rax, rax
0x67c3: je      0x57dc
0x67c9: mov     rdi, qword ptr [rip + 0x240e0]
0x67d0: call    0xf480
0x67d5: mov     rdi, qword ptr [rip + 0x240cc]
0x67dc: call    0xf480
0x67e1: cmp     qword ptr [r13 + 8], 0
0x67e6: je      0x70b7
0x67ec: lea     rbp, [rsp + 0x60]
0x67f1: mov     rcx, qword ptr [rip + 0x240b0]
0x67f8: xor     eax, eax
0x67fa: mov     rdi, r13
0x67fd: mov     rdx, qword ptr [rip + 0x240ac]
0x6804: mov     rsi, rbp
0x6807: mov     qword ptr [rsp + 0x60], 0
0x6810: call    0x8b50
0x6815: test    eax, eax
0x6817: je      0x6c1f
0x681d: jle     0x5730
0x6823: mov     rdi, qword ptr [r13]
0x6827: mov     dword ptr [rsp + 0xc], eax
0x682b: call    0x49b0
0x6830: mov     rbp, qword ptr [rsp + 0x60]
0x6835: mov     r12, rax
0x6838: mov     rdi, rbp
0x683b: call    0x49b0
0x6840: lea     rdi, [r12 + rax + 0x21]
0x6845: call    0x1d670
0x684a: mov     r8, qword ptr [r13]
0x684e: mov     r9, rbp

```

```

0x6851: or      rdx, 0xffffffffffffffff
0x6855: lea     rcx, [rip + 0x19cd9]
0x685c: mov     r12, rax
0x685f: mov     rdi, rax
0x6862: mov     esi, 1
0x6867: xor     eax, eax
0x6869: call    0x4f10
0x686e: mov     edx, 1
0x6873: mov     esi, 1
0x6878: mov     rdi, r12
0x687b: call    0xe480
0x6880: xor     eax, eax
0x6882: call    0xa820
0x6887: mov     rdi, r12
0x688a: call    0x18d10
0x688f: cmp     byte ptr [r13 + 0x1a], 0
0x6894: jne     0x6dc9
0x689a: mov     rdi, rbp
0x689d: call    0x185d0
0x68a2: test    eax, eax
0x68a4: je      0x6dbc
0x68aa: movsx   esi, byte ptr [r13 + 0x1a]
0x68af: mov     rdi, rbp
0x68b2: call    0x193f0
0x68b7: mov     rdi, rbp
0x68ba: mov     r12d, eax
0x68bd: call    0x18d10
0x68c2: cmp     byte ptr [r13 + 0x1a], 0
0x68c7: je      0x6ccf
0x68cd: mov     eax, r12d
0x68d0: and     eax, 0x7f
0x68d3: add     eax, 1
0x68d6: sub     al, 1
0x68d8: jg      0x6f1d
0x68de: test    r12b, 0x7f
0x68e2: jne     0x6955
0x68e4: and     r12d, 0xff00
0x68eb: jne     0x719d
0x68f1: mov     rdi, qword ptr [r13 + 0x10]
0x68f5: test    rdi, rdi
0x68f8: je      0x6943
0x68fa: xor     eax, eax
0x68fc: call    0x19940
0x6901: mov     r8d, 1
0x6907: mov     esi, 0x21
0x690c: mov     rdx, qword ptr [rip + 0x23f95]
0x6913: mov     rdi, qword ptr [rip + 0x23f96]
0x691a: mov     rcx, rax

```

```

0x691d: mov     r12, rax
0x6920: call    0x13e30
0x6925: mov     rdi, qword ptr [rip + 0x23f84]
0x692c: call    0x11100
0x6931: mov     rdi, rax
0x6934: xor     eax, eax
0x6936: call    0x89d0
0x693b: mov     rdi, r12
0x693e: call    0x18d10
0x6943: cmp     dword ptr [rsp + 0xc], 2
0x6948: je      0x71b4
0x694e: cmp     dword ptr [rsp + 0xc], 3
0x6953: je      0x695c
0x6955: xor     ebp, ebp
0x6957: jmp     0x565a
0x695c: mov     rdi, qword ptr [rip + 0x23f45]
0x6963: xor     ebp, ebp
0x6965: call    0x110b0
0x696a: jmp     0x565a
0x696f: lea     eax, [r12 + 0x22]
0x6974: cmp     eax, 0x29
0x6977: ja      0x65e3
0x697d: movsxd  rax, dword ptr [rbp + rax*4]
0x6982: add     rax, rbp
0x6985: jmp     rax
0x6988: -----

```

```

0x6a72: -----
0x6a73: -----
0x6a74: call    0x18d10
0x6a79: mov     rdi, qword ptr [rsp]
0x6a7d: call    0x4e70
0x6a82: mov     rdi, rax
0x6a85: mov     qword ptr [rsp + 0x58], rax
0x6a8a: call    0xd610
0x6a8f: mov     rax, qword ptr [rsp + 0x58]
0x6a94: mov     rcx, rbp
0x6a97: mov     rdx, qword ptr [rip + 0x23e0a]
0x6a9e: mov     rdi, qword ptr [rip + 0x23e0b]
0x6aa5: mov     r8d, 1
0x6aab: mov     esi, 0x29
0x6ab0: mov     qword ptr [rsp + 0x68], rax
0x6ab5: call    0x13e30
0x6aba: cmp     dword ptr [rsp + 0x60], -1
0x6abf: je      0x6da2
0x6ac5: mov     dword ptr [rsp + 0xc], 0

```



```

0x6acd: mov     rdi, qword ptr [rsp]
0x6ad1: mov     r13d, 1
0x6ad7: call    0x49b0
0x6adc: cmp     qword ptr [rsp + 0x70], rax
0x6ae1: jae     0x630f
0x6ae7: xor     edi, edi
0x6ae9: xor     r13d, r13d
0x6aec: call    0xd1a0
0x6af1: mov     rdi, qword ptr [rsp + 0x58]
0x6af6: call    0x49b0
0x6afb: cmp     rax, qword ptr [rsp + 0x70]
0x6b00: jbe     0x632b
0x6b06: jmp     0x6324
0x6b0b: call    0x18d10
0x6b10: mov     rdi, qword ptr [rsp]
0x6b14: call    0x4e70
0x6b19: mov     rdi, rax
0x6b1c: mov     qword ptr [rsp + 0x58], rax
0x6b21: call    0xd610
0x6b26: mov     rax, qword ptr [rsp + 0x58]
0x6b2b: mov     rcx, rbp
0x6b2e: mov     rdx, qword ptr [rip + 0x23d73]
0x6b35: mov     rdi, qword ptr [rip + 0x23d74]
0x6b3c: mov     r8d, 1
0x6b42: mov     esi, 0x28
0x6b47: mov     qword ptr [rsp + 0x68], rax
0x6b4c: call    0x13e30
0x6b51: cmp     dword ptr [rsp + 0x60], -1
0x6b56: je      0x6d88
0x6b5c: mov     dword ptr [rsp + 0xc], 0
0x6b64: mov     rdi, qword ptr [rsp]
0x6b68: mov     r13d, 1
0x6b6e: call    0x49b0
0x6b73: cmp     qword ptr [rsp + 0x70], rax
0x6b78: jae     0x6216
0x6b7e: xor     edi, edi
0x6b80: xor     r13d, r13d
0x6b83: call    0xd1a0
0x6b88: mov     rdi, qword ptr [rsp + 0x58]
0x6b8d: call    0x49b0
0x6b92: cmp     rax, qword ptr [rsp + 0x70]
0x6b97: jbe     0x6232
0x6b9d: jmp     0x622b
0x6ba2: mov     rax, qword ptr [rip + 0x23d17]
0x6ba9: mov     qword ptr [rip + 0x23d00], rax
0x6bb0: mov     rax, qword ptr [rip + 0x23d01]
0x6bb7: jmp     0x56b0
0x6bbc: cmp     dword ptr [rsp + 0xc], 0

```

```

0x6bc1: je      0x62d5
0x6bc7: xor      eax, eax
0x6bc9: call     0xa5a0
0x6bce: mov      rdi, qword ptr [rip + 0x23cdb]
0x6bd5: mov      esi, 1
0x6bda: call     0xf660
0x6bdf: mov      dword ptr [rsp + 0xc], 0
0x6be7: mov      rdi, qword ptr [rsp + 0x58]
0x6bec: jmp      0x62d5
0x6bf1: cmp      dword ptr [rsp + 0xc], 0
0x6bf6: je      0x61dc
0x6bfc: mov      rdi, qword ptr [rip + 0x23cad]
0x6c03: mov      esi, 1
0x6c08: call     0xf660
0x6c0d: mov      dword ptr [rsp + 0xc], 0
0x6c15: mov      rdi, qword ptr [rsp + 0x58]
0x6c1a: jmp      0x61dc
0x6c1f: mov      rdi, qword ptr [r12]
0x6c23: call     0x49b0
0x6c28: lea      rdi, [rax + 0x51]
0x6c2c: call     0x1d670
0x6c31: mov      r9, qword ptr [r13 + 0x20]
0x6c35: mov      r8, qword ptr [r13]
0x6c39: lea      rcx, [rip + 0x19d80]
0x6c40: mov      rbp, rax
0x6c43: or       rdx, 0xffffffffffffffff
0x6c47: mov      rdi, rax
0x6c4a: mov      esi, 1
0x6c4f: xor      eax, eax
0x6c51: call     0x4f10
0x6c56: mov      rdi, rbp
0x6c59: mov      edx, 0x1c
0x6c5e: xor      esi, esi
0x6c60: call     0x8070
0x6c65: mov      rdi, rbp
0x6c68: call     0x18d10
0x6c6d: jmp      0x5730
0x6c72: mov      rax, qword ptr [rsp]
0x6c76: test     rax, rax
0x6c79: je      0x6c83
0x6c7b: mov      rdi, rax
0x6c7e: call     0x18d10
0x6c83: mov      rdi, qword ptr [rsp + 0x58]
0x6c88: test     rdi, rdi
0x6c8b: je      0x5730
0x6c91: call     0x4e70
0x6c96: mov      rdx, qword ptr [rip + 0x23c0b]
0x6c9d: xor      ecx, ecx

```

```

0x6c9f: mov     rdi, qword ptr [rip + 0x23c0a]
0x6ca6: mov     r8d, 1
0x6cac: mov     esi, 0x2a
0x6cb1: mov     qword ptr [rsp], rax
0x6cb5: call    0x13e30
0x6cba: xor     ecx, ecx
0x6cbc: mov     edx, 3
0x6cc1: xor     esi, esi
0x6cc3: xor     edi, edi
0x6cc5: call    0x86c0
0x6cca: jmp     0x5730
0x6ccf: cmp     byte ptr [r13 + 0x18], 0
0x6cd4: jne     0x6f35
0x6cda: xor     eax, eax
0x6cdc: call    0xa480
0x6ce1: cmp     byte ptr [r13 + 0x19], 0
0x6ce6: je      0x6cf2
0x6ce8: mov     dword ptr [rip + 0x23bea], 1
0x6cf2: test    r12b, 0x7f
0x6cf6: jne     0x6955
0x6cfc: and     r12d, 0xff00
0x6d03: je      0x68f1
0x6d09: xor     ebp, ebp
0x6d0b: jmp     0x565a
0x6d10: xor     edi, edi
0x6d12: call    0x4aa0
0x6d17: mov     edx, 0x21
0x6d1c: mov     rcx, qword ptr [rip + 0x23b1d]
0x6d23: mov     esi, 1
0x6d28: lea     rdi, [rip + 0x19c49]
0x6d2f: call    0x4e40
0x6d34: mov     eax, dword ptr [rip + 0x23c62]
0x6d3a: xor     esi, esi
0x6d3c: lea     edi, [rax - 1]
0x6d3f: call    0xa4b0
0x6d44: xor     edi, edi
0x6d46: call    0xb180
0x6d4b: mov     dword ptr [rip + 0x23b87], 0
0x6d55: jmp     0x5667
0x6d5a: mov     rdi, qword ptr [rip + 0x23b4f]
0x6d61: call    0x11100
0x6d66: mov     rdi, rax
0x6d69: xor     eax, eax
0x6d6b: call    0x89d0
0x6d70: jmp     0x571c
0x6d75: xor     eax, eax
0x6d77: call    0xa6c0
0x6d7c: xor     eax, eax

```

```

0x6d7e: call    0xa300
0x6d83: jmp     0x56c4
0x6d88: mov     r13d, 1
0x6d8e: xor     eax, eax
0x6d90: call    0xa5a0
0x6d95: mov     dword ptr [rsp + 0xc], 1
0x6d9d: jmp     0x619a
0x6da2: mov     r13d, 1
0x6da8: xor     eax, eax
0x6daa: call    0xa5a0
0x6daf: mov     dword ptr [rsp + 0xc], 1
0x6db7: jmp     0x6293
0x6dbc: mov     rdi, rbp
0x6dbf: call    0xc4b0
0x6dc4: jmp     0x68aa
0x6dc9: mov     rdi, qword ptr [r13]
0x6dcd: call    0x49b0
0x6dd2: lea     rdi, [rax + 0x41]
0x6dd6: call    0x1d670
0x6ddb: mov     r9, qword ptr [r13]
0x6ddf: lea     r8, [rip + 0x19921]
0x6de6: lea     rcx, [rip + 0x19bb3]
0x6ded: mov     r12, rax
0x6df0: or      rdx, 0xffffffffffffffff
0x6df4: mov     esi, 1
0x6df9: mov     rdi, rax
0x6dfc: xor     eax, eax
0x6dfe: call    0x4f10
0x6e03: mov     rdi, r12
0x6e06: call    0xdb60
0x6e0b: xor     eax, eax
0x6e0d: call    0xa820
0x6e12: mov     rdi, r12
0x6e15: call    0x18d10
0x6e1a: jmp     0x689a
0x6e1f: cmp     r12d, -0x3d
0x6e23: jne     0x696f
0x6e29: mov     rax, qword ptr [rsp + 0x50]
0x6e2e: cmp     byte ptr [rax], 0
0x6e31: je      0x6611
0x6e37: mov     rdi, r14
0x6e3a: call    0xc7e0
0x6e3f: mov     rdi, qword ptr [rsp + 0x50]
0x6e44: xor     eax, eax
0x6e46: lea     rsi, [rsp + 0x48]
0x6e4b: call    0x1b5b0
0x6e50: test    eax, eax
0x6e52: js      0x7287

```

```

0x6e58: mov     rdi, qword ptr [rip + 0x23a69]
0x6e5f: call    0xbb40
0x6e64: xor     edi, edi
0x6e66: call    0xd1a0
0x6e6b: mov     rdi, qword ptr [rsp + 0x48]
0x6e70: call    0xd610
0x6e75: mov     rdi, qword ptr [rsp + 0x48]
0x6e7a: call    0xc4b0
0x6e7f: mov     rdi, qword ptr [rsp + 0x48]
0x6e84: xor     esi, esi
0x6e86: call    0x193f0
0x6e8b: mov     rdi, qword ptr [rip + 0x23a36]
0x6e92: call    0xbb10
0x6e97: xor     edi, edi
0x6e99: call    0xdbb0
0x6e9e: mov     rdi, qword ptr [rip + 0x22dc3]
0x6ea5: xor     edx, edx
0x6ea7: xor     esi, esi
0x6ea9: call    0xe480
0x6eae: xor     edi, edi
0x6eb0: call    0xd1a0
0x6eb5: xor     eax, eax
0x6eb7: call    0xc8c0
0x6ebc: mov     rdi, qword ptr [rip + 0x239ed]
0x6ec3: mov     r14, rax
0x6ec6: call    0x11100
0x6ecb: mov     rdi, rax
0x6ece: call    0xc4b0
0x6ed3: xor     eax, eax
0x6ed5: call    0xa820
0x6eda: jmp     0x6611
0x6edf: cmp     al, 0x2d
0x6ee1: jne     0x60c3
0x6ee7: cmp     byte ptr [rbp + 1], 0
0x6eeb: mov     rdx, qword ptr [rip + 0x239b6]
0x6ef2: mov     rdi, qword ptr [rip + 0x239b7]
0x6ef9: jne     0x726e
0x6eff: mov     r8d, 1
0x6f05: xor     ecx, ecx
0x6f07: mov     esi, 6
0x6f0c: call    0x13e30
0x6f11: xor     edi, edi
0x6f13: call    0xd1a0
0x6f18: jmp     0x5723
0x6f1d: mov     edx, 0xc
0x6f22: xor     esi, esi
0x6f24: lea     rdi, [rip + 0x197e9]
0x6f2b: call    0x8070

```

```

0x6f30: jmp      0x6cf2
0x6f35: mov      rdi, qword ptr [rip + 0x2398c]
0x6f3c: call     0xbb10
0x6f41: jmp      0x6cda
0x6f46: mov      rdi, r12
0x6f49: call     0x49b0
0x6f4e: mov      rdi, r12
0x6f51: mov      rsi, rax
0x6f54: call     0x18530
0x6f59: test     eax, eax
0x6f5b: je       0x70a8
0x6f61: mov      r8, r12
0x6f64: lea      rcx, [rip + 0x197d6]
0x6f6b: jmp      0x5f42
0x6f70: mov      rdi, r14
0x6f73: call     0x1d7a0
0x6f78: mov      r12, rax
0x6f7b: jmp      0x5593
0x6f80: xor      esi, esi
0x6f82: lea      rdi, [rip + 0x19717]
0x6f89: call     0x17e50
0x6f8e: mov      dword ptr [rsp], eax
0x6f91: jmp      0x53b4
0x6f96: mov      rdi, qword ptr [rbx + rax*8]
0x6f9a: lea      edx, [rax + 1]
0x6f9d: mov      dword ptr [rip + 0x238a5], edx
0x6fa3: call     0x1d7a0
0x6fa8: mov      r13, rax
0x6fab: movsxd   rax, dword ptr [rip + 0x23896]
0x6fb2: cmp      eax, ebp
0x6fb4: jge      0x5124
0x6fba: mov      rdi, qword ptr [rbx + rax*8]
0x6fbe: lea      edx, [rax + 1]
0x6fc1: mov      dword ptr [rip + 0x23881], edx
0x6fc7: call     0x1d7a0
0x6fcc: cmp      dword ptr [rip + 0x23876], ebp
0x6fd2: mov      r14, rax
0x6fd5: jge      0x5127
0x6fdb: mov      rcx, qword ptr [rip + 0x23906]
0x6fe2: mov      rdi, qword ptr [rip + 0x23897]
0x6fe9: lea      rdx, [rip + 0x198f8]
0x6ff0: xor      eax, eax
0x6ff2: mov      esi, 1
0x6ff7: call     0x4e60
0x6ffc: jmp      0x5127
0x7001: mov      rdi, r13
0x7004: call     0x1d7a0
0x7009: mov      r12, rax

```

```

0x700c: jmp      0x5532
0x7011: lea       rbx, [rip + 0x19543]
0x7018: jmp      0x529c
0x701d: mov      rdi, qword ptr [rip + 0x238ac]
0x7024: xor      edx, edx
0x7026: lea      rsi, [rip + 0x19505]
0x702d: call     0x8070
0x7032: cmp      al, 0x79
0x7034: jne      0x5730
0x703a: jmp      0x6442
0x703f: mov      edi, 1
0x7044: xor      eax, eax
0x7046: call     0x7bd0
0x704b: jmp      0x59ce
0x7050: mov      esi, 1
0x7055: lea      rdi, [rip + 0x19664]
0x705c: call     0x17e50
0x7061: jmp      0x53df
0x7066: mov      rcx, qword ptr [rip + 0x2387b]
0x706d: mov      rdi, qword ptr [rip + 0x2380c]
0x7074: xor      eax, eax
0x7076: mov      r8d, 0x800
0x707c: lea      rdx, [rip + 0x19895]
0x7083: mov      esi, 1
0x7088: call     0x4e60
0x708d: jmp      0x54a3
0x7092: xor      esi, esi
0x7094: lea      rdi, [rip + 0x195fa]
0x709b: call     0x17e50
0x70a0: mov      r12d, eax
0x70a3: jmp      0x53a3
0x70a8: mov      r8, r12
0x70ab: lea      rcx, [rip + 0x19696]
0x70b2: jmp      0x5f42
0x70b7: mov      rdi, qword ptr [r13 + 0x10]
0x70bb: test     rdi, rdi
0x70be: je       0x6955
0x70c4: xor      eax, eax
0x70c6: call     0x19940
0x70cb: mov      r8d, 1
0x70d1: mov      esi, 0x21
0x70d6: mov      rdx, qword ptr [rip + 0x237cb]
0x70dd: mov      rdi, qword ptr [rip + 0x237cc]
0x70e4: mov      rcx, rax
0x70e7: mov      rbp, rax
0x70ea: call     0x13e30
0x70ef: mov      rdi, qword ptr [rip + 0x237ba]
0x70f6: call     0x11100

```

```

0x70fb: mov     rdi, rax
0x70fe: xor     eax, eax
0x7100: call    0x89d0
0x7105: mov     rdi, rbp
0x7108: xor     ebp, ebp
0x710a: call    0x18d10
0x710f: jmp     0x565a
0x7114: mov     qword ptr [rsp + 0x40], 0
0x711d: lea     r12, [rsp + 0x40]
0x7122: lea     rbp, [rip + 0x1966f]
0x7129: jmp     0x714f
0x712b: lea     rdi, [rip + 0x22ade]
0x7132: call    0xdb60
0x7137: xor     eax, eax
0x7139: call    0xa5a0
0x713e: xor     eax, eax
0x7140: call    0xa820
0x7145: mov     edi, 2
0x714a: call    0x4ea0
0x714f: mov     rsi, r12
0x7152: xor     ecx, ecx
0x7154: xor     edx, edx
0x7156: mov     rdi, rbp
0x7159: call    0x83e0
0x715e: xor     eax, eax
0x7160: call    0xa820
0x7165: mov     rsi, qword ptr [rsp + 0x40]
0x716a: test    rsi, rsi
0x716d: je      0x712b
0x716f: mov     rdi, qword ptr [rsp + 0x38]
0x7174: call    0x4b30
0x7179: test    eax, eax
0x717b: jne     0x712b
0x717d: xor     eax, eax
0x717f: call    0xa820
0x7184: mov     rdi, qword ptr [rsp + 0x38]
0x7189: call    0x18d10
0x718e: mov     rdi, qword ptr [rsp + 0x40]
0x7193: call    0x18d10
0x7198: jmp     0x5a59
0x719d: xor     eax, eax
0x719f: xor     ebp, ebp
0x71a1: call    0xa5a0
0x71a6: mov     rdi, qword ptr [r13]
0x71aa: call    0x195e0
0x71af: jmp     0x565a
0x71b4: mov     rdi, qword ptr [rip + 0x236f5]
0x71bb: xor     ebp, ebp

```



```

0x71bd: call    0x110b0
0x71c2: jmp     0x565a
0x71c7: cmp     byte ptr [rbp + 1], 0
0x71cb: mov     rdx, qword ptr [rip + 0x236d6]
0x71d2: mov     rdi, qword ptr [rip + 0x236d7]
0x71d9: jne     0x72a7
0x71df: mov     r8d, 1
0x71e5: xor     ecx, ecx
0x71e7: mov     esi, 5
0x71ec: call    0x13e30
0x71f1: jmp     0x6f11
0x71f6: mov     rcx, qword ptr [rip + 0x236cb]
0x71fd: mov     rdx, qword ptr [rip + 0x236a4]
0x7204: mov     esi, 1
0x7209: mov     r8d, 1
0x720f: mov     rdi, qword ptr [rip + 0x2369a]
0x7216: call    0x13e30
0x721b: mov     rdi, qword ptr [rip + 0x2368e]
0x7222: mov     ebp, eax
0x7224: call    0x11100
0x7229: mov     rdi, rax
0x722c: call    0xc4b0
0x7231: xor     edi, edi
0x7233: call    0xd1a0
0x7238: xor     eax, eax
0x723a: call    0x9640
0x723f: xor     eax, eax
0x7241: call    0xc8c0
0x7246: mov     r14, rax
0x7249: test    ebp, ebp
0x724b: je      0x5730
0x7251: xor     ebp, ebp
0x7253: jmp     0x565a
0x7258: mov     rdi, qword ptr [rsp + 0x48]
0x725d: mov     edx, 0x1c
0x7262: xor     esi, esi
0x7264: call    0x8070
0x7269: jmp     0x5723
0x726e: lea     rcx, [rbp + 1]
0x7272: mov     r8d, 1
0x7278: mov     esi, 0x25
0x727d: call    0x13e30
0x7282: jmp     0x6f11
0x7287: mov     rdi, qword ptr [rsp + 0x48]
0x728c: mov     edx, 0x1c
0x7291: xor     esi, esi
0x7293: call    0x8070
0x7298: xor     eax, eax

```

```

0x729a: call    0xc8c0
0x729f: mov     r14, rax
0x72a2: jmp     0x6611
0x72a7: lea     rcx, [rbp + 1]
0x72ab: mov     r8d, 1
0x72b1: mov     esi, 0x24
0x72b6: call    0x13e30
0x72bb: jmp     0x6f11
0x72c0: mov     rdi, qword ptr [rsp + 0x48]
0x72c5: call    0x185d0
0x72ca: mov     ebp, eax
0x72cc: test    eax, eax
0x72ce: je      0x73fa
0x72d4: xor     edi, edi
0x72d6: call    0xd1a0
0x72db: mov     rdi, qword ptr [rsp + 0x48]
0x72e0: call    0xd610
0x72e5: mov     rdi, qword ptr [rsp + 0x48]
0x72ea: mov     esi, ebp
0x72ec: call    0x193f0
0x72f1: xor     edi, edi
0x72f3: call    0xdbb0
0x72f8: jmp     0x6f11
0x72fd: mov     rdx, qword ptr [rip + 0x235a4]
0x7304: mov     rdi, qword ptr [rip + 0x235a5]
0x730b: xor     ecx, ecx
0x730d: mov     esi, 7
0x7312: mov     r8d, 1
0x7318: call    0x13e30
0x731d: xor     edi, edi
0x731f: call    0xd1a0
0x7324: jmp     0x5723
0x7329: mov     rdi, qword ptr [rip + 0x23580]
0x7330: mov     r15d, dword ptr [rsp + 0x10]
0x7335: call    0xf480
0x733a: mov     rdi, qword ptr [rip + 0x23567]
0x7341: call    0xf480
0x7346: xor     eax, eax
0x7348: call    0xa480
0x734d: xor     eax, eax
0x734f: call    0xe4e0
0x7354: xor     eax, eax
0x7356: call    0xa820
0x735b: xor     eax, eax
0x735d: call    0x18330
0x7362: mov     r8, rax
0x7365: mov     eax, 0x3c
0x736a: sub     eax, dword ptr [r8]

```

```

0x736d: mov     edi, eax
0x736f: call    0x4aa0
0x7374: mov     dword ptr [rip + 0x23566], 0
0x737e: jmp     0x6447
0x7383: mov     rdi, r14
0x7386: xor     ebp, ebp
0x7388: call    0xc7e0
0x738d: xor     eax, eax
0x738f: call    0xc8c0
0x7394: mov     rdi, qword ptr [rip + 0x23515]
0x739b: mov     r14, rax
0x739e: call    0xf480
0x73a3: mov     rdi, qword ptr [rip + 0x234fe]
0x73aa: call    0xf480
0x73af: xor     eax, eax
0x73b1: call    0xa480
0x73b6: xor     eax, eax
0x73b8: call    0xe4e0
0x73bd: xor     eax, eax
0x73bf: call    0xa820
0x73c4: xor     eax, eax
0x73c6: call    0x18330
0x73cb: mov     r8, rax
0x73ce: mov     eax, 0x3c
0x73d3: sub     eax, dword ptr [r8]
0x73d6: mov     edi, eax
0x73d8: call    0x4aa0
0x73dd: mov     dword ptr [rip + 0x234fd], 0
0x73e7: jmp     0x565a
0x73ec: lea     rdi, [rip + 0x1955d]
0x73f3: xor     eax, eax
0x73f5: call    0x8b00
0x73fa: mov     rdi, qword ptr [rsp + 0x48]
0x73ff: xor     ebp, ebp
0x7401: call    0xc4b0
0x7406: xor     edi, edi
0x7408: call    0xd1a0
0x740d: mov     rdi, qword ptr [rsp + 0x48]
0x7412: call    0xd610
0x7417: mov     rdi, qword ptr [rsp + 0x48]
0x741c: xor     esi, esi
0x741e: call    0x193f0
0x7423: xor     edi, edi
0x7425: call    0xdbb0
0x742a: xor     edi, edi
0x742c: call    0xd1a0
0x7431: mov     rdi, qword ptr [rip + 0x23478]
0x7438: call    0xf480

```

```

0x743d: mov     rdi, qword ptr [rip + 0x23464]
0x7444: call    0xf480
0x7449: xor     eax, eax
0x744b: call    0xa480
0x7450: mov     rdi, qword ptr [rip + 0x23471]
0x7457: call    0xbb10
0x745c: xor     eax, eax
0x745e: mov     dword ptr [rip + 0x23474], 1
0x7468: call    0xc8c0
0x746d: mov     r14, rax
0x7470: jmp     0x565a
0x7475: call    0x49d0
0x747a: nop     word ptr [rax + rax]
0x7480: endbr64
0x7484: xor     ebp, ebp
0x7486: mov     r9, rdx
0x7489: pop     rsi
0x748a: mov     rdx, rsp
0x748d: and     rsp, 0xfffffffffffffff0
0x7491: push    rax
0x7492: push    rsp
0x7493: xor     r8d, r8d
0x7496: xor     ecx, ecx
0x7498: lea     rdi, [rip - 0x256f]
0x749f: call    qword ptr [rip + 0x21b23]
0x74a5: hlt
0x74a6: nop     word ptr cs:[rax + rax]
0x74b0: lea     rdi, [rip + 0x23379]
0x74b7: lea     rax, [rip + 0x23372]
0x74be: cmp     rax, rdi
0x74c1: je      0x74d8
0x74c3: mov     rax, qword ptr [rip + 0x21b06]
0x74ca: test    rax, rax
0x74cd: je      0x74d8
0x74cf: jmp     rax
0x74d1: -----

0x74d6: -----
0x74d7: -----
0x74d8: ret
0x74d9: nop     dword ptr [rax]
0x74e0: lea     rdi, [rip + 0x23349]
0x74e7: lea     rsi, [rip + 0x23342]
0x74ee: sub     rsi, rdi
0x74f1: mov     rax, rsi
0x74f4: shr     rsi, 0x3f

```

```

0x74f8: sar      rax, 3
0x74fc: add      rsi, rax
0x74ff: sar      rsi, 1
0x7502: je       0x7518
0x7504: mov      rax, qword ptr [rip + 0x21ae5]
0x750b: test     rax, rax
0x750e: je       0x7518
0x7510: jmp      rax
0x7512: -----

```

```

0x7516: -----
0x7517: -----
0x7518: ret
0x7519: nop      dword ptr [rax]
0x7520: endbr64
0x7524: cmp      byte ptr [rip + 0x2335d], 0
0x752b: jne      0x7558
0x752d: push     rbp
0x752e: cmp      qword ptr [rip + 0x21ac2], 0
0x7536: mov      rbp, rsp
0x7539: je       0x7547
0x753b: mov      rdi, qword ptr [rip + 0x21ac6]
0x7542: call     0x47b0
0x7547: call     0x74b0
0x754c: mov      byte ptr [rip + 0x23335], 1
0x7553: pop      rbp
0x7554: ret
0x7555: nop      dword ptr [rax]
0x7558: ret
0x7559: nop      dword ptr [rax]
0x7560: endbr64
0x7564: jmp      0x74e0
0x7569: nop      dword ptr [rax]
0x7570: push     r12
0x7572: lea      eax, [rdi + 0x42]
0x7575: mov      r12d, esi
0x7578: sub      rsp, 0x10
0x757c: cmp      eax, 0x17
0x757f: ja       0x77e0
0x7585: lea      rdx, [rip + 0x18a78]
0x758c: movsxd   rax, dword ptr [rdx + rax*4]
0x7590: add      rax, rdx
0x7593: jmp      rax
0x7596: -----

```

```

0x75ab: -----
0x75ac: -----
0x75ad: add      rsp, 0x10
0x75b1: mov      eax, r12d
0x75b4: pop      r12
0x75b6: ret
0x75b7: nop      word ptr [rax + rax]
0x75c0: xor      eax, eax
0x75c2: mov      r12d, 1
0x75c8: call     0xcb30
0x75cd: jmp      0x75ad
0x75cf: nop
0x75d0: xor      eax, eax
0x75d2: mov      r12d, 1
0x75d8: call     0xcb60
0x75dd: jmp      0x75ad
0x75df: nop
0x75e0: xor      eax, eax
0x75e2: mov      r12d, 1
0x75e8: call     0xcc30
0x75ed: jmp      0x75ad
0x75ef: nop
0x75f0: xor      eax, eax
0x75f2: mov      r12d, 1
0x75f8: call     0xccf0
0x75fd: jmp      0x75ad
0x75ff: nop
0x7600: xor      eax, eax
0x7602: mov      r12d, 1
0x7608: call     0xcd10
0x760d: jmp      0x75ad
0x760f: nop
0x7610: and      r12d, 1
0x7614: jne      0x7820
0x761a: mov      r12d, 1
0x7620: add      rsp, 0x10
0x7624: mov      eax, r12d
0x7627: pop      r12
0x7629: ret
0x762a: nop      word ptr [rax + rax]
0x7630: and      r12d, 1
0x7634: je       0x761a
0x7636: xor      eax, eax
0x7638: call     0xce70
0x763d: jmp      0x75ad
0x7642: nop      word ptr [rax + rax]
0x7648: and      r12d, 1

```

```

0x764c: je      0x761a
0x764e: xor      eax, eax
0x7650: call     0xd380
0x7655: jmp      0x75ad
0x765a: nop      word ptr [rax + rax]
0x7660: and      r12d, 1
0x7664: je      0x761a
0x7666: xor      eax, eax
0x7668: call     0xd400
0x766d: jmp      0x75ad
0x7672: nop      word ptr [rax + rax]
0x7678: and      r12d, 1
0x767c: je      0x761a
0x767e: xor      eax, eax
0x7680: call     0xd480
0x7685: jmp      0x75ad
0x768a: nop      word ptr [rax + rax]
0x7690: xor      eax, eax
0x7692: mov      r12d, 1
0x7698: call     0xca60
0x769d: jmp      0x75ad
0x76a2: nop      word ptr [rax + rax]
0x76a8: and      r12d, 1
0x76ac: je      0x761a
0x76b2: xor      eax, eax
0x76b4: call     0xca80
0x76b9: jmp      0x75ad
0x76be: nop
0x76c0: xor      eax, eax
0x76c2: mov      r12d, 1
0x76c8: call     0xcab0
0x76cd: jmp      0x75ad
0x76d2: nop      word ptr [rax + rax]
0x76d8: and      r12d, 1
0x76dc: je      0x761a
0x76e2: xor      eax, eax
0x76e4: call     0xd700
0x76e9: jmp      0x75ad
0x76ee: nop
0x76f0: xor      eax, eax
0x76f2: mov      r12d, 1
0x76f8: call     0xcae0
0x76fd: jmp      0x75ad
0x7702: nop      word ptr [rax + rax]
0x7708: and      r12d, 1
0x770c: je      0x761a
0x7712: xor      eax, eax
0x7714: call     0xd1f0

```

```

0x7719: jmp      0x75ad
0x771e: nop
0x7720: and      r12d, 1
0x7724: je       0x761a
0x772a: xor      eax, eax
0x772c: call     0xd250
0x7731: jmp      0x75ad
0x7736: nop      word ptr cs:[rax + rax]
0x7740: and      r12d, 1
0x7744: je       0x761a
0x774a: xor      eax, eax
0x774c: call     0xd330
0x7751: jmp      0x75ad
0x7756: nop      word ptr cs:[rax + rax]
0x7760: and      r12d, 1
0x7764: je       0x761a
0x776a: xor      eax, eax
0x776c: call     0xd2b0
0x7771: jmp      0x75ad
0x7776: nop      word ptr cs:[rax + rax]
0x7780: and      r12d, 1
0x7784: je       0x761a
0x778a: xor      eax, eax
0x778c: call     0xcf80
0x7791: jmp      0x75ad
0x7796: nop      word ptr cs:[rax + rax]
0x77a0: and      r12d, 1
0x77a4: je       0x761a
0x77aa: mov      edi, 1
0x77af: call     0xd1a0
0x77b4: jmp      0x75ad
0x77b9: nop      dword ptr [rax]
0x77c0: and      r12d, 1
0x77c4: je       0x761a
0x77ca: xor      eax, eax
0x77cc: call     0xcf10
0x77d1: jmp      0x75ad
0x77d6: nop      word ptr cs:[rax + rax]
0x77e0: mov      dword ptr [rsp + 0xc], edi
0x77e4: and      r12d, 1
0x77e8: je       0x75ad
0x77ee: call     0x4f00
0x77f3: mov      r8, rax
0x77f6: movsxd   rax, dword ptr [rsp + 0xc]
0x77fb: mov      rdx, qword ptr [r8]
0x77fe: mov      rdi, rax
0x7801: test     byte ptr [rdx + rax*2 + 1], 0x40
0x7806: je       0x7818

```



```

0x7808: call    0xcd30
0x780d: jmp      0x75ad
0x7812: nop      word ptr [rax + rax]
0x7818: xor      r12d, r12d
0x781b: jmp      0x75ad
0x7820: xor      eax, eax
0x7822: call    0xcde0
0x7827: jmp      0x75ad
0x782c: nop      dword ptr [rax]
0x7830: endbr64
0x7834: xor      eax, eax
0x7836: cmp      dword ptr [rip + 0x2315f], 6
0x783d: jle      0x7869
0x783f: mov      edx, dword ptr [rip + 0x22413]
0x7845: mov      eax, dword ptr [rip + 0x2314d]
0x784b: test     edx, edx
0x784d: je       0x7860
0x784f: cmp      eax, 0xb
0x7852: setg     al
0x7855: movzx    eax, al
0x7858: ret
0x7859: nop      dword ptr [rax]
0x7860: cmp      eax, 5
0x7863: setg     al
0x7866: movzx    eax, al
0x7869: ret
0x786a: nop      word ptr [rax + rax]
0x7870: endbr64
0x7874: xor      eax, eax
0x7876: cmp      dword ptr [rip + 0x23067], 1
0x787d: sete     al
0x7880: ret
0x7881: nop      word ptr cs:[rax + rax]
0x788c: nop      dword ptr [rax]
0x7890: endbr64
0x7894: push     r13
0x7896: xor      eax, eax
0x7898: push     r12
0x789a: push     rbp
0x789b: push     rbx
0x789c: mov      ebx, edi
0x789e: sub      rsp, 8
0x78a2: mov      ebp, dword ptr [rip + 0x230f4]
0x78a8: mov      r12d, dword ptr [rip + 0x230e9]
0x78af: call    0xb520
0x78b4: mov      ecx, dword ptr [rip + 0x230e2]
0x78ba: test     ebx, ebx
0x78bc: jne      0x78c6

```

```

0x78be: cmp     ecx, ebp
0x78c0: je      0x7b00
0x78c6: mov     edx, dword ptr [rip + 0x230bc]
0x78cc: test    edx, edx
0x78ce: jne     0x7a60
0x78d4: mov     rdi, qword ptr [rip + 0x22fd5]
0x78db: cmp     ecx, 1
0x78de: jle     0x79e0
0x78e4: cmp     ecx, 2
0x78e7: je      0x7b70
0x78ed: cmp     ecx, 6
0x78f0: jle     0x790d
0x78f2: mov     eax, dword ptr [rip + 0x22360]
0x78f8: movsxd  r8, dword ptr [rip + 0x23099]
0x78ff: test    eax, eax
0x7901: je      0x7a90
0x7907: cmp     r8d, 0xb
0x790b: jg      0x7950
0x790d: mov     r8d, 0x50
0x7913: mov     ecx, 2
0x7918: mov     edx, 0x10000
0x791d: mov     esi, 0x10000
0x7922: call    0xf590
0x7927: mov     r8d, 0x50
0x792d: mov     ecx, 2
0x7932: mov     rdi, qword ptr [rip + 0x22f6f]
0x7939: mov     edx, 0x10000
0x793e: mov     esi, 0x10000
0x7943: call    0xf590
0x7948: jmp     0x7ad2
0x794d: nop     dword ptr [rax]
0x7950: mov     ebp, r8d
0x7953: and     r8d, 1
0x7957: sar     ebp, 1
0x7959: lea     r13d, [r8 + rbp]
0x795d: movsxd  rbp, ebp
0x7960: call    0xf470
0x7965: movsxd  r13, r13d
0x7968: mov     rdi, rax
0x796b: call    0xc6a0
0x7970: mov     rdi, qword ptr [rip + 0x22f31]
0x7977: mov     r12d, eax
0x797a: call    0xf470
0x797f: mov     rdi, rax
0x7982: call    0xc6a0
0x7987: cmp     r12d, eax
0x798a: jg      0x7b20
0x7990: mov     eax, dword ptr [rip + 0x23006]

```

```

0x7996: mov     r8, r13
0x7999: xor     esi, esi
0x799b: mov     edx, 1
0x79a0: mov     rdi, qword ptr [rip + 0x22f09]
0x79a7: lea     ecx, [rax - 3]
0x79aa: movsxd   rcx, ecx
0x79ad: call    0xf590
0x79b2: mov     r8, rbp
0x79b5: mov     edx, 1
0x79ba: mov     rsi, r13
0x79bd: mov     eax, dword ptr [rip + 0x22fd9]
0x79c3: mov     rdi, qword ptr [rip + 0x22ede]
0x79ca: lea     ecx, [rax - 3]
0x79cd: movsxd   rcx, ecx
0x79d0: call    0xf590
0x79d5: jmp     0x7ad2
0x79da: nop     word ptr [rax + rax]
0x79e0: mov     r8d, 0x50
0x79e6: mov     ecx, 2
0x79eb: mov     edx, 0x10000
0x79f0: mov     esi, 0x10000
0x79f5: call    0xf590
0x79fa: mov     r8d, 0x50
0x7a00: mov     ecx, 2
0x7a05: mov     rdi, qword ptr [rip + 0x22e9c]
0x7a0c: mov     edx, 0x10000
0x7a11: mov     esi, 0x10000
0x7a16: call    0xf590
0x7a1b: xor     esi, esi
0x7a1d: xor     edi, edi
0x7a1f: call    0x16d30
0x7a24: mov     eax, dword ptr [rip + 0x22f72]
0x7a2a: xor     edi, edi
0x7a2c: lea     esi, [rax - 1]
0x7a2f: movsxd   rsi, esi
0x7a32: call    0xde50
0x7a37: mov     eax, dword ptr [rip + 0x22f5f]
0x7a3d: mov     edi, dword ptr [rip + 0x22f55]
0x7a43: cmp     eax, 1
0x7a46: lea     esi, [rax - 2]
0x7a49: mov     eax, 0
0x7a4e: cmov     esi, eax
0x7a51: add     rsp, 8
0x7a55: pop     rbx
0x7a56: pop     rbp
0x7a57: pop     r12
0x7a59: pop     r13
0x7a5b: jmp     0xc890

```

```

0x7a60: imul    ecx, dword ptr [rip + 0x22f31]
0x7a67: mov     rdi, qword ptr [rip + 0x22e5a]
0x7a6e: lea     esi, [rcx + rcx + 4]
0x7a72: movsxd  rsi, esi
0x7a75: call    0x1d690
0x7a7a: mov     ecx, dword ptr [rip + 0x22f1c]
0x7a80: mov     qword ptr [rip + 0x22e41], rax
0x7a87: jmp     0x78d4
0x7a8c: nop     dword ptr [rax]
0x7a90: cmp     r8d, 5
0x7a94: jle     0x790d
0x7a9a: sub     ecx, 3
0x7a9d: mov     edx, 1
0x7aa2: xor     esi, esi
0x7aa4: movsxd  rcx, ecx
0x7aa7: call    0xf590
0x7aac: mov     eax, dword ptr [rip + 0x22eea]
0x7ab2: mov     edx, 1
0x7ab7: xor     esi, esi
0x7ab9: movsxd  r8, dword ptr [rip + 0x22ed8]
0x7ac0: mov     rdi, qword ptr [rip + 0x22de1]
0x7ac7: lea     ecx, [rax - 3]
0x7aca: movsxd  rcx, ecx
0x7acd: call    0xf590
0x7ad2: movsxd  rdi, dword ptr [rip + 0x22ebf]
0x7ad9: xor     esi, esi
0x7adb: call    0x16d30
0x7ae0: mov     eax, dword ptr [rip + 0x22eb6]
0x7ae6: lea     esi, [rax - 1]
0x7ae9: movsxd  rsi, esi
0x7aec: movsxd  rdi, dword ptr [rip + 0x22ea5]
0x7af3: jmp     0x7a32
0x7af8: nop     dword ptr [rax + rax]
0x7b00: cmp     dword ptr [rip + 0x22e91], r12d
0x7b07: jne     0x78c6
0x7b0d: add     rsp, 8
0x7b11: pop     rbx
0x7b12: pop     rbp
0x7b13: pop     r12
0x7b15: pop     r13
0x7b17: ret
0x7b18: nop     dword ptr [rax + rax]
0x7b20: mov     eax, dword ptr [rip + 0x22e76]
0x7b26: mov     rdi, qword ptr [rip + 0x22d83]
0x7b2d: mov     r8, rbp
0x7b30: mov     rsi, r13
0x7b33: mov     edx, 1
0x7b38: lea     ecx, [rax - 3]

```

```

0x7b3b: movsxd rcx, ecx
0x7b3e: call 0xf590
0x7b43: mov eax, dword ptr [rip + 0x22e53]
0x7b49: mov r8, r13
0x7b4c: xor esi, esi
0x7b4e: mov rdi, qword ptr [rip + 0x22d53]
0x7b55: mov edx, 1
0x7b5a: lea ecx, [rax - 3]
0x7b5d: movsxd rcx, ecx
0x7b60: call 0xf590
0x7b65: jmp 0x7ad2
0x7b6a: nop word ptr [rax + rax]
0x7b70: mov r8d, 0x50
0x7b76: mov ecx, 2
0x7b7b: mov edx, 0x10000
0x7b80: mov esi, 0x10000
0x7b85: call 0xf590
0x7b8a: mov r8d, 0x50
0x7b90: mov ecx, 2
0x7b95: mov rdi, qword ptr [rip + 0x22d0c]
0x7b9c: mov edx, 0x10000
0x7ba1: mov esi, 0x10000
0x7ba6: call 0xf590
0x7bab: xor esi, esi
0x7bad: xor edi, edi
0x7baf: call 0x16d30
0x7bb4: mov eax, dword ptr [rip + 0x22de2]
0x7bba: lea esi, [rax - 1]
0x7bbd: movsxd rsi, esi
0x7bc0: jmp 0x7aec
0x7bc5: nop word ptr cs:[rax + rax]
0x7bd0: endbr64
0x7bd4: push rbx
0x7bd5: xor eax, eax
0x7bd7: mov ebx, edi
0x7bd9: xor edi, edi
0x7bdb: call 0x7890
0x7be0: cmp ebx, 0x12
0x7be3: je 0x7cc8
0x7be9: mov esi, dword ptr [rip + 0x22ced]
0x7bef: test esi, esi
0x7bf1: je 0x7bf8
0x7bf3: pop rbx
0x7bf4: ret
0x7bf5: nop dword ptr [rax]
0x7bf8: mov rdi, qword ptr [rip + 0x22cb1]
0x7bff: call 0xf480
0x7c04: mov rdi, qword ptr [rip + 0x22c9d]

```

```

0x7c0b: call    0xf480
0x7c10: mov     rdi, qword ptr [rip + 0x22c99]
0x7c17: call    0xf530
0x7c1c: mov     rdi, qword ptr [rip + 0x22c85]
0x7c23: call    0xf530
0x7c28: mov     ecx, dword ptr [rip + 0x22cb6]
0x7c2e: test    ecx, ecx
0x7c30: jne     0x7ce0
0x7c36: cmp     dword ptr [rip + 0x22d5f], 6
0x7c3d: jle     0x7c60
0x7c3f: mov     edx, dword ptr [rip + 0x22013]
0x7c45: mov     eax, dword ptr [rip + 0x22d4d]
0x7c4b: test    edx, edx
0x7c4d: je      0x7d00
0x7c53: cmp     eax, 0xb
0x7c56: jg      0x7c6e
0x7c58: nop     dword ptr [rax + rax]
0x7c60: xor     eax, eax
0x7c62: call    0xa6c0
0x7c67: xor     eax, eax
0x7c69: call    0xa300
0x7c6e: xor     eax, eax
0x7c70: call    0x16d50
0x7c75: mov     rdi, qword ptr [rip + 0x22c34]
0x7c7c: call    0x11010
0x7c81: mov     eax, dword ptr [rip + 0x21fd1]
0x7c87: test    eax, eax
0x7c89: jne     0x7cf0
0x7c8b: xor     eax, eax
0x7c8d: call    0xdea0
0x7c92: xor     eax, eax
0x7c94: call    0xd870
0x7c99: xor     eax, eax
0x7c9b: call    0xd740
0x7ca0: xor     eax, eax
0x7ca2: call    0xa820
0x7ca7: cmp     ebx, 0x12
0x7caa: jne     0x7bf3
0x7cb0: mov     rdi, qword ptr [rip + 0x22bf9]
0x7cb7: call    0x11100
0x7cbc: pop     rbx
0x7cbd: mov     rdi, rax
0x7cc0: jmp     0xc4b0
0x7cc5: nop     dword ptr [rax]
0x7cc8: mov     edi, 1
0x7ccd: call    0x9b00
0x7cd2: xor     eax, eax
0x7cd4: call    0xa6c0

```

```

0x7cd9: jmp      0x7be9
0x7cde: nop
0x7ce0: mov      rdi, qword ptr [rip + 0x22be1]
0x7ce7: call     0xbb40
0x7cec: jmp      0x7c8b
0x7cee: nop
0x7cf0: mov      rdi, qword ptr [rip + 0x22bb1]
0x7cf7: call     0x11010
0x7cfc: jmp      0x7c8b
0x7cfe: nop
0x7d00: cmp      eax, 5
0x7d03: jle      0x7c60
0x7d09: jmp      0x7c6e
0x7d0e: nop
0x7d10: endbr64
0x7d14: push     r12
0x7d16: xor      eax, eax
0x7d18: mov      r12, rdi
0x7d1b: push     rbp
0x7d1c: push     rbx
0x7d1d: call     0xb510
0x7d22: mov      rdi, rax
0x7d25: mov      rbp, rax
0x7d28: call     0x49b0
0x7d2d: test     rax, rax
0x7d30: je       0x7d39
0x7d32: cmp      byte ptr [rbp + rax - 1], 7
0x7d37: jne      0x7d70
0x7d39: xor      eax, eax
0x7d3b: call     0xa5a0
0x7d40: test     r12, r12
0x7d43: je       0x7ded
0x7d49: xor      edx, edx
0x7d4b: xor      esi, esi
0x7d4d: mov      rdi, r12
0x7d50: call     0xe480
0x7d55: xor      eax, eax
0x7d57: call     0xd740
0x7d5c: pop      rbx
0x7d5d: xor      eax, eax
0x7d5f: pop      rbp
0x7d60: pop      r12
0x7d62: jmp      0xa820
0x7d67: nop      word ptr [rax + rax]
0x7d70: mov      rdi, rbp
0x7d73: call     0x9fe0
0x7d78: mov      rdi, rax
0x7d7b: mov      rbx, rax

```

```

0x7d7e: call    0x49b0
0x7d83: lea     rdi, [rax + 0x80]
0x7d8a: call    0x1d670
0x7d8f: mov     r8, rbx
0x7d92: mov     esi, 1
0x7d97: lea     rcx, [rip + 0x18762]
0x7d9e: mov     rbp, rax
0x7da1: mov     rdi, rax
0x7da4: mov     rdx, -1
0x7dab: xor     eax, eax
0x7dad: call    0x4f10
0x7db2: mov     edx, 1
0x7db7: mov     esi, 2
0x7dbc: mov     rdi, rbp
0x7dbf: call    0xe480
0x7dc4: mov     rdi, rbp
0x7dc7: call    0x18d10
0x7dcc: xor     eax, eax
0x7dce: call    0xa5a0
0x7dd3: xor     eax, eax
0x7dd5: call    0xa820
0x7dda: mov     edi, 1
0x7ddf: call    0x4ea0
0x7de4: test    r12, r12
0x7de7: jne     0x7d49
0x7ded: xor     eax, eax
0x7def: call    0xe4e0
0x7df4: jmp     0x7d55
0x7df9: nop     dword ptr [rax]
0x7e00: endbr64
0x7e04: push    r13
0x7e06: mov     edx, 1
0x7e0b: push    r12
0x7e0d: mov     r12, rsi
0x7e10: push    rbp
0x7e11: mov     rbp, rdi
0x7e14: sub     rsp, 0x10
0x7e18: mov     rax, qword ptr fs:[0x28]
0x7e21: mov     qword ptr [rsp + 8], rax
0x7e26: xor     eax, eax
0x7e28: mov     r13, rsp
0x7e2b: mov     rsi, r13
0x7e2e: call    0x19200
0x7e33: test    rax, rax
0x7e36: je      0x7e48
0x7e38: mov     rdi, qword ptr [rsp]
0x7e3c: mov     rsi, r12
0x7e3f: call    0x4b30

```



```

0x7e44: test    eax, eax
0x7e46: je      0x7e5f
0x7e48: mov     rdi, r12
0x7e4b: call    0x1d7a0
0x7e50: mov     rsi, r13
0x7e53: mov     rdi, rbp
0x7e56: mov     qword ptr [rsp], rax
0x7e5a: call    0x19150
0x7e5f: mov     rax, qword ptr [rsp + 8]
0x7e64: sub     rax, qword ptr fs:[0x28]
0x7e6d: jne     0x7e79
0x7e6f: add     rsp, 0x10
0x7e73: pop     rbp
0x7e74: pop     r12
0x7e76: pop     r13
0x7e78: ret
0x7e79: call    0x49d0
0x7e7e: nop
0x7e80: endbr64
0x7e84: sub     rsp, 0x18
0x7e88: mov     edx, esi
0x7e8a: mov     rax, qword ptr fs:[0x28]
0x7e93: mov     qword ptr [rsp + 8], rax
0x7e98: xor     eax, eax
0x7e9a: mov     rsi, rsp
0x7e9d: call    0x19200
0x7ea2: test    rax, rax
0x7ea5: cmovne  rax, qword ptr [rsp]
0x7eaa: mov     rdx, qword ptr [rsp + 8]
0x7eaf: sub     rdx, qword ptr fs:[0x28]
0x7eb8: jne     0x7ebf
0x7eba: add     rsp, 0x18
0x7ebe: ret
0x7ebf: call    0x49d0
0x7ec4: nop     word ptr cs:[rax + rax]
0x7ecf: nop
0x7ed0: endbr64
0x7ed4: test    sil, 2
0x7ed8: je      0x7ee0
0x7eda: jmp     0x7570
0x7edf: nop
0x7ee0: xor     eax, eax
0x7ee2: ret
0x7ee3: nop     word ptr cs:[rax + rax]
0x7eee: nop
0x7ef0: endbr64
0x7ef4: push    r15
0x7ef6: push    r14

```

```

0x7ef8: push    r13
0x7efa: push    r12
0x7efc: push    rbp
0x7efd: push    rbx
0x7efe: sub     rsp, 8
0x7f02: test    rdi, rdi
0x7f05: je      0x7fc8
0x7f0b: mov     rbp, rdi
0x7f0e: call    0x49b0
0x7f13: lea     r13, [rax + 1]
0x7f17: mov     rdi, r13
0x7f1a: call    0x1d670
0x7f1f: movzx   ebx, byte ptr [rbp]
0x7f23: mov     r15, rax
0x7f26: test    bl, bl
0x7f28: je      0x7fd0
0x7f2e: add     rbp, 1
0x7f32: xor     r12d, r12d
0x7f35: jmp     0x7f6d
0x7f37: nop     word ptr [rax + rax]
0x7f40: call    0x4f00
0x7f45: movsx   rdx, bl
0x7f49: add     r12d, 1
0x7f4d: mov     rax, qword ptr [rax]
0x7f50: test    byte ptr [rax + rdx*2 + 1], 0x40
0x7f55: mov     eax, 0x3f
0x7f5a: cmov    ebx, eax
0x7f5d: add     rbp, 1
0x7f61: mov     byte ptr [r15 + r14], bl
0x7f65: movzx   ebx, byte ptr [rbp - 1]
0x7f69: test    bl, bl
0x7f6b: je      0x7fa5
0x7f6d: movsxd  r14, r12d
0x7f70: cmp     bl, 9
0x7f73: jne     0x7f40
0x7f75: add     r13, 8
0x7f79: mov     rdi, r15
0x7f7c: add     rbp, 1
0x7f80: add     r12d, 8
0x7f84: mov     rsi, r13
0x7f87: call    0x1d690
0x7f8c: mov     r15, rax
0x7f8f: movabs  rax, 0x2020202020202020
0x7f99: mov     qword ptr [r15 + r14], rax
0x7f9d: movzx   ebx, byte ptr [rbp - 1]
0x7fa1: test    bl, bl
0x7fa3: jne     0x7f6d
0x7fa5: movsxd  r12, r12d

```

```

0x7fa8: add    r12, r15
0x7fab: mov     byte ptr [r12], 0
0x7fb0: add     rsp, 8
0x7fb4: mov     rax, r15
0x7fb7: pop      rbx
0x7fb8: pop      rbp
0x7fb9: pop      r12
0x7fbb: pop      r13
0x7fbd: pop      r14
0x7fbf: pop      r15
0x7fc1: ret
0x7fc2: nop      word ptr [rax + rax]
0x7fc8: xor      r15d, r15d
0x7fcb: jmp     0x7fb0
0x7fcd: nop      dword ptr [rax]
0x7fd0: mov     r12, rax
0x7fd3: jmp     0x7fab
0x7fd5: nop      word ptr cs:[rax + rax]
0x7fe0: endbr64
0x7fe4: push    rbx
0x7fe5: mov     rbx, rdi
0x7fe8: call    0x49b0
0x7fed: lea     rdi, [rax + rax*2 + 9]
0x7ff2: call    0x1d670
0x7ff7: mov     byte ptr [rax], 0x28
0x7ffa: cmp     byte ptr [rbx + 1], 0
0x7ffe: je      0x8050
0x8000: mov     edx, 1
0x8005: nop      dword ptr [rax]
0x8008: movzx   ecx, byte ptr [rbx]
0x800b: mov     byte ptr [rax + rdx + 1], 0x2c
0x8010: add     rbx, 1
0x8014: mov     byte ptr [rax + rdx], cl
0x8017: mov     rcx, rdx
0x801a: add     rdx, 3
0x801e: mov     byte ptr [rax + rdx - 1], 0x20
0x8023: cmp     byte ptr [rbx + 1], 0
0x8027: jne     0x8008
0x8029: lea     r8, [rcx + 4]
0x802d: lea     rdi, [rcx + 5]
0x8031: add     rcx, 6
0x8035: movzx   esi, byte ptr [rbx]
0x8038: mov     byte ptr [rax + rdx], sil
0x803c: mov     byte ptr [rax + r8], 0x29
0x8041: mov     byte ptr [rax + rdi], 0x20
0x8045: mov     byte ptr [rax + rcx], 0
0x8049: pop      rbx
0x804a: ret

```

```

0x804b: nop      dword ptr [rax + rax]
0x8050: mov      ecx, 4
0x8055: mov      edi, 3
0x805a: mov      r8d, 2
0x8060: mov      edx, 1
0x8065: jmp      0x8035
0x8067: nop      word ptr [rax + rax]
0x8070: endbr64
0x8074: push     r15
0x8076: push     r14
0x8078: push     r13
0x807a: push     r12
0x807c: mov      r12, rdi
0x807f: push     rbp
0x8080: mov      rbp, rsi
0x8083: push     rbx
0x8084: mov      ebx, edx
0x8086: sub      rsp, 0x28
0x808a: mov      rax, qword ptr fs:[0x28]
0x8093: mov      qword ptr [rsp + 0x18], rax
0x8098: mov      eax, edx
0x809a: and      eax, 0x10
0x809d: mov      dword ptr [rsp + 0xc], eax
0x80a1: jne      0x82f0
0x80a7: xor      r13d, r13d
0x80aa: xor      eax, eax
0x80ac: call     0xcff0
0x80b1: test     r12, r12
0x80b4: je       0x8103
0x80b6: mov      rdi, r12
0x80b9: call     0x7ef0
0x80be: mov      r12, rax
0x80c1: test     bl, 8
0x80c4: jne      0x8330
0x80ca: mov      rdi, r12
0x80cd: call     0xd610
0x80d2: test     bl, 0x20
0x80d5: jne      0x8310
0x80db: mov      rdi, r12
0x80de: call     0x18d10
0x80e3: test     rbp, rbp
0x80e6: je       0x8103
0x80e8: mov      rdi, rbp
0x80eb: call     0x7fe0
0x80f0: mov      r12, rax
0x80f3: mov      rdi, rax
0x80f6: call     0xd610
0x80fb: mov      rdi, r12

```

```

0x80fe: call    0x18d10
0x8103: xor     eax, eax
0x8105: call    0xd870
0x810a: xor     eax, eax
0x810c: call    0xd740
0x8111: xor     eax, eax
0x8113: call    0xa820
0x8118: test    bl, 4
0x811b: jne     0x8300
0x8121: mov     r14d, ebx
0x8124: lea     r12, [rsp + 0x14]
0x8129: and     r14d, 2
0x812d: mov     rdi, r12
0x8130: call    0xb180
0x8135: test    rax, rax
0x8138: je      0x81b2
0x813a: mov     rdx, qword ptr [rax]
0x813d: mov     rax, qword ptr [rax + 0x10]
0x8141: movzx   r15d, byte ptr [rdx]
0x8145: test    rax, rax
0x8148: je      0x8250
0x814e: cmp     byte ptr [rax + 0x1b], 0
0x8152: je      0x8250
0x8158: mov     rcx, qword ptr [rax]
0x815b: lea     rax, [rip + 0x20ebe]
0x8162: sub     rcx, rax
0x8165: movabs  rax, 0xea0ea0ea0ea0ea1
0x816f: imul    rcx
0x8172: sar     rcx, 0x3f
0x8176: sar     rdx, 1
0x8179: sub     rdx, rcx
0x817c: mov     r15d, edx
0x817f: not     r15d
0x8182: cmp     edx, 7
0x8185: jne     0x81d0
0x8187: xor     edi, edi
0x8189: xor     eax, eax
0x818b: call    0x7bd0
0x8190: xor     eax, eax
0x8192: call    0xdea0
0x8197: xor     eax, eax
0x8199: call    0xd740
0x819e: xor     eax, eax
0x81a0: call    0xa820
0x81a5: mov     rdi, r12
0x81a8: call    0xb180
0x81ad: test    rax, rax
0x81b0: jne     0x813a

```

```

0x81b2: xor     eax, eax
0x81b4: call    0xa5a0
0x81b9: xor     eax, eax
0x81bb: call    0xdea0
0x81c0: xor     eax, eax
0x81c2: call    0xd740
0x81c7: jmp     0x812d
0x81cc: nop     dword ptr [rax]
0x81d0: cmp     r15d, 7
0x81d4: je      0x82b0
0x81da: cmp     r15d, -0x3d
0x81de: jne     0x8256
0x81e0: test    rbp, rbp
0x81e3: je      0x8360
0x81e9: xor     edi, edi
0x81eb: movsx   r15d, byte ptr [rbp]
0x81f0: call    0xdba0
0x81f5: mov     esi, dword ptr [rsp + 0xc]
0x81f9: test    esi, esi
0x81fb: je      0x821f
0x81fd: test    r13, r13
0x8200: je      0x821f
0x8202: mov     rdi, r13
0x8205: call    0xc970
0x820a: xor     eax, eax
0x820c: call    0xd870
0x8211: xor     eax, eax
0x8213: call    0xd740
0x8218: xor     eax, eax
0x821a: call    0xa820
0x821f: cmp     r15d, 7
0x8223: je      0x82e1
0x8229: mov     eax, r15d
0x822c: mov     rdx, qword ptr [rsp + 0x18]
0x8231: sub     rdx, qword ptr fs:[0x28]
0x823a: jne     0x83d9
0x8240: add     rsp, 0x28
0x8244: pop     rbx
0x8245: pop     rbp
0x8246: pop     r12
0x8248: pop     r13
0x824a: pop     r14
0x824c: pop     r15
0x824e: ret
0x824f: nop
0x8250: cmp     r15d, 7
0x8254: je      0x82b0
0x8256: mov     eax, dword ptr [rsp + 0x14]

```

```

0x825a: lea     edx, [rax - 1]
0x825d: mov     dword ptr [rsp + 0x14], edx
0x8261: test    eax, eax
0x8263: je      0x83a8
0x8269: test    r14d, r14d
0x826c: je      0x8280
0x826e: mov     esi, ebx
0x8270: mov     edi, r15d
0x8273: call    0x7570
0x8278: test    eax, eax
0x827a: jne     0x8256
0x827c: nop     dword ptr [rax]
0x8280: test    rbp, rbp
0x8283: je      0x83b8
0x8289: mov     esi, r15d
0x828c: mov     rdi, rbp
0x828f: call    0x4a00
0x8294: test    rax, rax
0x8297: jne     0x83b8
0x829d: call    0xa5a0
0x82a2: jmp     0x8190
0x82a7: nop     word ptr [rax + rax]
0x82b0: xor     edi, edi
0x82b2: call    0xdba0
0x82b7: mov     ecx, dword ptr [rsp + 0xc]
0x82bb: test    ecx, ecx
0x82bd: je      0x82e1
0x82bf: test    r13, r13
0x82c2: je      0x82e1
0x82c4: mov     rdi, r13
0x82c7: call    0xc970
0x82cc: xor     eax, eax
0x82ce: call    0xd870
0x82d3: xor     eax, eax
0x82d5: call    0xd740
0x82da: xor     eax, eax
0x82dc: call    0xa820
0x82e1: xor     eax, eax
0x82e3: jmp     0x822c
0x82e8: nop     dword ptr [rax + rax]
0x82f0: xor     eax, eax
0x82f2: call    0xc8c0
0x82f7: mov     r13, rax
0x82fa: jmp     0x80aa
0x82ff: nop
0x8300: xor     eax, eax
0x8302: call    0xa5a0
0x8307: jmp     0x8121

```

```

0x830c: nop      dword ptr [rax]
0x8310: xor      eax, eax
0x8312: call     0xccf0
0x8317: mov      rdi, r12
0x831a: call     0x18d10
0x831f: test     rbp, rbp
0x8322: jne      0x80e8
0x8328: jmp      0x8103
0x832d: nop      dword ptr [rax]
0x8330: lea      rdi, [rip + 0x181da]
0x8337: call     0xd610
0x833c: mov      edi, 1
0x8341: call     0xdba0
0x8346: mov      rdi, r12
0x8349: call     0xd610
0x834e: test     bl, 0x20
0x8351: je       0x80db
0x8357: jmp      0x8310
0x8359: nop      dword ptr [rax]
0x8360: xor      edi, edi
0x8362: call     0xdba0
0x8367: mov      edx, dword ptr [rsp + 0xc]
0x836b: mov      eax, 0xffffffffc3
0x8370: test     edx, edx
0x8372: je       0x822c
0x8378: test     r13, r13
0x837b: je       0x822c
0x8381: mov      rdi, r13
0x8384: call     0xc970
0x8389: xor      eax, eax
0x838b: call     0xd870
0x8390: xor      eax, eax
0x8392: call     0xd740
0x8397: xor      eax, eax
0x8399: call     0xa820
0x839e: jmp      0x8229
0x83a3: nop      dword ptr [rax + rax]
0x83a8: call     0xd870
0x83ad: jmp      0x8190
0x83b2: nop      word ptr [rax + rax]
0x83b8: xor      edi, edi
0x83ba: call     0xdba0
0x83bf: mov      eax, dword ptr [rsp + 0xc]
0x83c3: test     eax, eax
0x83c5: je       0x8229
0x83cb: test     r13, r13
0x83ce: jne      0x8202
0x83d4: jmp      0x8229

```



```

0x83d9: call    0x49d0
0x83de: nop
0x83e0: endbr64
0x83e4: push     r15
0x83e6: push     r14
0x83e8: mov      r14, rdi
0x83eb: push     r13
0x83ed: mov      r13, rsi
0x83f0: push     r12
0x83f2: mov      r12, rdx
0x83f5: push     rbp
0x83f6: mov      rbp, rcx
0x83f9: push     rbx
0x83fa: sub      rsp, 0x38
0x83fe: mov      rax, qword ptr fs:[0x28]
0x8407: mov      qword ptr [rsp + 0x28], rax
0x840c: xor      eax, eax
0x840e: call     0xcff0
0x8413: test     r14, r14
0x8416: je       0x8420
0x8418: mov      rdi, r14
0x841b: call     0xd540
0x8420: test     r12, r12
0x8423: je       0x8450
0x8425: mov      rdi, r12
0x8428: call     0xd610
0x842d: test     rbp, rbp
0x8430: je       0x8450
0x8432: mov      rsi, r12
0x8435: mov      rdi, rbp
0x8438: xor      eax, eax
0x843a: mov      r14d, 1
0x8440: call     0x7e00
0x8445: jmp      0x8453
0x8447: nop      word ptr [rax + rax]
0x8450: xor      r14d, r14d
0x8453: xor      eax, eax
0x8455: lea      r12, [rsp + 0x1c]
0x845a: lea      rbx, [rip + 0x17c03]
0x8461: call     0xd870
0x8466: xor      eax, eax
0x8468: call     0xd740
0x846d: xor      eax, eax
0x846f: call     0xa820
0x8474: lea      rax, [rsp + 0x20]
0x8479: mov      qword ptr [rsp], rax
0x847d: mov      rdi, r12
0x8480: call     0xb180

```

```

0x8485: test    rax, rax
0x8488: je      0x853f
0x848e: mov     rdx, qword ptr [rax]
0x8491: mov     rax, qword ptr [rax + 0x10]
0x8495: movzx   r15d, byte ptr [rdx]
0x8499: test    rax, rax
0x849c: je      0x84d7
0x849e: cmp     byte ptr [rax + 0x1b], 0
0x84a2: je      0x84d7
0x84a4: mov     rcx, qword ptr [rax]
0x84a7: lea     rax, [rip + 0x20b72]
0x84ae: sub     rcx, rax
0x84b1: movabs  rax, 0xea0ea0ea0ea0ea1
0x84bb: imul    rcx
0x84be: sar     rcx, 0x3f
0x84c2: sar     rdx, 1
0x84c5: sub     rdx, rcx
0x84c8: mov     r15d, edx
0x84cb: not     r15d
0x84ce: cmp     edx, 0x3c
0x84d1: je      0x8560
0x84d7: lea     eax, [r15 + 0x22]
0x84db: cmp     eax, 0x29
0x84de: ja      0x8505
0x84e0: movsxd  rax, dword ptr [rbx + rax*4]
0x84e4: add     rax, rbx
0x84e7: jmp     rax
0x84ea: -----

```

```

0x84ee: -----
0x84ef: -----
0x84f0: mov     esi, 3
0x84f5: mov     edi, r15d
0x84f8: call    0x7570
0x84fd: test    eax, eax
0x84ff: je      0x8670
0x8505: mov     eax, dword ptr [rsp + 0x1c]
0x8509: lea     edx, [rax - 1]
0x850c: mov     dword ptr [rsp + 0x1c], edx
0x8510: test    eax, eax
0x8512: jne     0x84f0
0x8514: call    0xd870
0x8519: xor     eax, eax
0x851b: call    0xdea0
0x8520: xor     eax, eax
0x8522: call    0xd740

```

```

0x8527: xor     eax, eax
0x8529: call    0xa820
0x852e: mov     rdi, r12
0x8531: call    0xb180
0x8536: test    rax, rax
0x8539: jne     0x848e
0x853f: xor     eax, eax
0x8541: call    0xa5a0
0x8546: xor     eax, eax
0x8548: call    0xdea0
0x854d: xor     eax, eax
0x854f: call    0xd740
0x8554: jmp     0x847d
0x8559: nop     dword ptr [rax]
0x8560: mov     rdi, r13
0x8563: call    0xdb10
0x8568: test    rbp, rbp
0x856b: je      0x857b
0x856d: mov     rsi, qword ptr [r13]
0x8571: mov     rdi, rbp
0x8574: xor     eax, eax
0x8576: call    0x7e00
0x857b: mov     rax, qword ptr [r13]
0x857f: mov     rdx, qword ptr [rsp + 0x28]
0x8584: sub     rdx, qword ptr fs:[0x28]
0x858d: jne     0x86b3
0x8593: add     rsp, 0x38
0x8597: pop     rbx
0x8598: pop     rbp
0x8599: pop     r12
0x859b: pop     r13
0x859d: pop     r14
0x859f: pop     r15
0x85a1: ret
0x85a2: nop     word ptr [rax + rax]
0x85a8: test    rbp, rbp
0x85ab: je      0x8519
0x85b1: lea     eax, [r14 + 1]
0x85b5: mov     rsi, qword ptr [rsp]
0x85b9: mov     rdi, rbp
0x85bc: mov     edx, eax
0x85be: mov     r15d, eax
0x85c1: call    0x19200
0x85c6: test    rax, rax
0x85c9: je      0x86a7
0x85cf: mov     r8, qword ptr [rsp + 0x20]
0x85d4: test    r8, r8
0x85d7: je      0x86a7

```

```

0x85dd: xor     edi, edi
0x85df: mov     qword ptr [rsp + 8], r8
0x85e4: mov     r14d, r15d
0x85e7: call    0xd1a0
0x85ec: mov     rdi, qword ptr [rsp + 8]
0x85f1: call    0xd610
0x85f6: xor     eax, eax
0x85f8: call    0xd870
0x85fd: xor     eax, eax
0x85ff: call    0xd740
0x8604: jmp     0x8519
0x8609: nop     dword ptr [rax]
0x8610: test    rbp, rbp
0x8613: je      0x8519
0x8619: xor     edi, edi
0x861b: test    r14d, r14d
0x861e: je      0x8680
0x8620: call    0xd1a0
0x8625: sub     r14d, 1
0x8629: je      0x864b
0x862b: mov     rsi, qword ptr [rsp]
0x862f: mov     rdi, rbp
0x8632: mov     edx, r14d
0x8635: call    0x19200
0x863a: mov     rdi, rax
0x863d: test    rax, rax
0x8640: cmovne  rdi, qword ptr [rsp + 0x20]
0x8646: call    0xd610
0x864b: xor     eax, eax
0x864d: call    0xd870
0x8652: xor     eax, eax
0x8654: call    0xd740
0x8659: jmp     0x8519
0x865e: nop
0x8660: xor     edi, edi
0x8662: xor     eax, eax
0x8664: call    0x7bd0
0x8669: jmp     0x8519
0x866e: nop
0x8670: call    0xa5a0
0x8675: jmp     0x8505
0x867a: nop     word ptr [rax + rax]
0x8680: call    0xd1a0
0x8685: xor     eax, eax
0x8687: call    0xd870
0x868c: xor     eax, eax
0x868e: call    0xd740
0x8693: jmp     0x8519

```

```

0x8698: nop      dword ptr [rax + rax]
0x86a0: xor      eax, eax
0x86a2: jmp      0x857f
0x86a7: xor      eax, eax
0x86a9: call     0xa5a0
0x86ae: jmp      0x8519
0x86b3: call     0x49d0
0x86b8: nop      dword ptr [rax + rax]
0x86c0: endbr64
0x86c4: push     r15
0x86c6: push     r14
0x86c8: push     r13
0x86ca: push     r12
0x86cc: push     rbp
0x86cd: push     rbx
0x86ce: sub      rsp, 0x18
0x86d2: test     edx, edx
0x86d4: je       0x8890
0x86da: mov      ebx, edx
0x86dc: cmp      edx, 3
0x86df: je       0x88f8
0x86e5: mov      rbp, rcx
0x86e8: test     rcx, rcx
0x86eb: je       0x8864
0x86f1: mov      dword ptr [rcx], 0
0x86f7: xor      eax, eax
0x86f9: mov      r13, rsi
0x86fc: lea      r15, [rip + 0x2091d]
0x8703: movabs   r14, 0xea0ea0ea0ea0ea1
0x870d: call     0xd870
0x8712: xor      eax, eax
0x8714: call     0xd740
0x8719: xor      eax, eax
0x871b: call     0xa820
0x8720: xor      edi, edi
0x8722: call     0xb180
0x8727: mov      r12, rax
0x872a: test     rax, rax
0x872d: je       0x8928
0x8733: mov      rax, qword ptr [rax]
0x8736: movzx    edi, byte ptr [rax]
0x8739: mov      rax, qword ptr [r12 + 0x10]
0x873e: test     rax, rax
0x8741: je       0x874d
0x8743: cmp      byte ptr [rax + 0x1b], 0
0x8747: jne      0x8808
0x874d: cmp      edi, -7
0x8750: jge      0x87b0

```

```

0x8752: cmp     edi, -0x3d
0x8755: je       0x8840
0x875b: cmp     edi, -0x32
0x875e: jne      0x88bb
0x8764: xor      eax, eax
0x8766: call     0xca40
0x876b: test     eax, eax
0x876d: jne      0x8880
0x8773: mov      dword ptr [rbp], 1
0x877a: xor      eax, eax
0x877c: call     0xdea0
0x8781: xor      eax, eax
0x8783: call     0xd870
0x8788: xor      eax, eax
0x878a: call     0xd740
0x878f: xor      eax, eax
0x8791: call     0xa820
0x8796: mov      rdi, r13
0x8799: call     0xdb10
0x879e: mov      r12, qword ptr [r13]
0x87a2: jmp      0x8867
0x87a7: nop      word ptr [rax + rax]
0x87b0: cmp      edi, 7
0x87b3: je       0x8840
0x87b9: mov      dword ptr [rsp + 8], edi
0x87bd: call     0x4f00
0x87c2: mov      r8, rax
0x87c5: movsxd   rax, dword ptr [rsp + 8]
0x87ca: mov      rdx, qword ptr [r8]
0x87cd: mov      rdi, rax
0x87d0: test     byte ptr [rdx + rax*2 + 1], 0x40
0x87d5: jne      0x8948
0x87db: xor      eax, eax
0x87dd: xor      r12d, r12d
0x87e0: call     0xdea0
0x87e5: xor      eax, eax
0x87e7: call     0xd870
0x87ec: xor      eax, eax
0x87ee: call     0xd740
0x87f3: xor      eax, eax
0x87f5: call     0xa820
0x87fa: mov      rdi, r13
0x87fd: call     0xdb10
0x8802: jmp      0x8867
0x8804: nop      dword ptr [rax]
0x8808: mov      rcx, qword ptr [rax]
0x880b: sub      rcx, r15
0x880e: mov      rax, rcx

```

```

0x8811: sar    rcx, 0x3f
0x8815: imul   r14
0x8818: sar    rdx, 1
0x881b: sub    rdx, rcx
0x881e: mov    edi, edx
0x8820: not    edi
0x8822: cmp    edx, 7
0x8825: jne    0x874d
0x882b: xor    edi, edi
0x882d: xor    eax, eax
0x882f: call   0x7bd0
0x8834: jmp    0x8720
0x8839: nop     dword ptr [rax]
0x8840: xor    eax, eax
0x8842: call   0xde0
0x8847: xor    eax, eax
0x8849: call   0xd870
0x884e: xor    eax, eax
0x8850: call   0xd740
0x8855: xor    eax, eax
0x8857: call   0xa820
0x885c: mov    rdi, r13
0x885f: call   0xdb10
0x8864: xor    r12d, r12d
0x8867: add    rsp, 0x18
0x886b: mov    rax, r12
0x886e: pop    rbx
0x886f: pop    rbp
0x8870: pop    r12
0x8872: pop    r13
0x8874: pop    r14
0x8876: pop    r15
0x8878: ret
0x8879: nop     dword ptr [rax]
0x8880: xor    eax, eax
0x8882: call   0xa5a0
0x8887: jmp    0x877a
0x888c: nop     dword ptr [rax]
0x8890: xor    eax, eax
0x8892: mov    qword ptr [rsp + 8], rdi
0x8897: call   0xc8c0
0x889c: mov    qword ptr [rip + 0x2205d], rax
0x88a3: xor    eax, eax
0x88a5: call   0xcff0
0x88aa: mov    rdi, qword ptr [rsp + 8]
0x88af: test   rdi, rdi
0x88b2: je     0x8864
0x88b4: call   0xd540

```

```

0x88b9: jmp      0x8864
0x88bb: cmp      edi, -0x44
0x88be: jne      0x88df
0x88c0: cmp      ebx, 1
0x88c3: jne      0x88df
0x88c5: xor      eax, eax
0x88c7: call     0xca40
0x88cc: cmp      eax, 1
0x88cf: mov      eax, 2
0x88d4: sbb      eax, -1
0x88d7: mov      dword ptr [rbp], eax
0x88da: jmp      0x877a
0x88df: cmp      edi, -0x45
0x88e2: jne      0x87b9
0x88e8: cmp      ebx, 2
0x88eb: je       0x88c5
0x88ed: jmp      0x87b9
0x88f2: nop      word ptr [rax + rax]
0x88f8: mov      rdi, qword ptr [rip + 0x22001]
0x88ff: xor      r12d, r12d
0x8902: call     0xc970
0x8907: xor      eax, eax
0x8909: call     0xd870
0x890e: xor      eax, eax
0x8910: call     0xd740
0x8915: xor      eax, eax
0x8917: call     0xa820
0x891c: jmp      0x8867
0x8921: nop      dword ptr [rax]
0x8928: xor      eax, eax
0x892a: call     0xdea0
0x892f: xor      eax, eax
0x8931: call     0xd740
0x8936: xor      eax, eax
0x8938: call     0xa820
0x893d: jmp      0x8867
0x8942: nop      word ptr [rax + rax]
0x8948: call     0xcd30
0x894d: mov      dword ptr [rbp], 3
0x8954: jmp      0x877a
0x8959: nop      dword ptr [rax]
0x8960: endbr64
0x8964: mov      rdi, qword ptr [rip + 0x21f2d]
0x896b: push     rbx
0x896c: test     rdi, rdi
0x896f: je       0x89ac
0x8971: mov      eax, dword ptr [rip + 0x21f1d]
0x8977: test     eax, eax

```



```

0x8979: jle      0x899c
0x897b: xor      ebx, ebx
0x897d: nop      dword ptr [rax]
0x8980: mov      rdi, qword ptr [rdi + rbx*8]
0x8984: add      rbx, 1
0x8988: call     0x18d10
0x898d: cmp      dword ptr [rip + 0x21f01], ebx
0x8993: mov      rdi, qword ptr [rip + 0x21efe]
0x899a: jg       0x8980
0x899c: call     0x18d10
0x89a1: mov      qword ptr [rip + 0x21eec], 0
0x89ac: mov      dword ptr [rip + 0x21ede], 0
0x89b6: pop      rbx
0x89b7: mov      dword ptr [rip + 0x21ecf], 0
0x89c1: ret
0x89c2: nop      word ptr cs:[rax + rax]
0x89cd: nop      dword ptr [rax]
0x89d0: endbr64
0x89d4: push     rbp
0x89d5: mov      rbp, rdi
0x89d8: push     rbx
0x89d9: sub      rsp, 8
0x89dd: mov      esi, dword ptr [rip + 0x21eb1]
0x89e3: mov      rdi, qword ptr [rip + 0x21eae]
0x89ea: mov      dword ptr [rip + 0x21ea0], esi
0x89f0: add      esi, 1
0x89f3: mov      dword ptr [rip + 0x21e9b], esi
0x89f9: movsxd   rsi, esi
0x89fc: shl      rsi, 3
0x8a00: call     0x1d690
0x8a05: movsxd   rdx, dword ptr [rip + 0x21e84]
0x8a0c: mov      rdi, rbp
0x8a0f: mov      qword ptr [rip + 0x21e82], rax
0x8a16: lea      rbx, [rax + rdx*8]
0x8a1a: call     0x1d7a0
0x8a1f: mov      qword ptr [rbx], rax
0x8a22: add      rsp, 8
0x8a26: pop      rbx
0x8a27: pop      rbp
0x8a28: ret
0x8a29: nop      dword ptr [rax]
0x8a30: endbr64
0x8a34: mov      rdx, rsi
0x8a37: mov      esi, dword ptr [rip + 0x21e57]
0x8a3d: mov      eax, dword ptr [rip + 0x21e4d]
0x8a43: lea      ecx, [rsi - 1]
0x8a46: cmp      ecx, eax
0x8a48: jle      0x8a70

```

```

0x8a4a: add     eax, 1
0x8a4d: mov     r8d, 1
0x8a53: mov     esi, 0x21
0x8a58: mov     rcx, qword ptr [rip + 0x21e39]
0x8a5f: mov     dword ptr [rip + 0x21e2b], eax
0x8a65: cdqe
0x8a67: mov     rcx, qword ptr [rcx + rax*8]
0x8a6b: jmp     0x13e30
0x8a70: xor     eax, eax
0x8a72: jmp     0xa5a0
0x8a77: nop     word ptr [rax + rax]
0x8a80: endbr64
0x8a84: mov     eax, dword ptr [rip + 0x21e06]
0x8a8a: mov     rdx, rsi
0x8a8d: test    eax, eax
0x8a8f: je      0x8ac0
0x8a91: sub     eax, 1
0x8a94: mov     r8d, 1
0x8a9a: mov     esi, 0x21
0x8a9f: mov     rcx, qword ptr [rip + 0x21df2]
0x8aa6: mov     dword ptr [rip + 0x21de4], eax
0x8aac: cdqe
0x8aae: mov     rcx, qword ptr [rcx + rax*8]
0x8ab2: jmp     0x13e30
0x8ab7: nop     word ptr [rax + rax]
0x8ac0: xor     eax, eax
0x8ac2: jmp     0xa5a0
0x8ac7: nop     word ptr [rax + rax]
0x8ad0: endbr64
0x8ad4: sub     rsp, 8
0x8ad8: xor     edi, edi
0x8ada: call    0xbbc0
0x8adf: xor     eax, eax
0x8ae1: call    0xde40
0x8ae6: xor     eax, eax
0x8ae8: add     rsp, 8
0x8aec: jmp     0x19580
0x8af1: nop     word ptr cs:[rax + rax]
0x8afc: nop     dword ptr [rax]
0x8b00: endbr64
0x8b04: push    rbx
0x8b05: xor     eax, eax
0x8b07: mov     rbx, rdi
0x8b0a: call    0x9c90
0x8b0f: cmp     eax, 1
0x8b12: je      0x8b42
0x8b14: mov     rdi, qword ptr [rip + 0x21d65]
0x8b1b: mov     rcx, qword ptr [rip + 0x21dc6]

```

```

0x8b22: mov     r8, rbx
0x8b25: xor     eax, eax
0x8b27: lea     rdx, [rip + 0x179e8]
0x8b2e: mov     esi, 1
0x8b33: call    0x4e60
0x8b38: mov     edi, 1
0x8b3d: call    0x4e20
0x8b42: xor     eax, eax
0x8b44: call    0x8ad0
0x8b49: jmp     0x8b14
0x8b4b: nop     dword ptr [rax + rax]
0x8b50: endbr64
0x8b54: push    r15
0x8b56: push    r14
0x8b58: push    r13
0x8b5a: push    r12
0x8b5c: mov     r12, rsi
0x8b5f: push    rbp
0x8b60: push    rbx
0x8b61: sub     rsp, 0x68
0x8b65: mov     r14, qword ptr [rdi + 8]
0x8b69: mov     qword ptr [rsp + 0x18], rdi
0x8b6e: mov     qword ptr [rsp + 8], rdx
0x8b73: mov     rdi, r14
0x8b76: mov     qword ptr [rsp + 0x10], rcx
0x8b7b: mov     rax, qword ptr fs:[0x28]
0x8b84: mov     qword ptr [rsp + 0x58], rax
0x8b89: xor     eax, eax
0x8b8b: mov     qword ptr [rsp + 0x48], 0
0x8b94: mov     qword ptr [rsp + 0x50], 0
0x8b9d: call    0x49b0
0x8ba2: lea     r13, [rax + 1]
0x8ba6: mov     rdi, r13
0x8ba9: call    0x1d670
0x8bae: mov     qword ptr [r12], rax
0x8bb2: mov     rbx, rax
0x8bb5: movzx   eax, byte ptr [r14]
0x8bb9: test    al, al
0x8bbb: je      0x925c
0x8bc1: mov     dword ptr [rsp + 0x24], 0
0x8bc9: jmp     0x8be5
0x8bcb: nop     dword ptr [rax + rax]
0x8bd0: mov     byte ptr [rbx], al
0x8bd2: mov     r14, rbp
0x8bd5: add     rbx, 1
0x8bd9: movzx   eax, byte ptr [r14]
0x8bdd: test    al, al
0x8bdf: je      0x8d38

```

```

0x8be5: lea      rbp, [r14 + 1]
0x8be9: cmp      al, 0x25
0x8beb: jne      0x8bd0
0x8bed: call     0x4f00
0x8bf2: movsx    rcx, byte ptr [r14 + 1]
0x8bf7: mov      r15, qword ptr [rsp + 0x10]
0x8bfc: mov      rax, qword ptr [rax]
0x8bff: mov      rdx, rcx
0x8c02: test     byte ptr [rax + rcx*2 + 1], 2
0x8c07: lea      eax, [rcx - 0x3f]
0x8c0a: cmovne   r15, qword ptr [rsp + 8]
0x8c10: cmp      al, 0x3b
0x8c12: ja       0x8c38
0x8c14: lea      rcx, [rip + 0x174f1]
0x8c1b: movzx    eax, al
0x8c1e: movsxd   rax, dword ptr [rcx + rax*4]
0x8c22: add      rax, rcx
0x8c25: jmp      rax
0x8c28: mov      dword ptr [rip + 0x21cc6], 0
0x8c32: nop      word ptr [rax + rax]
0x8c38: mov      rdi, qword ptr [r12]
0x8c3c: call     0x18d10
0x8c41: mov      qword ptr [r12], 0
0x8c49: xor      eax, eax
0x8c4b: mov      rdx, qword ptr [rsp + 0x58]
0x8c50: sub      rdx, qword ptr fs:[0x28]
0x8c59: jne      0x929f
0x8c5f: add      rsp, 0x68
0x8c63: pop      rbx
0x8c64: pop      rbp
0x8c65: pop      r12
0x8c67: pop      r13
0x8c69: pop      r14
0x8c6b: pop      r15
0x8c6d: ret
0x8c6e: nop
0x8c70: mov      rdi, r15
0x8c73: call     0xf510
0x8c78: cmp      eax, 1
0x8c7b: jne      0x8d61
0x8c81: nop      dword ptr [rax]
0x8c88: mov      rdi, r15
0x8c8b: call     0xf490
0x8c90: mov      r14, rax
0x8c93: mov      rdi, r14
0x8c96: call     0x49b0
0x8c9b: lea      rdi, [rax + 3]
0x8c9f: call     0x1d670

```

```

0x8ca4: mov     r8, r14
0x8ca7: mov     qword ptr [rsp + 0x50], rax
0x8cac: mov     rdi, rax
0x8caf: lea     rcx, [rip + 0x185df]
0x8cb6: mov     rdx, -1
0x8cbd: mov     esi, 1
0x8cc2: xor     eax, eax
0x8cc4: call    0x4f10
0x8cc9: mov     byte ptr [rbx], 0
0x8ccc: mov     rdi, qword ptr [rsp + 0x50]
0x8cd1: lea     r14, [rbp + 1]
0x8cd5: test    rdi, rdi
0x8cd8: je      0x8bd9
0x8cde: call    0x49b0
0x8ce3: mov     rdi, qword ptr [r12]
0x8ce7: add     r13, rax
0x8cea: mov     rsi, r13
0x8ced: call    0x1d690
0x8cf2: mov     rsi, qword ptr [rsp + 0x50]
0x8cf7: mov     qword ptr [r12], rax
0x8cfb: mov     rdi, rax
0x8cfe: call    0x4df0
0x8d03: mov     rbx, qword ptr [r12]
0x8d07: mov     rdi, rbx
0x8d0a: call    0x49b0
0x8d0f: mov     rdi, qword ptr [rsp + 0x50]
0x8d14: add     rbx, rax
0x8d17: call    0x18d10
0x8d1c: mov     qword ptr [rsp + 0x50], 0
0x8d25: movzx   eax, byte ptr [r14]
0x8d29: test    al, al
0x8d2b: jne     0x8be5
0x8d31: nop     dword ptr [rax]
0x8d38: mov     eax, dword ptr [rsp + 0x24]
0x8d3c: add     eax, 1
0x8d3f: mov     byte ptr [rbx], 0
0x8d42: jmp     0x8c4b
0x8d47: nop     word ptr [rax + rax]
0x8d50: mov     rdi, r15
0x8d53: call    0xf510
0x8d58: cmp     eax, 4
0x8d5b: je      0x8c88
0x8d61: mov     rdi, qword ptr [rsp + 0x50]
0x8d66: test    rdi, rdi
0x8d69: je      0x8d70
0x8d6b: call    0x18d10
0x8d70: mov     qword ptr [r12], 0
0x8d78: mov     eax, 0xffffffff

```

```

0x8d7d: jmp      0x8c4b
0x8d82: nop      word ptr [rax + rax]
0x8d88: mov      rdi, r15
0x8d8b: call     0xf510
0x8d90: cmp      eax, 3
0x8d93: je       0x8c88
0x8d99: jmp      0x8d61
0x8d9b: nop      dword ptr [rax + rax]
0x8da0: mov      rdi, qword ptr [r15 + 0x18]
0x8da4: call     0x49b0
0x8da9: lea      rdi, [rax + 3]
0x8dad: call     0x1d670
0x8db2: mov      r8, qword ptr [r15 + 0x18]
0x8db6: mov      qword ptr [rsp + 0x50], rax
0x8dbb: mov      rdi, rax
0x8dbe: jmp      0x8caf
0x8dc3: nop      dword ptr [rax + rax]
0x8dc8: mov      rdi, r15
0x8dcb: call     0xf4b0
0x8dd0: mov      edi, eax
0x8dd2: mov      r14d, eax
0x8dd5: call     0x4970
0x8dda: test     rax, rax
0x8ddd: je       0x8eab
0x8de3: mov      rdi, qword ptr [rax]
0x8de6: call     0x1d7a0
0x8deb: mov      qword ptr [rsp + 0x50], rax
0x8df0: jmp      0x8cc9
0x8df5: nop      dword ptr [rax]
0x8df8: mov      edi, 0x10
0x8dfd: call     0x1d670
0x8e02: mov      rdi, r15
0x8e05: mov      qword ptr [rsp + 0x50], rax
0x8e0a: call     0xf4f0
0x8e0f: mov      rdi, qword ptr [rsp + 0x50]
0x8e14: mov      esi, 1
0x8e19: lea      rcx, [rip + 0x1771c]
0x8e20: and      eax, 0xfff
0x8e25: mov      rdx, -1
0x8e2c: mov      r8d, eax
0x8e2f: xor      eax, eax
0x8e31: call     0x4f10
0x8e36: jmp      0x8cc9
0x8e3b: nop      dword ptr [rax + rax]
0x8e40: xor      eax, eax
0x8e42: cmp      dl, 0x69
0x8e45: mov      edx, dword ptr [rip + 0x21aad]
0x8e4b: setne    al

```

```

0x8e4e: add     eax, 1
0x8e51: mov     dword ptr [rsp + 0x24], eax
0x8e55: test    edx, edx
0x8e57: je      0x9198
0x8e5d: mov     eax, dword ptr [r15 + 0x70]
0x8e61: test    eax, eax
0x8e63: je      0x9198
0x8e69: mov     edi, 0x15
0x8e6e: call    0x1d670
0x8e73: movdqa  xmm0, xmmword ptr [rip + 0x17d35]
0x8e7b: mov     qword ptr [rsp + 0x50], rax
0x8e80: mov     byte ptr [rax + 0x10], 0
0x8e84: movups  xmmword ptr [rax], xmm0
0x8e87: jmp     0x8cc9
0x8e8c: nop     dword ptr [rax]
0x8e90: mov     rdi, r15
0x8e93: call    0xf4d0
0x8e98: mov     edi, eax
0x8e9a: mov     r14d, eax
0x8e9d: call    0x4a10
0x8ea2: test    rax, rax
0x8ea5: jne     0x8de3
0x8eab: mov     edi, 0x10
0x8eb0: call    0x1d670
0x8eb5: mov     r8d, r14d
0x8eb8: mov     esi, 1
0x8ebd: lea     rcx, [rip + 0x17678]
0x8ec4: mov     qword ptr [rsp + 0x50], rax
0x8ec9: mov     rdi, rax
0x8ecc: mov     rdx, -1
0x8ed3: xor     eax, eax
0x8ed5: call    0x4f10
0x8eda: jmp     0x8cc9
0x8edf: nop
0x8ee0: mov     rdi, r15
0x8ee3: call    0xf510
0x8ee8: cmp     eax, 2
0x8eeb: je      0x8c88
0x8ef1: jmp     0x8d61
0x8ef6: nop     word ptr cs:[rax + rax]
0x8f00: mov     rdi, r15
0x8f03: call    0xf510
0x8f08: test    eax, eax
0x8f0a: je      0x8c88
0x8f10: jmp     0x8d61
0x8f15: nop     dword ptr [rax]
0x8f18: mov     rdi, qword ptr [r15 + 0x18]
0x8f1c: mov     esi, 0x2f

```

```

0x8f21: lea     r14, [rip + 0x17604]
0x8f28: call    0x4a20
0x8f2d: mov     edi, 8
0x8f32: cmp     byte ptr [rax + 1], 0
0x8f36: je      0x8c9f
0x8f3c: lea     r14, [rax + 1]
0x8f40: jmp     0x8c93
0x8f45: mov     esi, dword ptr [rip + 0x219ad]
0x8f4b: test    esi, esi
0x8f4d: jne     0x8c28
0x8f53: cmp     byte ptr [r14 + 2], 0x7b
0x8f58: jne     0x8c38
0x8f5e: add     r14, 3
0x8f62: mov     esi, 0x7d
0x8f67: mov     rdi, r14
0x8f6a: call    0x4a00
0x8f6f: mov     rbp, rax
0x8f72: test    rax, rax
0x8f75: je      0x8c38
0x8f7b: mov     byte ptr [rax], 0
0x8f7e: mov     edx, 2
0x8f83: lea     rsi, [rip + 0x175a8]
0x8f8a: mov     rdi, r14
0x8f8d: call    0x8070
0x8f92: mov     byte ptr [rbp], 0x7d
0x8f96: cmp     al, 0x79
0x8f98: je      0x8cc9
0x8f9e: jmp     0x8d61
0x8fa3: nop     dword ptr [rax + rax]
0x8fa8: mov     ecx, dword ptr [rip + 0x2194a]
0x8fae: test    ecx, ecx
0x8fb0: jne     0x8c28
0x8fb6: cmp     byte ptr [r14 + 2], 0x7b
0x8fbb: jne     0x8c38
0x8fc1: add     r14, 3
0x8fc5: mov     esi, 0x2c
0x8fca: mov     rdi, r14
0x8fcd: call    0x4a00
0x8fd2: mov     rbp, rax
0x8fd5: test    rax, rax
0x8fd8: je      0x8c38
0x8fde: mov     byte ptr [rax], 0
0x8fe1: mov     rax, qword ptr [rsp + 0x18]
0x8fe6: lea     r15, [rsp + 0x48]
0x8feb: mov     rcx, qword ptr [rsp + 0x10]
0x8ff0: mov     rdx, qword ptr [rsp + 8]
0x8ff5: mov     rsi, r15
0x8ff8: mov     dword ptr [rip + 0x218f6], 1

```



```

0x9002: mov     r8, qword ptr [rax + 8]
0x9006: mov     qword ptr [rax + 8], r14
0x900a: mov     rdi, rax
0x900d: mov     r14, rax
0x9010: xor     eax, eax
0x9012: mov     qword ptr [rsp + 0x28], r8
0x9017: call    0x8b50
0x901c: mov     r8, qword ptr [rsp + 0x28]
0x9021: mov     dword ptr [rip + 0x218cd], 0
0x902b: mov     qword ptr [r14 + 8], r8
0x902f: test    eax, eax
0x9031: jle     0x9290
0x9037: mov     rax, qword ptr [rsp + 0x18]
0x903c: mov     rdi, qword ptr [rax]
0x903f: call    0x49b0
0x9044: mov     r14, qword ptr [rsp + 0x48]
0x9049: mov     qword ptr [rsp + 0x28], rax
0x904e: mov     rdi, r14
0x9051: call    0x49b0
0x9056: mov     rdx, qword ptr [rsp + 0x28]
0x905b: lea     rdi, [rdx + rax + 0x11]
0x9060: call    0x1d670
0x9065: mov     r9, r14
0x9068: mov     esi, 1
0x906d: lea     rcx, [rip + 0x174c1]
0x9074: mov     r10, rax
0x9077: mov     rax, qword ptr [rsp + 0x18]
0x907c: mov     rdx, -1
0x9083: mov     rdi, r10
0x9086: mov     qword ptr [rsp + 0x28], r10
0x908b: mov     r8, qword ptr [rax]
0x908e: xor     eax, eax
0x9090: call    0x4f10
0x9095: mov     rdi, r14
0x9098: lea     r14, [rbp + 1]
0x909c: call    0x18d10
0x90a1: mov     byte ptr [rbp], 0x2c
0x90a5: mov     esi, 0x7d
0x90aa: mov     rdi, r14
0x90ad: mov     qword ptr [rsp + 0x48], 0
0x90b6: call    0x4a00
0x90bb: mov     rbp, rax
0x90be: test    rax, rax
0x90c1: je      0x8c38
0x90c7: mov     byte ptr [rax], 0
0x90ca: mov     rdi, qword ptr [rsp + 0x28]
0x90cf: call    0x49b0
0x90d4: mov     r10, qword ptr [rsp + 0x28]

```

```

0x90d9: cmp     rax, 0x32
0x90dd: jbe     0x90e4
0x90df: mov     byte ptr [r10 + 0x32], 0
0x90e4: mov     rax, qword ptr [rsp + 0x18]
0x90e9: mov     rcx, qword ptr [rsp + 0x10]
0x90ee: mov     rsi, r15
0x90f1: mov     qword ptr [rsp + 0x30], r10
0x90f6: mov     dword ptr [rip + 0x217f8], 1
0x9100: mov     rdx, qword ptr [rsp + 8]
0x9105: mov     r8, qword ptr [rax + 8]
0x9109: mov     qword ptr [rax + 8], r14
0x910d: mov     r15, rax
0x9110: mov     rdi, rax
0x9113: xor     eax, eax
0x9115: mov     qword ptr [rsp + 0x28], r8
0x911a: call   0x8b50
0x911f: mov     r8, qword ptr [rsp + 0x28]
0x9124: mov     r10, qword ptr [rsp + 0x30]
0x9129: mov     dword ptr [rip + 0x217c5], 0
0x9133: test    eax, eax
0x9135: mov     qword ptr [r15 + 8], r8
0x9139: jle     0x9266
0x913f: mov     rax, qword ptr [rsp + 0x18]
0x9144: mov     r15, qword ptr [rsp + 0x48]
0x9149: mov     rdi, r10
0x914c: lea     rsi, [rsp + 0x50]
0x9151: mov     qword ptr [rsp + 0x28], r10
0x9156: mov     rcx, qword ptr [rax + 0x28]
0x915a: mov     rdx, r15
0x915d: call   0x83e0
0x9162: mov     rdi, qword ptr [rsp + 0x28]
0x9167: mov     r14, rax
0x916a: call   0x18d10
0x916f: mov     rdi, r15
0x9172: call   0x18d10
0x9177: mov     byte ptr [rbp], 0x7d
0x917b: mov     qword ptr [rsp + 0x48], 0
0x9184: test    r14, r14
0x9187: jne     0x8cc9
0x918d: jmp     0x8d61
0x9192: nop     word ptr [rax + rax]
0x9198: mov     qword ptr [rsp + 0x50], 0
0x91a1: mov     rdi, r15
0x91a4: xor     r14d, r14d
0x91a7: call   0x100c0
0x91ac: mov     qword ptr [rsp + 0x28], rbx
0x91b1: mov     qword ptr [rsp + 0x30], rbp
0x91b6: mov     qword ptr [rsp + 0x38], r12

```

```

0x91bb: jmp      0x9232
0x91bd: nop
0x91c0: cdqe
0x91c2: lea      rcx, [rax + rax*4]
0x91c6: mov      rax, qword ptr [r15 + 0x38]
0x91ca: shl      rcx, 4
0x91ce: mov      rdi, qword ptr [rax + rcx]
0x91d2: mov      r12, rcx
0x91d5: call     0x49b0
0x91da: mov      rdi, qword ptr [rsp + 0x50]
0x91df: lea      rbp, [rax + r14]
0x91e3: lea      rsi, [rbp + 4]
0x91e7: lea      rbx, [rbp + 3]
0x91eb: call     0x1d690
0x91f0: mov      qword ptr [rsp + 0x50], rax
0x91f5: mov      byte ptr [rax + r14], 0x22
0x91fa: mov      rax, qword ptr [r15 + 0x38]
0x91fe: mov      rsi, qword ptr [rax + r12]
0x9202: mov      rax, qword ptr [rsp + 0x50]
0x9207: lea      rdi, [rax + r14 + 1]
0x920c: mov      r14, rbx
0x920f: call     0x48b0
0x9214: mov      rax, qword ptr [rsp + 0x50]
0x9219: mov      byte ptr [rax + rbp + 1], 0x22
0x921e: mov      rax, qword ptr [rsp + 0x50]
0x9223: mov      byte ptr [rax + rbp + 2], 0x20
0x9228: mov      rax, qword ptr [rsp + 0x50]
0x922d: mov      byte ptr [rax + rbp + 3], 0
0x9232: mov      rdi, r15
0x9235: call     0x100e0
0x923a: cmp      eax, -1
0x923d: jne      0x91c0
0x923f: mov      rbx, qword ptr [rsp + 0x28]
0x9244: mov      rbp, qword ptr [rsp + 0x30]
0x9249: mov      r12, qword ptr [rsp + 0x38]
0x924e: test     r14, r14
0x9251: jne      0x8cc9
0x9257: jmp      0x8d61
0x925c: mov      eax, 1
0x9261: jmp      0x8d3f
0x9266: mov      byte ptr [rbp], 0x7d
0x926a: mov      rdi, r10
0x926d: mov      dword ptr [rsp + 8], eax
0x9271: call     0x18d10
0x9276: mov      eax, dword ptr [rsp + 8]
0x927a: test     eax, eax
0x927c: je       0x8c38
0x9282: jmp      0x8d61

```

```

0x9287: nop      word ptr [rax + rax]
0x9290: mov      byte ptr [rbp], 0x2c
0x9294: je        0x8c38
0x929a: jmp      0x8d61
0x929f: call     0x49d0
0x92a4: nop      word ptr cs:[rax + rax]
0x92af: nop
0x92b0: endbr64
0x92b4: push     r13
0x92b6: mov      r13, rdx
0x92b9: push     r12
0x92bb: mov      r12, rsi
0x92be: push     rbp
0x92bf: mov      rbp, rdi
0x92c2: call     0x47e0
0x92c7: test     rax, rax
0x92ca: je        0x92d8
0x92cc: pop      rbp
0x92cd: pop      r12
0x92cf: pop      r13
0x92d1: ret
0x92d2: nop      word ptr [rax + rax]
0x92d8: test     r12, r12
0x92db: je        0x9300
0x92dd: mov      rdi, r12
0x92e0: call     0x47e0
0x92e5: mov      rsi, rax
0x92e8: test     rax, rax
0x92eb: je        0x9300
0x92ed: mov      rdi, rbp
0x92f0: pop      rbp
0x92f1: pop      r12
0x92f3: pop      r13
0x92f5: jmp      0x18380
0x92fa: nop      word ptr [rax + rax]
0x9300: mov      rsi, r13
0x9303: mov      rdi, rbp
0x9306: pop      rbp
0x9307: pop      r12
0x9309: pop      r13
0x930b: jmp      0x18380
0x9310: endbr64
0x9314: push     r15
0x9316: push     r14
0x9318: push     r13
0x931a: push     r12
0x931c: push     rbp
0x931d: push     rbx

```

```

0x931e: mov     ebx, edi
0x9320: sub     rsp, 0xb8
0x9327: mov     qword ptr [rsp + 0x18], rsi
0x932c: mov     rax, qword ptr fs:[0x28]
0x9335: mov     qword ptr [rsp + 0xa8], rax
0x933d: xor     eax, eax
0x933f: mov     dword ptr [rsi], 0
0x9345: cmp     edi, 0x7ff
0x934b: jg      0x95d8
0x9351: lea     rax, [rsp + 0x20]
0x9356: lea     r12, [rsp + 0x50]
0x935b: mov     qword ptr [rsp + 8], rax
0x9360: lea     r13, [rip + 0x2089e]
0x9367: jmp     0x94df
0x936c: nop     dword ptr [rax]
0x9370: mov     rdi, r12
0x9373: call    0xbd80
0x9378: test    rax, rax
0x937b: je      0x9508
0x9381: cmp     byte ptr [rax], 0
0x9384: je      0x94d0
0x938a: mov     edx, 0x50
0x938f: mov     rsi, rax
0x9392: mov     rdi, r12
0x9395: call    0x4cf0
0x939a: mov     dword ptr [rsp + 4], 0
0x93a2: mov     edi, 1
0x93a7: mov     esi, 0x30
0x93ac: call    0x1d740
0x93b1: mov     rdi, qword ptr [rsp + 0x20]
0x93b6: mov     rbp, rax
0x93b9: test    rdi, rdi
0x93bc: je      0x9560
0x93c2: call    0x1d7a0
0x93c7: mov     edi, 8
0x93cc: mov     qword ptr [rbp], rax
0x93d0: call    0x19100
0x93d5: mov     rdi, qword ptr [rsp + 0x30]
0x93da: mov     qword ptr [rbp + 0x28], rax
0x93de: test    rdi, rdi
0x93e1: je      0x93ec
0x93e3: call    0x1d7a0
0x93e8: mov     qword ptr [rbp + 0x10], rax
0x93ec: mov     rdi, qword ptr [rsp + 0x28]
0x93f1: test    rdi, rdi
0x93f4: je      0x9474
0x93f6: call    0x1d7a0
0x93fb: mov     qword ptr [rbp + 8], rax

```

```

0x93ff: mov     r14, qword ptr [rsp + 0x38]
0x9404: mov     eax, 1
0x9409: test    r14, r14
0x940c: je      0x9421
0x940e: call    0x4ef0
0x9413: movsx   rdx, byte ptr [r14]
0x9417: mov     rax, qword ptr [rax]
0x941a: cmp     dword ptr [rax + rdx*4], 0x79
0x941e: sete    al
0x9421: mov     byte ptr [rbp + 0x18], al
0x9424: mov     rdx, qword ptr [rsp + 0x40]
0x9429: test    rdx, rdx
0x942c: je      0x944c
0x942e: mov     qword ptr [rsp + 0x10], rdx
0x9433: call    0x4ef0
0x9438: mov     rdx, qword ptr [rsp + 0x10]
0x943d: mov     rax, qword ptr [rax]
0x9440: movsx   rdx, byte ptr [rdx]
0x9444: cmp     dword ptr [rax + rdx*4], 0x79
0x9448: sete    byte ptr [rbp + 0x19]
0x944c: mov     rdx, qword ptr [rsp + 0x48]
0x9451: test    rdx, rdx
0x9454: je      0x9474
0x9456: mov     qword ptr [rsp + 0x10], rdx
0x945b: call    0x4ef0
0x9460: mov     rdx, qword ptr [rsp + 0x10]
0x9465: mov     rax, qword ptr [rax]
0x9468: movsx   rdx, byte ptr [rdx]
0x946c: cmp     dword ptr [rax + rdx*4], 0x79
0x9470: sete    byte ptr [rbp + 0x1a]
0x9474: call    0x4f00
0x9479: mov     r15, qword ptr [rbp]
0x947d: mov     rax, qword ptr [rax]
0x9480: movsx   rdx, byte ptr [r15]
0x9484: test    byte ptr [rax + rdx*2 + 1], 2
0x9489: jne     0x9518
0x948f: mov     rdi, r12
0x9492: call    0x1d7a0
0x9497: cmp     byte ptr [rbp + 0x1b], 0
0x949b: mov     qword ptr [rbp + 0x20], rax
0x949f: jne     0x94ac
0x94a1: cmp     qword ptr [rbp + 8], 0
0x94a6: je      0x9590
0x94ac: mov     eax, dword ptr [rsp + 4]
0x94b0: test    eax, eax
0x94b2: je      0x94c5
0x94b4: mov     rdi, r12
0x94b7: call    0x9f00

```

```

0x94bc: test    rax, rax
0x94bf: je      0x95a0
0x94c5: mov     rsi, rbp
0x94c8: mov     rdi, r12
0x94cb: call    0xaf70
0x94d0: add     ebx, 1
0x94d3: cmp     ebx, 0x800
0x94d9: je      0x9580
0x94df: mov     rsi, qword ptr [rsp + 8]
0x94e4: xor     ecx, ecx
0x94e6: mov     edx, 6
0x94eb: mov     rdi, r12
0x94ee: call    0x17320
0x94f3: movzx   eax, byte ptr [rsp + 0x50]
0x94f8: test    al, al
0x94fa: je      0x95d8
0x9500: cmp     al, 0x5e
0x9502: jne     0x9370
0x9508: mov     dword ptr [rsp + 4], 1
0x9510: jmp     0x93a2
0x9515: nop     dword ptr [rax]
0x9518: lea     r14, [rip + 0x1fb01]
0x951f: jmp     0x9535
0x9521: nop     dword ptr [rax]
0x9528: add     r14, 0x23
0x952c: cmp     r13, r14
0x952f: je      0x948f
0x9535: mov     rsi, r14
0x9538: mov     rdi, r15
0x953b: call    0x4b30
0x9540: test    eax, eax
0x9542: jne     0x9528
0x9544: mov     rdi, r15
0x9547: call    0x18d10
0x954c: mov     qword ptr [rbp], r14
0x9550: mov     byte ptr [rbp + 0x1b], 1
0x9554: jmp     0x948f
0x9559: nop     dword ptr [rax]
0x9560: mov     rdi, rax
0x9563: add     ebx, 1
0x9566: call    0x18d10
0x956b: cmp     ebx, 0x800
0x9571: jne     0x94df
0x9577: nop     word ptr [rax + rax]
0x9580: mov     eax, 0x800
0x9585: jmp     0x95da
0x9587: nop     word ptr [rax + rax]
0x9590: cmp     qword ptr [rbp + 0x10], 0

```

```

0x9595: jne      0x94ac
0x959b: jmp      0x94d0
0x95a0: mov     rcx, qword ptr [rip + 0x21341]
0x95a7: mov     rdi, qword ptr [rip + 0x212d2]
0x95ae: mov     r8, r12
0x95b1: xor     eax, eax
0x95b3: lea     rdx, [rip + 0x1720e]
0x95ba: mov     esi, 1
0x95bf: call    0x4e60
0x95c4: mov     rax, qword ptr [rsp + 0x18]
0x95c9: add     dword ptr [rax], 1
0x95cc: jmp     0x94d0
0x95d1: nop     dword ptr [rax]
0x95d8: mov     eax, ebx
0x95da: mov     rdx, qword ptr [rsp + 0xa8]
0x95e2: sub     rdx, qword ptr fs:[0x28]
0x95eb: jne     0x95ff
0x95ed: add     rsp, 0xb8
0x95f4: pop     rbx
0x95f5: pop     rbp
0x95f6: pop     r12
0x95f8: pop     r13
0x95fa: pop     r14
0x95fc: pop     r15
0x95fe: ret
0x95ff: call    0x49d0
0x9604: nop     word ptr cs:[rax + rax]
0x960f: nop
0x9610: endbr64
0x9614: sub     rsp, 8
0x9618: xor     edi, edi
0x961a: call    0x9b00
0x961f: xor     eax, eax
0x9621: call    0xa6c0
0x9626: mov     rdi, qword ptr [rip + 0x2129b]
0x962d: add     rsp, 8
0x9631: jmp     0xbb40
0x9636: nop     word ptr cs:[rax + rax]
0x9640: endbr64
0x9644: push    rbp
0x9645: lea     rbp, [rip + 0x205b9]
0x964c: push    rbx
0x964d: mov     rdi, rbp
0x9650: sub     rsp, 0x48
0x9654: mov     rax, qword ptr fs:[0x28]
0x965d: mov     qword ptr [rsp + 0x38], rax
0x9662: xor     eax, eax
0x9664: call    0x49b0

```



```

0x9669: mov     rdi, qword ptr [rip + 0x21240]
0x9670: mov     rbx, rax
0x9673: call    0x11100
0x9678: mov     edx, 0x33
0x967d: mov     rsi, rsp
0x9680: sub     edx, ebx
0x9682: mov     rdi, rax
0x9685: call    0x182b0
0x968a: mov     rsi, rbp
0x968d: mov     rdi, rax
0x9690: call    0x4df0
0x9695: mov     rdi, rax
0x9698: call    0xd540
0x969d: mov     rax, qword ptr [rsp + 0x38]
0x96a2: sub     rax, qword ptr fs:[0x28]
0x96ab: jne     0x96b4
0x96ad: add     rsp, 0x48
0x96b1: pop     rbx
0x96b2: pop     rbp
0x96b3: ret
0x96b4: call    0x49d0
0x96b9: nop     dword ptr [rax]
0x96c0: endbr64
0x96c4: sub     rsp, 8
0x96c8: mov     r8d, 1
0x96ce: mov     rdx, qword ptr [rip + 0x211db]
0x96d5: mov     rdi, qword ptr [rip + 0x211cc]
0x96dc: mov     rcx, -1
0x96e3: mov     esi, 0x22
0x96e8: call    0x13e30
0x96ed: mov     r8d, 1
0x96f3: mov     rdx, qword ptr [rip + 0x211ae]
0x96fa: mov     rdi, qword ptr [rip + 0x211af]
0x9701: mov     rcx, -1
0x9708: mov     esi, 0x22
0x970d: add     rsp, 8
0x9711: jmp     0x13e30
0x9716: nop     word ptr cs:[rax + rax]
0x9720: endbr64
0x9724: sub     rsp, 8
0x9728: mov     rdx, qword ptr [rip + 0x211b9]
0x972f: mov     edi, 1
0x9734: xor     eax, eax
0x9736: lea     rsi, [rip + 0x170b3]
0x973d: call    0x4d20
0x9742: lea     rdi, [rip + 0x170cf]
0x9749: call    0x48d0
0x974e: lea     rdi, [rip + 0x170eb]

```

```

0x9755: call    0x48d0
0x975a: lea     rdi, [rip + 0x16dde]
0x9761: call    0x48d0
0x9766: lea     rdi, [rip + 0x170fb]
0x976d: call    0x48d0
0x9772: lea     rdi, [rip + 0x17117]
0x9779: call    0x48d0
0x977e: lea     rdi, [rip + 0x1713b]
0x9785: add     rsp, 8
0x9789: jmp     0x48d0
0x978e: nop
0x9790: cmp     dword ptr [rip + 0x21ba9], 1
0x9797: push    rbx
0x9798: mov     ebx, edi
0x979a: je      0x9830
0x97a0: cmp     ebx, 1
0x97a3: je      0x984a
0x97a9: mov     rdi, qword ptr [rip + 0x20598]
0x97b0: test    rdi, rdi
0x97b3: je      0x97c6
0x97b5: lea     rdx, [rip + 0x2b4]
0x97bc: mov     esi, 1
0x97c1: call    0x4820
0x97c6: movzx   edx, byte ptr [rip + 0x21b83]
0x97cd: mov     dword ptr [rip + 0x21b75], 0
0x97d7: mov     dword ptr [rip + 0x21b67], 0
0x97e1: mov     eax, edx
0x97e3: and     eax, 0x7f
0x97e6: and     edx, 0x40
0x97e9: mov     byte ptr [rip + 0x21b61], al
0x97ef: jne     0x9810
0x97f1: mov     dword ptr [rip + 0x21b45], 1
0x97fb: and     eax, 0x7f
0x97fe: shl     ebx, 7
0x9801: or      eax, ebx
0x9803: mov     byte ptr [rip + 0x21b47], al
0x9809: pop     rbx
0x980a: ret
0x980b: nop     dword ptr [rax + rax]
0x9810: mov     dword ptr [rip + 0x21b2a], 0
0x981a: mov     edi, 1
0x981f: call    0x9870
0x9824: movzx   eax, byte ptr [rip + 0x21b25]
0x982b: jmp     0x97f1
0x982d: nop     dword ptr [rax]
0x9830: movzx   eax, byte ptr [rip + 0x21b19]
0x9837: shr     al, 7
0x983a: movzx   eax, al

```

```

0x983d: cmp     eax, edi
0x983f: je      0x9809
0x9841: cmp     ebx, 1
0x9844: jne     0x97a9
0x984a: mov     rdi, qword ptr [rip + 0x20517]
0x9851: test    rdi, rdi
0x9854: je      0x9824
0x9856: lea     rdx, [rip + 0x213]
0x985d: mov     esi, 1
0x9862: call    0x4820
0x9867: movzx   eax, byte ptr [rip + 0x21ae2]
0x986e: jmp     0x97f1
0x9870: cmp     dword ptr [rip + 0x21acd], 1
0x9877: push    rbx
0x9878: mov     ebx, edi
0x987a: je      0x98e0
0x987c: cmp     ebx, 1
0x987f: je      0x98f6
0x9881: mov     rdi, qword ptr [rip + 0x204c0]
0x9888: test    rdi, rdi
0x988b: je      0x989e
0x988d: lea     rdx, [rip + 0x1dc]
0x9894: mov     esi, 1
0x9899: call    0x4820
0x989e: movzx   eax, byte ptr [rip + 0x21aab]
0x98a5: mov     dword ptr [rip + 0x21a9d], 0
0x98af: mov     dword ptr [rip + 0x21a8f], 0
0x98b9: and     eax, 0xffffffffbf
0x98bc: mov     byte ptr [rip + 0x21a8e], al
0x98c2: test    al, al
0x98c4: js      0x9920
0x98c6: mov     dword ptr [rip + 0x21a74], 1
0x98d0: and     eax, 0xffffffffbf
0x98d3: shl     ebx, 6
0x98d6: or      eax, ebx
0x98d8: mov     byte ptr [rip + 0x21a72], al
0x98de: pop     rbx
0x98df: ret
0x98e0: movzx   eax, byte ptr [rip + 0x21a69]
0x98e7: sar     eax, 6
0x98ea: and     eax, 1
0x98ed: cmp     eax, edi
0x98ef: je      0x98de
0x98f1: cmp     ebx, 1
0x98f4: jne     0x9881
0x98f6: mov     rdi, qword ptr [rip + 0x2048b]
0x98fd: test    rdi, rdi
0x9900: je      0x9934

```

```

0x9902: lea     rdx, [rip + 0x167]
0x9909: mov     esi, 1
0x990e: call    0x4820
0x9913: movzx   eax, byte ptr [rip + 0x21a36]
0x991a: jmp     0x98c6
0x991c: nop     dword ptr [rax]
0x9920: mov     dword ptr [rip + 0x21a16], 0
0x992a: mov     edi, 1
0x992f: call    0x9790
0x9934: movzx   eax, byte ptr [rip + 0x21a15]
0x993b: jmp     0x98c6
0x993d: nop     dword ptr [rax]
0x9940: push    r12
0x9942: mov     edx, 5
0x9947: lea     rsi, [rip + 0x17293]
0x994e: mov     r12d, 1
0x9954: push    rbp
0x9955: mov     rbp, rdi
0x9958: sub     rsp, 8
0x995c: call    0x4890
0x9961: test    eax, eax
0x9963: je      0x99d3
0x9965: mov     edx, 4
0x996a: lea     rsi, [rip + 0x1725f]
0x9971: mov     rdi, rbp
0x9974: call    0x4890
0x9979: test    eax, eax
0x997b: je      0x99d3
0x997d: mov     edx, 9
0x9982: lea     rsi, [rip + 0x1724c]
0x9989: mov     rdi, rbp
0x998c: call    0x4890
0x9991: test    eax, eax
0x9993: je      0x99d3
0x9995: lea     rsi, [rip + 0x17243]
0x999c: mov     rdi, rbp
0x999f: call    0x4b30
0x99a4: test    eax, eax
0x99a6: je      0x99d3
0x99a8: lea     rsi, [rip + 0x17238]
0x99af: mov     rdi, rbp
0x99b2: call    0x4b30
0x99b7: test    eax, eax
0x99b9: je      0x99d3
0x99bb: lea     rsi, [rip + 0x1722b]
0x99c2: mov     rdi, rbp
0x99c5: xor     r12d, r12d
0x99c8: call    0x4b30

```

```

0x99cd: test    eax, eax
0x99cf: sete    r12b
0x99d3: add     rsp, 8
0x99d7: mov     eax, r12d
0x99da: pop     rbp
0x99db: pop     r12
0x99dd: ret
0x99de: nop
0x99e0: push    r13
0x99e2: push    r12
0x99e4: mov     r12, qword ptr [rip + 0x20f8d]
0x99eb: push    rbp
0x99ec: test    r12, r12
0x99ef: je      0x9a58
0x99f1: mov     r13, rdi
0x99f4: jmp     0x9a13
0x99f6: mov     word ptr cs:[rax + rax]
0x9a00: js      0x9a30
0x9a02: mov     r12, qword ptr [r12 + 8]
0x9a07: mov     qword ptr [rip + 0x20f6a], r12
0x9a0e: test    r12, r12
0x9a11: je      0x9a58
0x9a13: mov     rbp, qword ptr [r12]
0x9a17: mov     rdi, r13
0x9a1a: mov     rsi, rbp
0x9a1d: call    0x4b30
0x9a22: test    eax, eax
0x9a24: jne     0x9a00
0x9a26: mov     rax, r12
0x9a29: pop     rbp
0x9a2a: pop     r12
0x9a2c: pop     r13
0x9a2e: ret
0x9a2f: nop
0x9a30: mov     rdi, r13
0x9a33: call    0x49b0
0x9a38: mov     rsi, rbp
0x9a3b: mov     rdi, r13
0x9a3e: mov     rdx, rax
0x9a41: call    0x4890
0x9a46: pop     rbp
0x9a47: neg     eax
0x9a49: sbb     r12, r12
0x9a4c: mov     rax, r12
0x9a4f: pop     r12
0x9a51: pop     r13
0x9a53: ret
0x9a54: nop     dword ptr [rax]

```

```

0x9a58: mov     r12, -1
0x9a5f: pop     rbp
0x9a60: mov     rax, r12
0x9a63: pop     r12
0x9a65: pop     r13
0x9a67: ret
0x9a68: nop     dword ptr [rax + rax]
0x9a70: endbr64
0x9a74: movsxd  rdx, dword ptr [rip + 0x2148d]
0x9a7b: push    r12
0x9a7d: push    rbp
0x9a7e: mov     ebp, edi
0x9a80: push    rbx
0x9a81: cmp     edx, 0x400
0x9a87: je      0x9ab0
0x9a89: lea     eax, [rdx + 1]
0x9a8c: lea     r12, [rip + 0x2148d]
0x9a93: mov     byte ptr [r12 + rdx], bpl
0x9a97: pop     rbx
0x9a98: mov     dword ptr [rip + 0x2146a], eax
0x9a9e: pop     rbp
0x9a9f: mov     eax, 1
0x9aa4: pop     r12
0x9aa6: ret
0x9aa7: nop     word ptr [rax + rax]
0x9ab0: xor     ebx, ebx
0x9ab2: lea     r12, [rip + 0x21467]
0x9ab9: jmp     0x9acc
0x9abb: nop     dword ptr [rax + rax]
0x9ac0: mov     edx, dword ptr [rip + 0x21442]
0x9ac6: add     ebx, eax
0x9ac8: cmp     edx, ebx
0x9aca: jle     0x9ae5
0x9acc: sub     edx, ebx
0x9ace: movsxd  rsi, ebx
0x9ad1: mov     edi, 1
0x9ad6: add     rsi, r12
0x9ad9: movsxd  rdx, edx
0x9adc: call    0x18da0
0x9ae1: test    eax, eax
0x9ae3: jns     0x9ac0
0x9ae5: mov     eax, 1
0x9aea: xor     edx, edx
0x9aec: jmp     0x9a93
0x9aee: nop
0x9af0: endbr64
0x9af4: mov     dword ptr [rip + 0x21862], edi
0x9afa: ret

```

```

0x9afb: nop      dword ptr [rax + rax]
0x9b00: endbr64
0x9b04: push     r12
0x9b06: push     rbp
0x9b07: push     rbx
0x9b08: mov      ebx, edi
0x9b0a: sub      rsp, 0x50
0x9b0e: mov      rax, qword ptr fs:[0x28]
0x9b17: mov      qword ptr [rsp + 0x48], rax
0x9b1c: xor      eax, eax
0x9b1e: cmp      edi, 1
0x9b21: je       0x9b60
0x9b23: lea      rdx, [rip + 0x21896]
0x9b2a: mov      esi, 1
0x9b2f: mov      edi, 1
0x9b34: call     0x4d70
0x9b39: mov      dword ptr [rip + 0x20e19], ebx
0x9b3f: mov      rax, qword ptr [rsp + 0x48]
0x9b44: sub      rax, qword ptr fs:[0x28]
0x9b4d: jne      0x9c7e
0x9b53: add      rsp, 0x50
0x9b57: pop      rbx
0x9b58: pop      rbp
0x9b59: pop      r12
0x9b5b: ret
0x9b5c: nop      dword ptr [rax]
0x9b60: mov      rax, qword ptr [rip + 0x21889]
0x9b67: mov      rdx, qword ptr [rip + 0x21852]
0x9b6e: lea      r12, [rip + 0x2180b]
0x9b75: mov      esi, 1
0x9b7a: movdqa   xmm0, xmmword ptr [rip + 0x2183e]
0x9b82: movdqa   xmm1, xmmword ptr [rip + 0x21846]
0x9b8a: mov      edi, 1
0x9b8f: mov      qword ptr [rip + 0x2181a], rax
0x9b96: mov      eax, dword ptr [rip + 0x2185c]
0x9b9c: movaps   xmmword ptr [rip + 0x217dd], xmm0
0x9ba3: movdqa   xmm2, xmmword ptr [rip + 0x21835]
0x9bab: mov      dword ptr [rip + 0x21807], eax
0x9bb1: movabs   rax, 0xffffffffffffa3c
0x9bbb: and      rax, rdx
0x9bbe: mov      rdx, r12
0x9bc1: movaps   xmmword ptr [rip + 0x217c8], xmm1
0x9bc8: mov      qword ptr [rip + 0x217b1], rax
0x9bcf: mov      eax, dword ptr [rip + 0x217b7]
0x9bd5: movaps   xmmword ptr [rip + 0x217c4], xmm2
0x9bdc: and      eax, 0xffffffff5
0x9bdf: mov      byte ptr [rip + 0x217bb], 0
0x9be6: mov      dword ptr [rip + 0x217a1], 7

```

```

0x9bf0: or      al, 0x81
0x9bf2: mov     dword ptr [rip + 0x21794], eax
0x9bf8: mov     eax, 0x100
0x9bfd: mov     word ptr [rip + 0x21792], ax
0x9c04: mov     qword ptr [rip + 0x2178a], 0x1a0000
0x9c0f: call    0x4d70
0x9c14: mov     rdi, r12
0x9c17: mov     r12, rsp
0x9c1a: call    0x4d10
0x9c1f: mov     edx, 1
0x9c24: mov     esi, 0x540a
0x9c29: mov     edi, 1
0x9c2e: mov     dword ptr [rip + 0x20cfc], eax
0x9c34: xor     eax, eax
0x9c36: call    0x4a70
0x9c3b: mov     esi, 1
0x9c40: mov     edi, 1
0x9c45: call    0x4db0
0x9c4a: mov     ebp, dword ptr [rip + 0x200d4]
0x9c50: mov     rsi, r12
0x9c53: mov     edi, 1
0x9c58: call    0x4d60
0x9c5d: mov     rdx, r12
0x9c60: mov     esi, 1
0x9c65: mov     edi, 1
0x9c6a: mov     byte ptr [rsp + 0x11], bpl
0x9c6f: mov     byte ptr [rsp + 0x12], 0
0x9c74: call    0x4d70
0x9c79: jmp     0x9b39
0x9c7e: call    0x49d0
0x9c83: nop     word ptr cs:[rax + rax]
0x9c8e: nop
0x9c90: endbr64
0x9c94: mov     eax, dword ptr [rip + 0x20cbe]
0x9c9a: ret
0x9c9b: nop     dword ptr [rax + rax]
0x9ca0: endbr64
0x9ca4: push    r12
0x9ca6: push    rbx
0x9ca7: mov     ebx, edi
0x9ca9: mov     edi, 1
0x9cae: sub     rsp, 0x58
0x9cb2: mov     rax, qword ptr fs:[0x28]
0x9cbb: mov     qword ptr [rsp + 0x48], rax
0x9cc0: xor     eax, eax
0x9cc2: mov     r12, rsp
0x9cc5: mov     rsi, r12
0x9cc8: call    0x4d60

```



```

0x9ccd: mov     rdx, r12
0x9cd0: mov     esi, 1
0x9cd5: mov     edi, 1
0x9cda: mov     byte ptr [rsp + 0x11], bl
0x9cde: mov     byte ptr [rsp + 0x12], 0
0x9ce3: call    0x4d70
0x9ce8: mov     dword ptr [rip + 0x20036], ebx
0x9cee: mov     rax, qword ptr [rsp + 0x48]
0x9cf3: sub     rax, qword ptr fs:[0x28]
0x9cfc: jne     0x9d06
0x9cfe: add     rsp, 0x58
0x9d02: pop     rbx
0x9d03: pop     r12
0x9d05: ret
0x9d06: call    0x49d0
0x9d0b: nop     dword ptr [rax + rax]
0x9d10: endbr64
0x9d14: mov     edx, dword ptr [rip + 0x211ee]
0x9d1a: test    edx, edx
0x9d1c: jle     0x9d66
0x9d1e: push    rbp
0x9d1f: xor     ebp, ebp
0x9d21: push    rbx
0x9d22: lea     rbx, [rip + 0x211f7]
0x9d29: sub     rsp, 8
0x9d2d: jmp     0x9d3c
0x9d2f: nop
0x9d30: mov     edx, dword ptr [rip + 0x211d2]
0x9d36: add     ebp, eax
0x9d38: cmp     edx, ebp
0x9d3a: jle     0x9d55
0x9d3c: sub     edx, ebp
0x9d3e: movsxd  rsi, ebp
0x9d41: mov     edi, 1
0x9d46: add     rsi, rbx
0x9d49: movsxd  rdx, edx
0x9d4c: call    0x18da0
0x9d51: test    eax, eax
0x9d53: jns     0x9d30
0x9d55: mov     dword ptr [rip + 0x211a9], 0
0x9d5f: add     rsp, 8
0x9d63: pop     rbx
0x9d64: pop     rbp
0x9d65: ret
0x9d66: mov     dword ptr [rip + 0x21198], 0
0x9d70: ret
0x9d71: nop     word ptr cs:[rax + rax]
0x9d7c: nop     dword ptr [rax]

```

```

0x9d80: endbr64
0x9d84: push    rbp
0x9d85: lea      rdx, [rip - 0x31c]
0x9d8c: push    rbx
0x9d8d: sub      rsp, 8
0x9d91: mov      esi, dword ptr [rip + 0x20c05]
0x9d97: mov      rdi, qword ptr [rip + 0x2004a]
0x9d9e: call     0x4820
0x9da3: mov      edx, dword ptr [rip + 0x2115f]
0x9da9: test     edx, edx
0x9dab: jle      0x9de5
0x9dad: xor      ebp, ebp
0x9daf: lea      rbx, [rip + 0x2116a]
0x9db6: jmp      0x9dcc
0x9db8: nop      dword ptr [rax + rax]
0x9dc0: mov      edx, dword ptr [rip + 0x21142]
0x9dc6: add      ebp, eax
0x9dc8: cmp      edx, ebp
0x9dca: jle      0x9de5
0x9dcc: sub      edx, ebp
0x9dce: movsxd   rsi, ebp
0x9dd1: mov      edi, 1
0x9dd6: add      rsi, rbx
0x9dd9: movsxd   rdx, edx
0x9ddc: call     0x18da0
0x9de1: test     eax, eax
0x9de3: jns      0x9dc0
0x9de5: mov      dword ptr [rip + 0x21119], 0
0x9def: add      rsp, 8
0x9df3: pop      rbx
0x9df4: pop      rbp
0x9df5: ret
0x9df6: nop      word ptr cs:[rax + rax]
0x9e00: endbr64
0x9e04: push    rbp
0x9e05: lea      rdx, [rip - 0x39c]
0x9e0c: push    rbx
0x9e0d: sub      rsp, 8
0x9e11: mov      eax, dword ptr [rip + 0x20b85]
0x9e17: mov      rdi, qword ptr [rip + 0x2010a]
0x9e1e: lea      esi, [rax - 1]
0x9e21: call     0x4820
0x9e26: mov      edx, dword ptr [rip + 0x210dc]
0x9e2c: test     edx, edx
0x9e2e: jle      0x9e65
0x9e30: xor      ebp, ebp
0x9e32: lea      rbx, [rip + 0x210e7]
0x9e39: jmp      0x9e4c

```

```

0x9e3b: nop      dword ptr [rax + rax]
0x9e40: mov      edx, dword ptr [rip + 0x210c2]
0x9e46: add      ebp, eax
0x9e48: cmp      edx, ebp
0x9e4a: jle      0x9e65
0x9e4c: sub      edx, ebp
0x9e4e: movsxd   rsi, ebp
0x9e51: mov      edi, 1
0x9e56: add      rsi, rbx
0x9e59: movsxd   rdx, edx
0x9e5c: call     0x18da0
0x9e61: test     eax, eax
0x9e63: jns      0x9e40
0x9e65: mov      dword ptr [rip + 0x21099], 0
0x9e6f: add      rsp, 8
0x9e73: pop      rbx
0x9e74: pop      rbp
0x9e75: ret
0x9e76: nop      word ptr cs:[rax + rax]
0x9e80: endbr64
0x9e84: push     rbp
0x9e85: lea      rdx, [rip - 0x41c]
0x9e8c: push     rbx
0x9e8d: sub      rsp, 8
0x9e91: mov      eax, dword ptr [rip + 0x20b05]
0x9e97: mov      rdi, qword ptr [rip + 0x200aa]
0x9e9e: lea      esi, [rax - 1]
0x9ea1: call     0x4820
0x9ea6: mov      edx, dword ptr [rip + 0x2105c]
0x9eac: test     edx, edx
0x9eae: jle      0x9ee5
0x9eb0: xor      ebp, ebp
0x9eb2: lea      rbx, [rip + 0x21067]
0x9eb9: jmp      0x9ecc
0x9ebb: nop      dword ptr [rax + rax]
0x9ec0: mov      edx, dword ptr [rip + 0x21042]
0x9ec6: add      ebp, eax
0x9ec8: cmp      edx, ebp
0x9eca: jle      0x9ee5
0x9ecc: sub      edx, ebp
0x9ece: movsxd   rsi, ebp
0x9ed1: mov      edi, 1
0x9ed6: add      rsi, rbx
0x9ed9: movsxd   rdx, edx
0x9edc: call     0x18da0
0x9ee1: test     eax, eax
0x9ee3: jns      0x9ec0
0x9ee5: mov      dword ptr [rip + 0x21019], 0

```

```

0x9eef: add    rsp, 8
0x9ef3: pop     rbx
0x9ef4: pop     rbp
0x9ef5: ret
0x9ef6: nop     word ptr cs:[rax + rax]
0x9f00: endbr64
0x9f04: mov     edx, dword ptr [rip + 0x214f2]
0x9f0a: movzx   eax, byte ptr [rdi]
0x9f0d: push    r14
0x9f0f: push    r13
0x9f11: mov     r13, rdi
0x9f14: push    r12
0x9f16: push    rbp
0x9f17: push    rbx
0x9f18: test    edx, edx
0x9f1a: jne     0x9fc8
0x9f20: cmp     al, 0x5e
0x9f22: jne     0x9fb8
0x9f28: lea     rbx, [r13 + 1]
0x9f2c: mov     rbp, r13
0x9f2f: lea     r14, [rip + 0x1fd6a]
0x9f36: jmp     0x9f56
0x9f38: nop     dword ptr [rax + rax]
0x9f40: add     rbp, 1
0x9f44: mov     byte ptr [rbx - 1], al
0x9f47: mov     rdx, rbx
0x9f4a: add     rbx, 1
0x9f4e: movzx   eax, byte ptr [rbp]
0x9f52: test    al, al
0x9f54: je      0x9fa6
0x9f56: cmp     al, 0x5e
0x9f58: jne     0x9f40
0x9f5a: movzx   r12d, byte ptr [rbp + 1]
0x9f5f: test    r12b, r12b
0x9f62: je      0x9fb8
0x9f64: call    0x47c0
0x9f69: mov     r8, rax
0x9f6c: movzx   eax, r12b
0x9f70: mov     rdx, qword ptr [r8]
0x9f73: mov     eax, dword ptr [rdx + rax*4]
0x9f76: cmp     eax, 0x47
0x9f79: je      0x9fb8
0x9f7b: cmp     eax, 0x5a
0x9f7e: je      0x9fb8
0x9f80: and     r12d, 0x7f
0x9f84: add     rbp, 2
0x9f88: mov     rdx, rbx
0x9f8b: add     rbx, 1

```

```

0x9f8f: sub     r12d, 0x20
0x9f93: movsxd   r12, r12d
0x9f96: movzx    eax, byte ptr [r14 + r12]
0x9f9b: mov      byte ptr [rbx - 2], al
0x9f9e: movzx    eax, byte ptr [rbp]
0x9fa2: test     al, al
0x9fa4: jne      0x9f56
0x9fa6: mov      byte ptr [rdx], 0
0x9fa9: mov      rax, r13
0x9fac: pop      rbx
0x9fad: pop      rbp
0x9fae: pop      r12
0x9fb0: pop      r13
0x9fb2: pop      r14
0x9fb4: ret
0x9fb5: nop      dword ptr [rax]
0x9fb8: pop      rbx
0x9fb9: xor      eax, eax
0x9fbb: pop      rbp
0x9fbc: pop      r12
0x9fbe: pop      r13
0x9fc0: pop      r14
0x9fc2: ret
0x9fc3: nop      dword ptr [rax + rax]
0x9fc8: test     al, al
0x9fca: jne      0x9f28
0x9fd0: mov      rdx, rdi
0x9fd3: jmp      0x9fa6
0x9fd5: nop      word ptr cs:[rax + rax]
0x9fe0: endbr64
0x9fe4: push     r15
0x9fe6: push     r14
0x9fe8: mov      r14, rdi
0x9feb: push     r13
0x9fed: lea     r13, [rip + 0x16c12]
0x9ff4: push     r12
0x9ff6: lea     r12, [rip + 0x209c3]
0x9ffd: push     rbp
0x9ffe: lea     rbp, [rip + 0x16bef]
0xa005: push     rbx
0xa006: mov      rbx, rdi
0xa009: sub      rsp, 0x18
0xa00d: mov      rax, qword ptr fs:[0x28]
0xa016: mov      qword ptr [rsp + 8], rax
0xa01b: xor      eax, eax
0xa01d: mov      byte ptr [rip + 0x2099c], 0
0xa024: movzx    r15d, byte ptr [rdi]
0xa028: test     r15b, r15b

```

```

0xa02b: jne      0xa0c2
0xa031: jmp      0xa0f0
0xa036: nop      word ptr cs:[rax + rax]
0xa040: cmp      r15b, 0x20
0xa044: je       0xa148
0xa04a: cmp      r15b, 0x7f
0xa04e: je       0xa160
0xa054: cmp      r15b, 0xff
0xa058: je       0xa180
0xa05e: call     0x4f00
0xa063: mov      r8, rax
0xa066: movzx    eax, r15b
0xa06a: mov      rsi, qword ptr [r8]
0xa06d: test     byte ptr [rsi + rax*2], 2
0xa071: je       0xa120
0xa077: lea      edx, [r15 + 0x40]
0xa07b: lea      rsi, [rsp + 5]
0xa080: mov      rdi, r12
0xa083: mov      byte ptr [rsp + 5], 0x5e
0xa088: mov      byte ptr [rsp + 6], dl
0xa08c: mov      edx, 0x12c
0xa091: mov      byte ptr [rsp + 7], 0
0xa096: call     0x47d0
0xa09b: add      r14, 1
0xa09f: cmp      byte ptr [r14], 0
0xa0a3: je       0xa0f0
0xa0a5: cmp      rbx, r14
0xa0a8: je       0xa0be
0xa0aa: mov      edx, 0x12c
0xa0af: lea      rsi, [rip + 0x1668f]
0xa0b6: mov      rdi, r12
0xa0b9: call     0x47d0
0xa0be: movzx    r15d, byte ptr [r14]
0xa0c2: cmp      r15b, 0x1b
0xa0c6: jne      0xa040
0xa0cc: mov      edx, 0x12c
0xa0d1: mov      rsi, rbp
0xa0d4: mov      rdi, r12
0xa0d7: add      r14, 1
0xa0db: call     0x47d0
0xa0e0: cmp      byte ptr [r14], 0
0xa0e4: jne      0xa0a5
0xa0e6: nop      word ptr cs:[rax + rax]
0xa0f0: mov      rax, qword ptr [rsp + 8]
0xa0f5: sub      rax, qword ptr fs:[0x28]
0xa0fe: jne      0xa199
0xa104: add      rsp, 0x18
0xa108: mov      rax, r12

```

```

0xa10b: pop    rbx
0xa10c: pop    rbp
0xa10d: pop    r12
0xa10f: pop    r13
0xa111: pop    r14
0xa113: pop    r15
0xa115: ret
0xa116: nop     word ptr cs:[rax + rax]
0xa120: lea     rsi, [rsp + 5]
0xa125: mov     edx, 0x12c
0xa12a: mov     rdi, r12
0xa12d: mov     byte ptr [rsp + 5], r15b
0xa132: mov     byte ptr [rsp + 6], 0
0xa137: call    0x47d0
0xa13c: jmp     0xa09b
0xa141: nop     dword ptr [rax]
0xa148: mov     edx, 0x12c
0xa14d: mov     rsi, r13
0xa150: mov     rdi, r12
0xa153: call    0x47d0
0xa158: jmp     0xa09b
0xa15d: nop     dword ptr [rax]
0xa160: mov     edx, 0x12c
0xa165: lea     rsi, [rip + 0x16a8f]
0xa16c: mov     rdi, r12
0xa16f: call    0x47d0
0xa174: jmp     0xa09b
0xa179: nop     dword ptr [rax]
0xa180: mov     edx, 0x12c
0xa185: lea     rsi, [rip + 0x16a79]
0xa18c: mov     rdi, r12
0xa18f: call    0x47d0
0xa194: jmp     0xa09b
0xa199: call    0x49d0
0xa19e: nop
0xa1a0: endbr64
0xa1a4: mov     eax, dword ptr [rip + 0x211ae]
0xa1aa: mov     r8, rdi
0xa1ad: mov     edi, dword ptr [rip + 0x207e5]
0xa1b3: push    r13
0xa1b5: push    r12
0xa1b7: xor     r12d, r12d
0xa1ba: lea     edx, [rax + rsi]
0xa1bd: push    rbx
0xa1be: mov     dword ptr [rip + 0x21194], edx
0xa1c4: cmp     edi, eax
0xa1c6: jle     0xa21c
0xa1c8: mov     ecx, dword ptr [rip + 0x21186]

```

```

0xa1ce: cmp      ecx, dword ptr [rip + 0x207c8]
0xa1d4: jge      0xa21c
0xa1d6: mov      r12d, edi
0xa1d9: sub      r12d, eax
0xa1dc: cmp      edx, edi
0xa1de: cmovle   r12d, esi
0xa1e2: imul     edi, ecx
0xa1e5: mov      rsi, r8
0xa1e8: movsxd   r13, r12d
0xa1eb: lea      ebx, [rdi + rax]
0xa1ee: mov      rdi, qword ptr [rip + 0x21143]
0xa1f5: mov      rdx, r13
0xa1f8: movsxd   rbx, ebx
0xa1fb: add      rdi, rbx
0xa1fe: call     0x4bb0
0xa203: add      rbx, qword ptr [rip + 0x21126]
0xa20a: movzx    esi, byte ptr [rip + 0x21140]
0xa211: mov      rdx, r13
0xa214: mov      rdi, rbx
0xa217: call     0x4a50
0xa21c: mov      eax, r12d
0xa21f: pop      rbx
0xa220: pop      r12
0xa222: pop      r13
0xa224: ret
0xa225: nop      word ptr cs:[rax + rax]
0xa230: endbr64
0xa234: push     r13
0xa236: push     r12
0xa238: xor      r12d, r12d
0xa23b: push     rbx
0xa23c: sub      rsp, 0x10
0xa240: mov      rax, qword ptr fs:[0x28]
0xa249: mov      qword ptr [rsp + 8], rax
0xa24e: xor      eax, eax
0xa250: mov      eax, dword ptr [rip + 0x21102]
0xa256: mov      byte ptr [rsp + 7], dil
0xa25b: mov      edi, dword ptr [rip + 0x20737]
0xa261: lea      edx, [rax + 1]
0xa264: mov      dword ptr [rip + 0x210ee], edx
0xa26a: cmp      eax, edi
0xa26c: jge      0xa2c2
0xa26e: mov      ecx, dword ptr [rip + 0x210e0]
0xa274: cmp      ecx, dword ptr [rip + 0x20722]
0xa27a: jge      0xa2c2
0xa27c: cmp      edx, edi
0xa27e: jle      0xa2e0
0xa280: mov      r12d, edi

```



```

0xa283: sub     r12d, eax
0xa286: movsxd  r13, r12d
0xa289: imul    edi, ecx
0xa28c: lea     rsi, [rsp + 7]
0xa291: mov     rdx, r13
0xa294: lea     ebx, [rdi + rax]
0xa297: mov     rdi, qword ptr [rip + 0x2109a]
0xa29e: movsxd  rbx, ebx
0xa2a1: add     rdi, rbx
0xa2a4: call    0x4bb0
0xa2a9: add     rbx, qword ptr [rip + 0x21080]
0xa2b0: movzx   esi, byte ptr [rip + 0x2109a]
0xa2b7: mov     rdx, r13
0xa2ba: mov     rdi, rbx
0xa2bd: call    0x4a50
0xa2c2: mov     rax, qword ptr [rsp + 8]
0xa2c7: sub     rax, qword ptr fs:[0x28]
0xa2d0: jne     0xa2ee
0xa2d2: add     rsp, 0x10
0xa2d6: mov     eax, r12d
0xa2d9: pop     rbx
0xa2da: pop     r12
0xa2dc: pop     r13
0xa2de: ret
0xa2df: nop
0xa2e0: mov     r13d, 1
0xa2e6: mov     r12d, 1
0xa2ec: jmp     0xa289
0xa2ee: call    0x49d0
0xa2f3: nop     word ptr cs:[rax + rax]
0xa2fe: nop
0xa300: endbr64
0xa304: push    rbp
0xa305: lea     rdx, [rip - 0x89c]
0xa30c: push    rbx
0xa30d: sub     rsp, 8
0xa311: mov     esi, dword ptr [rip + 0x20685]
0xa317: mov     rdi, qword ptr [rip + 0x1faca]
0xa31e: call    0x4820
0xa323: mov     edx, dword ptr [rip + 0x20bdf]
0xa329: test    edx, edx
0xa32b: jle     0xa365
0xa32d: xor     ebp, ebp
0xa32f: lea     rbx, [rip + 0x20bea]
0xa336: jmp     0xa34c
0xa338: nop     dword ptr [rax + rax]
0xa340: mov     edx, dword ptr [rip + 0x20bc2]
0xa346: add     ebp, eax

```

```

0xa348: cmp     edx, ebp
0xa34a: jle     0xa365
0xa34c: sub     edx, ebp
0xa34e: movsxd  rsi, ebp
0xa351: mov     edi, 1
0xa356: add     rsi, rbx
0xa359: movsxd  rdx, edx
0xa35c: call    0x18da0
0xa361: test    eax, eax
0xa363: jns     0xa340
0xa365: mov     edx, dword ptr [rip + 0x20631]
0xa36b: xor     esi, esi
0xa36d: imul    edx, dword ptr [rip + 0x20624]
0xa374: mov     dword ptr [rip + 0x20b8a], 0
0xa37e: mov     rdi, qword ptr [rip + 0x20fb3]
0xa385: movsxd  rdx, edx
0xa388: call    0x4a50
0xa38d: mov     edx, dword ptr [rip + 0x20609]
0xa393: imul    edx, dword ptr [rip + 0x205fe]
0xa39a: xor     esi, esi
0xa39c: mov     rdi, qword ptr [rip + 0x20f8d]
0xa3a3: movsxd  rdx, edx
0xa3a6: call    0x4a50
0xa3ab: mov     edx, dword ptr [rip + 0x205eb]
0xa3b1: imul    edx, dword ptr [rip + 0x205e0]
0xa3b8: xor     esi, esi
0xa3ba: mov     rdi, qword ptr [rip + 0x20f67]
0xa3c1: movsxd  rdx, edx
0xa3c4: call    0x4a50
0xa3c9: mov     edx, dword ptr [rip + 0x205cd]
0xa3cf: imul    edx, dword ptr [rip + 0x205c2]
0xa3d6: xor     esi, esi
0xa3d8: mov     rdi, qword ptr [rip + 0x20f41]
0xa3df: movsxd  rdx, edx
0xa3e2: call    0x4a50
0xa3e7: mov     dword ptr [rip + 0x20f67], 0
0xa3f1: mov     dword ptr [rip + 0x20f59], 0
0xa3fb: add     rsp, 8
0xa3ff: pop     rbx
0xa400: pop     rbp
0xa401: ret
0xa402: nop     word ptr cs:[rax + rax]
0xa40d: nop     dword ptr [rax]
0xa410: endbr64
0xa414: sub     rsp, 8
0xa418: mov     edx, dword ptr [rip + 0x2057e]
0xa41e: imul    edx, dword ptr [rip + 0x20573]
0xa425: mov     esi, 0x20

```

```

0xa42a: mov     rdi, qword ptr [rip + 0x20f07]
0xa431: movsxd  rdx, edx
0xa434: call    0x4a50
0xa439: mov     edx, dword ptr [rip + 0x2055d]
0xa43f: imul    edx, dword ptr [rip + 0x20552]
0xa446: movzx   esi, byte ptr [rip + 0x20f04]
0xa44d: mov     rdi, qword ptr [rip + 0x20edc]
0xa454: movsxd  rdx, edx
0xa457: call    0x4a50
0xa45c: mov     edx, dword ptr [rip + 0x2053a]
0xa462: imul    edx, dword ptr [rip + 0x2052f]
0xa469: xor     esi, esi
0xa46b: mov     rdi, qword ptr [rip + 0x20eb6]
0xa472: add     rsp, 8
0xa476: movsxd  rdx, edx
0xa479: jmp     0x4a50
0xa47e: nop
0xa480: endbr64
0xa484: mov     edx, dword ptr [rip + 0x20512]
0xa48a: imul    edx, dword ptr [rip + 0x20507]
0xa491: xor     esi, esi
0xa493: mov     rdi, qword ptr [rip + 0x20e8e]
0xa49a: movsxd  rdx, edx
0xa49d: jmp     0x4a50
0xa4a2: nop     word ptr cs:[rax + rax]
0xa4ad: nop     dword ptr [rax]
0xa4b0: endbr64
0xa4b4: mov     dword ptr [rip + 0x20e9a], edi
0xa4ba: mov     dword ptr [rip + 0x20e98], esi
0xa4c0: ret
0xa4c1: nop     word ptr cs:[rax + rax]
0xa4cc: nop     dword ptr [rax]
0xa4d0: endbr64
0xa4d4: mov     eax, dword ptr [rip + 0x20e7a]
0xa4da: mov     dword ptr [rdi], eax
0xa4dc: mov     eax, dword ptr [rip + 0x20e76]
0xa4e2: mov     dword ptr [rsi], eax
0xa4e4: ret
0xa4e5: nop     word ptr cs:[rax + rax]
0xa4f0: endbr64
0xa4f4: movzx   eax, byte ptr [rip + 0x20e56]
0xa4fb: and     edi, 7
0xa4fe: and     eax, 0xffffffff8
0xa501: or      eax, edi
0xa503: mov     byte ptr [rip + 0x20e48], al
0xa509: ret
0xa50a: nop     word ptr [rax + rax]
0xa510: endbr64

```

```

0xa514: movzx    eax, byte ptr [rip + 0x20e36]
0xa51b: shl      edi, 3
0xa51e: and      edi, 0x38
0xa521: and      eax, 0xffffffffc7
0xa524: or       eax, edi
0xa526: mov     byte ptr [rip + 0x20e25], al
0xa52c: ret
0xa52d: nop     dword ptr [rax]
0xa530: endbr64
0xa534: movzx    eax, byte ptr [rip + 0x20e16]
0xa53b: shl      edi, 6
0xa53e: and      edi, 0x40
0xa541: and      eax, 0xffffffffbf
0xa544: or       eax, edi
0xa546: mov     byte ptr [rip + 0x20e05], al
0xa54c: ret
0xa54d: nop     dword ptr [rax]
0xa550: endbr64
0xa554: movzx    eax, byte ptr [rip + 0x20df6]
0xa55b: shl      edi, 7
0xa55e: and      eax, 0x7f
0xa561: or       eax, edi
0xa563: mov     byte ptr [rip + 0x20de8], al
0xa569: ret
0xa56a: nop     word ptr [rax + rax]
0xa570: endbr64
0xa574: shl      edi, 6
0xa577: and      esi, 7
0xa57a: shl      edx, 3
0xa57d: mov     eax, edi
0xa57f: movzx    edi, byte ptr [rip + 0x20dcb]
0xa586: and      edx, 0x38
0xa589: and      eax, 0x40
0xa58c: and      edi, 0xffffffffb8
0xa58f: or       edi, eax
0xa591: or       edi, esi
0xa593: and      edi, 0xffffffffc7
0xa596: or       edi, edx
0xa598: mov     byte ptr [rip + 0x20db2], dil
0xa59f: ret
0xa5a0: endbr64
0xa5a4: push     rbp
0xa5a5: push     rbx
0xa5a6: sub     rsp, 8
0xa5aa: movsxd   rax, dword ptr [rip + 0x20957]
0xa5b1: cmp     eax, 0x400
0xa5b6: je      0xa610
0xa5b8: lea     edx, [rax + 1]

```

```

0xa5bb: lea      rbx, [rip + 0x2095e]
0xa5c2: mov      dword ptr [rip + 0x20940], edx
0xa5c8: mov      byte ptr [rbx + rax], 7
0xa5cc: test     edx, edx
0xa5ce: jle      0xa5fd
0xa5d0: xor      ebp, ebp
0xa5d2: jmp      0xa5e4
0xa5d4: nop      dword ptr [rax]
0xa5d8: mov      edx, dword ptr [rip + 0x2092a]
0xa5de: add      ebp, eax
0xa5e0: cmp      edx, ebp
0xa5e2: jle      0xa5fd
0xa5e4: sub      edx, ebp
0xa5e6: movsxd   rsi, ebp
0xa5e9: mov      edi, 1
0xa5ee: add      rsi, rbx
0xa5f1: movsxd   rdx, edx
0xa5f4: call     0x18da0
0xa5f9: test     eax, eax
0xa5fb: jns      0xa5d8
0xa5fd: mov      dword ptr [rip + 0x20901], 0
0xa607: add      rsp, 8
0xa60b: pop      rbx
0xa60c: pop      rbp
0xa60d: ret
0xa60e: nop
0xa610: xor      ebp, ebp
0xa612: lea      rbx, [rip + 0x20907]
0xa619: jmp      0xa62c
0xa61b: nop      dword ptr [rax + rax]
0xa620: add      ebp, eax
0xa622: mov      eax, dword ptr [rip + 0x208e0]
0xa628: cmp      eax, ebp
0xa62a: jle      0xa645
0xa62c: sub      eax, ebp
0xa62e: movsxd   rsi, ebp
0xa631: mov      edi, 1
0xa636: add      rsi, rbx
0xa639: movsxd   rdx, eax
0xa63c: call     0x18da0
0xa641: test     eax, eax
0xa643: jns      0xa620
0xa645: mov      dword ptr [rip + 0x208b9], 1
0xa64f: mov      edx, 1
0xa654: mov      byte ptr [rip + 0x208c5], 7
0xa65b: jmp      0xa5d0
0xa660: endbr64
0xa664: test     edi, edi

```

```

0xa666: je      0xa688
0xa668: mov     rdi, qword ptr [rip + 0x1f759]
0xa66f: test    rdi, rdi
0xa672: je      0xa694
0xa674: lea     rdx, [rip - 0xc0b]
0xa67b: mov     esi, 1
0xa680: jmp     0x4820
0xa685: nop     dword ptr [rax]
0xa688: mov     rdi, qword ptr [rip + 0x1f719]
0xa68f: test    rdi, rdi
0xa692: jne     0xa674
0xa694: ret
0xa695: nop     word ptr cs:[rax + rax]
0xa6a0: endbr64
0xa6a4: movzx   eax, byte ptr [rip + 0x20ca6]
0xa6ab: mov     byte ptr [rdi], al
0xa6ad: ret
0xa6ae: nop
0xa6b0: endbr64
0xa6b4: movzx   eax, byte ptr [rdi]
0xa6b7: mov     byte ptr [rip + 0x20c94], al
0xa6bd: ret
0xa6be: nop
0xa6c0: endbr64
0xa6c4: push    rbp
0xa6c5: push    rbx
0xa6c6: sub     rsp, 8
0xa6ca: cmp     dword ptr [rip + 0x1f58f], 1
0xa6d1: je      0xa720
0xa6d3: mov     rdi, qword ptr [rip + 0x1f66e]
0xa6da: test    rdi, rdi
0xa6dd: je      0xa6f0
0xa6df: lea     rdx, [rip - 0xc76]
0xa6e6: mov     esi, 1
0xa6eb: call    0x4820
0xa6f0: mov     dword ptr [rip + 0x20c52], 0
0xa6fa: mov     dword ptr [rip + 0x20c44], 0
0xa704: mov     dword ptr [rip + 0x20c36], 0
0xa70e: mov     dword ptr [rip + 0x20c28], 0
0xa718: add     rsp, 8
0xa71c: pop     rbx
0xa71d: pop     rbp
0xa71e: ret
0xa71f: nop
0xa720: mov     edx, dword ptr [rip + 0x207e2]
0xa726: cmp     edx, 0x3fb
0xa72c: jg      0xa750
0xa72e: movsxd  rax, edx

```

```

0xa731: lea     rcx, [rip + 0x207e8]
0xa738: add     edx, 4
0xa73b: add     rax, rcx
0xa73e: mov     ecx, dword ptr [rip + 0x1f5db]
0xa744: mov     dword ptr [rip + 0x207be], edx
0xa74a: mov     dword ptr [rax], ecx
0xa74c: jmp     0xa6d3
0xa74e: nop
0xa750: xor     ebx, ebx
0xa752: lea     rbp, [rip + 0x207c7]
0xa759: jmp     0xa76c
0xa75b: nop     dword ptr [rax + rax]
0xa760: mov     edx, dword ptr [rip + 0x207a2]
0xa766: add     ebx, eax
0xa768: cmp     edx, ebx
0xa76a: jle     0xa785
0xa76c: sub     edx, ebx
0xa76e: movsxd  rsi, ebx
0xa771: mov     edi, 1
0xa776: add     rsi, rbp
0xa779: movsxd  rdx, edx
0xa77c: call    0x18da0
0xa781: test    eax, eax
0xa783: jns     0xa760
0xa785: mov     edx, 4
0xa78a: lea     rax, [rip + 0x2078f]
0xa791: jmp     0xa73e
0xa793: nop     word ptr cs:[rax + rax]
0xa79e: nop
0xa7a0: push    r12
0xa7a2: mov     r12d, edi
0xa7a5: push    rbp
0xa7a6: mov     ebp, esi
0xa7a8: sub     rsp, 8
0xa7ac: mov     eax, dword ptr [rip + 0x1f71e]
0xa7b2: test    eax, eax
0xa7b4: je      0xa800
0xa7b6: mov     eax, ebp
0xa7b8: mov     esi, dword ptr [rip + 0x201da]
0xa7be: mov     edx, dword ptr [rip + 0x201d8]
0xa7c4: or      eax, r12d
0xa7c7: mov     rdi, qword ptr [rip + 0x1f63a]
0xa7ce: js      0xa7d4
0xa7d0: cmp     ebp, esi
0xa7d2: jl      0xa810
0xa7d4: sub     edx, 1
0xa7d7: sub     esi, 1
0xa7da: call    0x4960

```

```

0xa7df: add     rsp, 8
0xa7e3: mov     esi, 1
0xa7e8: lea     rdx, [rip - 0xd7f]
0xa7ef: pop     rbp
0xa7f0: mov     rdi, rax
0xa7f3: pop     r12
0xa7f5: jmp     0x4820
0xa7fa: nop     word ptr [rax + rax]
0xa800: xor     eax, eax
0xa802: call    0xa6c0
0xa807: jmp     0xa7b6
0xa809: nop     dword ptr [rax]
0xa810: cmp     r12d, edx
0xa813: jge     0xa7d4
0xa815: mov     edx, r12d
0xa818: mov     esi, ebp
0xa81a: jmp     0xa7da
0xa81c: nop     dword ptr [rax]
0xa820: endbr64
0xa824: push    r15
0xa826: push    r14
0xa828: push    r13
0xa82a: push    r12
0xa82c: push    rbp
0xa82d: push    rbx
0xa82e: sub     rsp, 0x38
0xa832: mov     ebx, dword ptr [rip + 0x20160]
0xa838: imul    ebx, dword ptr [rip + 0x2015d]
0xa83f: mov     rax, qword ptr fs:[0x28]
0xa848: mov     qword ptr [rsp + 0x28], rax
0xa84d: xor     eax, eax
0xa84f: mov     rdi, qword ptr [rip + 0x1f552]
0xa856: cmp     dword ptr [rip + 0x20aff], 1
0xa85d: sbb     ebx, 0
0xa860: test    rdi, rdi
0xa863: je      0xa876
0xa865: lea     rdx, [rip - 0xd7c]
0xa86c: mov     esi, 1
0xa871: call    0x4820
0xa876: test    ebx, ebx
0xa878: jle     0xab70
0xa87e: movsxd  rax, ebx
0xa881: mov     dword ptr [rsp + 4], 0xffffffff
0xa889: xor     ebp, ebp
0xa88b: mov     r13d, 0xffffffff
0xa891: mov     qword ptr [rsp + 8], rax
0xa896: lea     r12, [rip + 0x20683]
0xa89d: jmp     0xa9ce

```



```

0xa8a2: nop      word ptr [rax + rax]
0xa8a8: cdq
0xa8a9: idiv     dword ptr [rip + 0x200e9]
0xa8af: mov      r14d, edx
0xa8b2: cmp      edx, r13d
0xa8b5: jne      0xaa20
0xa8bb: cmp      eax, dword ptr [rsp + 4]
0xa8bf: jne      0xaa20
0xa8c5: mov      rax, qword ptr [rip + 0x20a64]
0xa8cc: movzx    ebx, byte ptr [rax + rbp]
0xa8d0: mov      edi, ebx
0xa8d2: mov      r14d, ebx
0xa8d5: sar      edi, 6
0xa8d8: and      r14d, 7
0xa8dc: and      edi, 1
0xa8df: call     0x9870
0xa8e4: mov      eax, ebx
0xa8e6: and      eax, 7
0xa8e9: cmp      dword ptr [rip + 0x20a5c], 1
0xa8f0: je       0xaa40
0xa8f6: cmp      dword ptr [rip + 0x1f363], 1
0xa8fd: je       0xaa5f
0xa903: xor      edi, edi
0xa905: cmp      eax, 7
0xa908: setne    dil
0xa90c: call     0x9790
0xa911: movzx    eax, byte ptr [rip + 0x20a38]
0xa918: mov      dword ptr [rip + 0x20a2a], 1
0xa922: and      eax, 0xffffffff8
0xa925: or       eax, r14d
0xa928: mov      byte ptr [rip + 0x20a22], al
0xa92e: sar      ebx, 3
0xa931: and      ebx, 7
0xa934: cmp      dword ptr [rip + 0x20a0d], 1
0xa93b: je       0xaab0
0xa941: cmp      dword ptr [rip + 0x1f318], 1
0xa948: je       0xaad8
0xa94e: xor      edi, edi
0xa950: test     ebx, ebx
0xa952: setne    dil
0xa956: call     0x9790
0xa95b: mov      edx, dword ptr [rip + 0x205a7]
0xa961: movzx    eax, byte ptr [rip + 0x209e8]
0xa968: shl      ebx, 3
0xa96b: mov      dword ptr [rip + 0x209d3], 1
0xa975: and      eax, 0xffffffffc7
0xa978: or       ebx, eax
0xa97a: mov      byte ptr [rip + 0x209d0], bl

```

```

0xa980: mov     rax, qword ptr [rip + 0x209b1]
0xa987: movzx   ebx, byte ptr [rax + rbp]
0xa98b: cmp     edx, 0x400
0xa991: je      0xab30
0xa997: lea     eax, [rdx + 1]
0xa99a: movsxd  rdx, edx
0xa99d: add     r13d, 1
0xa9a1: cmp     dword ptr [rip + 0x1fff0], r13d
0xa9a8: mov     dword ptr [rip + 0x2055a], eax
0xa9ae: mov     byte ptr [r12 + rdx], bl
0xa9b2: jne     0xa9bf
0xa9b4: mov     dword ptr [rsp + 4], 0xffffffff
0xa9bc: xor     r13d, r13d
0xa9bf: add     rbp, 1
0xa9c3: cmp     rbp, qword ptr [rsp + 8]
0xa9c8: je      0xab70
0xa9ce: mov     rdx, qword ptr [rip + 0x20953]
0xa9d5: mov     rcx, qword ptr [rip + 0x2095c]
0xa9dc: mov     eax, ebp
0xa9de: movzx   ebx, byte ptr [rdx + rbp]
0xa9e2: cmp     byte ptr [rcx + rbp], bl
0xa9e5: jne     0xa8a8
0xa9eb: mov     rdx, qword ptr [rip + 0x2092e]
0xa9f2: mov     rcx, qword ptr [rip + 0x20937]
0xa9f9: movzx   edi, byte ptr [rdx + rbp]
0xa9fd: cmp     byte ptr [rcx + rbp], dil
0xaa01: je      0xa9bf
0xaa03: cdq
0xaa04: idiv    dword ptr [rip + 0x1ff8e]
0xaa0a: mov     r14d, edx
0xaa0d: cmp     edx, r13d
0xaa10: je      0xa8bb
0xaa16: nop     word ptr cs:[rax + rax]
0xaa20: mov     esi, r14d
0xaa23: mov     edi, eax
0xaa25: mov     dword ptr [rsp + 4], eax
0xaa29: mov     r13d, r14d
0xaa2c: call    0xa7a0
0xaa31: jmp     0xa8c5
0xaa36: nop     word ptr cs:[rax + rax]
0xaa40: movzx   edx, byte ptr [rip + 0x20909]
0xaa47: and     edx, 7
0xaa4a: cmp     eax, edx
0xaa4c: je      0xa92e
0xaa52: cmp     dword ptr [rip + 0x1f207], 1
0xaa59: jne     0xa903
0xaa5f: mov     eax, dword ptr [rip + 0x164b4]
0xaa65: mov     edx, dword ptr [rip + 0x2049d]

```

```

0xaa6b: mov     dword ptr [rsp + 0x10], eax
0xaa6f: movzx   eax, byte ptr [rip + 0x164a7]
0xaa76: mov     byte ptr [rsp + 0x14], al
0xaa7a: lea     eax, [r14 + 0x30]
0xaa7e: mov     byte ptr [rsp + 0x13], al
0xaa82: cmp     edx, 0x3fa
0xaa88: jg      0xac18
0xaa8e: movsxd  rax, edx
0xaa91: add     edx, 5
0xaa94: add     rax, r12
0xaa97: mov     ecx, dword ptr [rsp + 0x10]
0xaa9b: mov     dword ptr [rip + 0x20467], edx
0xaa9d: mov     dword ptr [rax], ecx
0xaa9f: movzx   ecx, byte ptr [rsp + 0x14]
0xaab0: mov     byte ptr [rax + 4], cl
0xaab1: jmp     0xa911
0xaab2: movzx   eax, byte ptr [rip + 0x20899]
0xaab7: mov     edx, dword ptr [rip + 0x2044b]
0xaabd: sar     eax, 3
0xaac0: and     eax, 7
0xaac3: cmp     ebx, eax
0xaac5: je      0xa980
0xaacb: cmp     dword ptr [rip + 0x1f18e], 1
0xaad2: jne     0xa94e
0xaad8: mov     eax, dword ptr [rip + 0x16436]
0xaade: mov     edx, dword ptr [rip + 0x20424]
0xaae4: mov     dword ptr [rsp + 0x10], eax
0xaae8: movzx   eax, byte ptr [rip + 0x16429]
0xaaef: mov     byte ptr [rsp + 0x14], al
0xaaf3: lea     eax, [rbx + 0x30]
0xaaf6: mov     byte ptr [rsp + 0x13], al
0xaafa: cmp     edx, 0x3fa
0xab00: jg      0xac58
0xab06: movsxd  rax, edx
0xab09: add     edx, 5
0xab0c: add     rax, r12
0xab0f: mov     ecx, dword ptr [rsp + 0x10]
0xab13: mov     dword ptr [rip + 0x203ef], edx
0xab19: mov     dword ptr [rax], ecx
0xab1b: movzx   ecx, byte ptr [rsp + 0x14]
0xab20: mov     byte ptr [rax + 4], cl
0xab23: jmp     0xa961
0xab28: nop     dword ptr [rax + rax]
0xab30: xor     r14d, r14d
0xab33: jmp     0xab46
0xab35: nop     dword ptr [rax]
0xab38: mov     edx, dword ptr [rip + 0x203ca]
0xab3e: add     r14d, eax

```

```

0xab41: cmp     edx, r14d
0xab44: jle     0xab60
0xab46: sub     edx, r14d
0xab49: movsxd  rsi, r14d
0xab4c: mov     edi, 1
0xab51: add     rsi, r12
0xab54: movsxd  rdx, edx
0xab57: call    0x18da0
0xab5c: test    eax, eax
0xab5e: jns     0xab38
0xab60: mov     eax, 1
0xab65: xor     edx, edx
0xab67: jmp     0xa99a
0xab6c: nop     dword ptr [rax]
0xab70: mov     edi, dword ptr [rip + 0x207de]
0xab76: mov     esi, dword ptr [rip + 0x207dc]
0xab7c: call    0xa7a0
0xab81: mov     rdi, qword ptr [rip + 0x1f240]
0xab88: test    rdi, rdi
0xab8b: je      0xab9e
0xab8d: lea     rdx, [rip - 0x1124]
0xab94: mov     esi, 1
0xab99: call    0x4820
0xab9e: mov     edx, dword ptr [rip + 0x20364]
0xaba4: test    edx, edx
0xaba6: jne     0xac98
0xabac: mov     edx, dword ptr [rip + 0x1fde6]
0xabbb: imul    edx, dword ptr [rip + 0x1fde3]
0xab9: mov     rsi, qword ptr [rip + 0x20778]
0xabc0: mov     rdi, qword ptr [rip + 0x20761]
0xabc7: movsxd  rdx, edx
0xabca: call    0x4bb0
0xabcf: mov     edx, dword ptr [rip + 0x1fdc3]
0xabd5: imul    edx, dword ptr [rip + 0x1fdc0]
0xabdc: mov     rdi, qword ptr [rip + 0x2073d]
0xabe3: mov     rsi, qword ptr [rip + 0x20746]
0xabea: movsxd  rdx, edx
0xabed: mov     rax, qword ptr [rsp + 0x28]
0xabf2: sub     rax, qword ptr fs:[0x28]
0xabfb: jne     0xace4
0xac01: add     rsp, 0x38
0xac05: pop     rbx
0xac06: pop     rbp
0xac07: pop     r12
0xac09: pop     r13
0xac0b: pop     r14
0xac0d: pop     r15
0xac0f: jmp     0x4bb0

```

```

0xac14: nop      dword ptr [rax]
0xac18: xor       r15d, r15d
0xac1b: jmp       0xac2e
0xac1d: nop      dword ptr [rax]
0xac20: mov      edx, dword ptr [rip + 0x202e2]
0xac26: add      r15d, eax
0xac29: cmp      edx, r15d
0xac2c: jle      0xac48
0xac2e: sub      edx, r15d
0xac31: movsxd   rsi, r15d
0xac34: mov      edi, 1
0xac39: add      rsi, r12
0xac3c: movsxd   rdx, edx
0xac3f: call     0x18da0
0xac44: test     eax, eax
0xac46: jns      0xac20
0xac48: mov      edx, 5
0xac4d: mov      rax, r12
0xac50: jmp      0xaa97
0xac55: nop      dword ptr [rax]
0xac58: xor      r14d, r14d
0xac5b: jmp      0xac6e
0xac5d: nop      dword ptr [rax]
0xac60: mov      edx, dword ptr [rip + 0x202a2]
0xac66: add      r14d, eax
0xac69: cmp      edx, r14d
0xac6c: jle      0xac88
0xac6e: sub      edx, r14d
0xac71: movsxd   rsi, r14d
0xac74: mov      edi, 1
0xac79: add      rsi, r12
0xac7c: movsxd   rdx, edx
0xac7f: call     0x18da0
0xac84: test     eax, eax
0xac86: jns      0xac60
0xac88: mov      edx, 5
0xac8d: mov      rax, r12
0xac90: jmp      0xab0f
0xac95: nop      dword ptr [rax]
0xac98: mov      ebx, 0
0xac9d: lea      rbp, [rip + 0x2027c]
0xaca4: jg       0xacbc
0xaca6: jmp      0xacd5
0xaca8: nop      dword ptr [rax + rax]
0xacb0: mov      edx, dword ptr [rip + 0x20252]
0xacb6: add      ebx, eax
0xacb8: cmp      edx, ebx
0xacba: jle      0xacd5

```

```

0xacbc: sub     edx, ebx
0xacbe: movsxd  rsi, ebx
0xacc1: mov     edi, 1
0xacc6: add     rsi, rbp
0xacc9: movsxd  rdx, edx
0xaccc: call    0x18da0
0xacd1: test    eax, eax
0xacd3: jns     0xacb0
0xacd5: mov     dword ptr [rip + 0x20229], 0
0xacdf: jmp     0xabac
0xace4: call    0x49d0
0xace9: nop     dword ptr [rax]
0xacf0: endbr64
0xacf4: push    r12
0xacf6: xor     eax, eax
0xacf8: mov     r12, rdi
0xacfb: push    rbx
0xacfc: mov     ebx, esi
0xacfe: sub     rsp, 8
0xad02: call    0xa820
0xad07: mov     rdx, qword ptr [rip + 0x1fc1a]
0xad0e: test    rdx, rdx
0xad11: je      0xad17
0xad13: xor     eax, eax
0xad15: call    rdx
0xad17: movsxd  rdx, ebx
0xad1a: mov     rsi, r12
0xad1d: xor     edi, edi
0xad1f: call    0x18d30
0xad24: mov     rdx, qword ptr [rip + 0x1fbf5]
0xad2b: mov     r12d, eax
0xad2e: test    rdx, rdx
0xad31: je      0xad37
0xad33: xor     eax, eax
0xad35: call    rdx
0xad37: add     rsp, 8
0xad3b: mov     eax, r12d
0xad3e: pop     rbx
0xad3f: pop     r12
0xad41: ret
0xad42: nop     word ptr cs:[rax + rax]
0xad4d: nop     dword ptr [rax]
0xad50: push    r12
0xad52: mov     edi, 1
0xad57: push    rbp
0xad58: lea     rbp, [rip + 0x1fda1]
0xad5f: push    rbx
0xad60: sub     rsp, 0x50

```

```

0xad64: mov     rax, qword ptr fs:[0x28]
0xad6d: mov     qword ptr [rsp + 0x48], rax
0xad72: xor     eax, eax
0xad74: mov     r12, rsp
0xad77: mov     rsi, r12
0xad7a: call    0x4d60
0xad7f: mov     edx, 0xff
0xad84: mov     esi, 1
0xad89: mov     edi, 1
0xad8e: mov     word ptr [rsp + 0x11], dx
0xad93: mov     rdx, r12
0xad96: call    0x4d70
0xad9b: mov     edi, 1
0xada0: mov     dword ptr [rip + 0x1ef7a], 0xffffffff
0xadaa: call    0x18b70
0xadaf: mov     dword ptr [rip + 0x20147], 0
0xadb9: nop     dword ptr [rax]
0xadc0: xor     eax, eax
0xadc2: call    0xa820
0xadc7: mov     rdx, qword ptr [rip + 0x1fb5a]
0xadce: test    rdx, rdx
0xadd1: je      0xadd7
0xadd3: xor     eax, eax
0xadd5: call    rdx
0xadd7: mov     edx, 0x400
0xaddc: mov     rsi, rbp
0xaddf: xor     edi, edi
0xade1: call    0x18d30
0xade6: mov     rdx, qword ptr [rip + 0x1fb33]
0xaded: mov     ebx, eax
0xadeb: test    rdx, rdx
0xadf2: je      0xadf8
0xadf4: xor     eax, eax
0xadf6: call    rdx
0xadf8: mov     dword ptr [rip + 0x20106], ebx
0xadfe: test    ebx, ebx
0xae00: js      0xadc0
0xae02: xor     edi, edi
0xae04: call    0x18b70
0xae09: mov     rsi, r12
0xae0c: mov     edi, 1
0xae11: call    0x4d60
0xae16: mov     eax, 7
0xae1b: mov     rdx, r12
0xae1e: mov     esi, 1
0xae23: mov     edi, 1
0xae28: mov     word ptr [rsp + 0x11], ax
0xae2d: call    0x4d70

```

```

0xae32: mov     eax, dword ptr [rip + 0x200cc]
0xae38: mov     dword ptr [rip + 0x1eee2], 7
0xae42: test    eax, eax
0xae44: je      0xae9f
0xae46: movsxd  rsi, dword ptr [rip + 0x200b3]
0xae4d: movsx   r8d, byte ptr [rbp + rsi]
0xae53: mov     rdx, rsi
0xae56: mov     ecx, r8d
0xae59: test    r8b, r8b
0xae5c: js      0xae90
0xae5e: sub     eax, 1
0xae61: add     edx, 1
0xae64: mov     dword ptr [rip + 0x2009a], eax
0xae6a: mov     dword ptr [rip + 0x20090], edx
0xae70: mov     rax, qword ptr [rsp + 0x48]
0xae75: sub     rax, qword ptr fs:[0x28]
0xae7e: jne     0xaea7
0xae80: add     rsp, 0x50
0xae84: mov     eax, r8d
0xae87: pop     rbx
0xae88: pop     rbp
0xae89: pop     r12
0xae8b: ret
0xae8c: nop     dword ptr [rax]
0xae90: and     ecx, 0x7f
0xae93: mov     r8d, 0x1b
0xae99: mov     byte ptr [rbp + rsi], cl
0xae9d: jmp     0xae70
0xae9f: mov     r8d, 0xffffffff
0xaea5: jmp     0xae70
0xaea7: call    0x49d0
0xaeac: nop     dword ptr [rax]
0xaeb0: endbr64
0xaeb4: xor     eax, eax
0xaeb6: sub     rsp, 8
0xaeba: call    0x18ab0
0xaebf: mov     eax, dword ptr [rip + 0x2003f]
0xaec5: test    eax, eax
0xaec7: jne     0xaed8
0xaec9: xor     eax, eax
0xaecb: add     rsp, 8
0xaecf: jmp     0xad50
0xaed4: nop     dword ptr [rax]
0xaed8: movsxd  rsi, dword ptr [rip + 0x20021]
0xaedf: lea     rdi, [rip + 0x1fc1a]
0xae6: movsx   r8d, byte ptr [rdi + rsi]
0xaeeb: mov     rdx, rsi
0xaeee: mov     ecx, r8d

```



```

0xaef1: test    r8b, r8b
0xaef4: js      0xaf10
0xaef6: sub     eax, 1
0xaef9: add     edx, 1
0xaefc: mov     dword ptr [rip + 0x20002], eax
0xaf02: mov     dword ptr [rip + 0x1fff8], edx
0xaf08: mov     eax, r8d
0xaf0b: add     rsp, 8
0xaf0f: ret
0xaf10: and     ecx, 0x7f
0xaf13: mov     r8d, 0x1b
0xaf19: mov     byte ptr [rdi + rsi], cl
0xaf1c: jmp     0xaf08
0xaf1e: nop
0xaf20: endbr64
0xaf24: push    r13
0xaf26: mov     r13, rsi
0xaf29: push    r12
0xaf2b: mov     r12, rdx
0xaf2e: push    rbp
0xaf2f: mov     rbp, rdi
0xaf32: mov     edi, 0x18
0xaf37: push    rbx
0xaf38: sub     rsp, 8
0xaf3c: call    0x1d670
0xaf41: mov     rdi, r13
0xaf44: mov     rbx, rax
0xaf47: call    0x1d7a0
0xaf4c: mov     qword ptr [rbx + 0x10], r12
0xaf50: mov     qword ptr [rbx], rax
0xaf53: mov     rax, qword ptr [rbp]
0xaf57: mov     qword ptr [rbx + 8], rax
0xaf5b: mov     qword ptr [rbp], rbx
0xaf5f: add     rsp, 8
0xaf63: pop     rbx
0xaf64: pop     rbp
0xaf65: pop     r12
0xaf67: pop     r13
0xaf69: ret
0xaf6a: nop     word ptr [rax + rax]
0xaf70: endbr64
0xaf74: cmp     byte ptr [rdi], 0
0xaf77: je      0xb030
0xaf7d: push    r13
0xaf7f: push    r12
0xaf81: mov     r12, rsi
0xaf84: push    rbp
0xaf85: mov     rbp, rdi

```

```

0xaf88: push    rbx
0xaf89: sub     rsp, 8
0xaf8d: mov     r13, qword ptr [rip + 0x1fa0c]
0xaf94: test    r13, r13
0xaf97: je      0xb010
0xaf99: mov     rbx, qword ptr [r13]
0xaf9d: mov     rsi, qword ptr [rbx]
0xafa0: call    0x4b30
0xafa5: test    eax, eax
0xafa7: jle     0xb010
0xafa9: nop     dword ptr [rax]
0xafb0: lea     r13, [rbx + 8]
0xafb4: mov     rbx, qword ptr [rbx + 8]
0xafb8: mov     qword ptr [rip + 0x1f9e1], r13
0xafbf: test    rbx, rbx
0xafc2: je      0xafd3
0xafc4: mov     rsi, qword ptr [rbx]
0xafc7: mov     rdi, rbp
0xafca: call    0x4b30
0xafcf: test    eax, eax
0xafd1: jg      0xafb0
0xafd3: mov     edi, 0x18
0xafd8: call    0x1d670
0xafdd: mov     rdi, rbp
0xafe0: mov     rbx, rax
0xafe3: call    0x1d7a0
0xafe8: mov     qword ptr [rbx + 0x10], r12
0xafec: mov     qword ptr [rbx], rax
0xafef: mov     rax, qword ptr [r13]
0xaff3: mov     qword ptr [rbx + 8], rax
0xaff7: mov     qword ptr [r13], rbx
0xaffb: add     rsp, 8
0xafff: pop     rbx
0xb000: pop     rbp
0xb001: pop     r12
0xb003: pop     r13
0xb005: ret
0xb006: nop     word ptr cs:[rax + rax]
0xb010: mov     rbx, qword ptr [rip + 0x1f969]
0xb017: lea     r13, [rip + 0x1f962]
0xb01e: mov     qword ptr [rip + 0x1f97b], r13
0xb025: test    rbx, rbx
0xb028: jne     0xafc4
0xb02a: jmp     0xafd3
0xb02c: nop     dword ptr [rax]
0xb030: ret
0xb031: nop     word ptr cs:[rax + rax]
0xb03c: nop     dword ptr [rax]

```

```

0xb040: endbr64
0xb044: mov     rax, qword ptr [rip + 0x1f935]
0xb04b: mov     qword ptr [rip + 0x1f926], rax
0xb052: ret
0xb053: nop     word ptr cs:[rax + rax]
0xb05e: nop
0xb060: endbr64
0xb064: cmp     qword ptr [rip + 0x1f90c], 0
0xb06c: je      0xb073
0xb06e: jmp     0x99e0
0xb073: xor     eax, eax
0xb075: ret
0xb076: nop     word ptr cs:[rax + rax]
0xb080: endbr64
0xb084: mov     eax, dword ptr [rip + 0x1f912]
0xb08a: push    r12
0xb08c: mov     r12, rdi
0xb08f: push    rbp
0xb090: push    rbx
0xb091: sub     eax, 1
0xb094: movzx   ebx, byte ptr [rip + 0x202b6]
0xb09b: mov     dword ptr [rip + 0x202b3], eax
0xb0a1: mov     dword ptr [rip + 0x202ad], 0
0xb0ab: mov     eax, ebx
0xb0ad: and     eax, 0xffffffffc0
0xb0b0: or      eax, 0x38
0xb0b3: mov     byte ptr [rip + 0x20298], al
0xb0b9: mov     eax, dword ptr [rip + 0x1f8d9]
0xb0bf: lea     edi, [rax + 1]
0xb0c2: movsxd  rdi, edi
0xb0c5: call    0x1d670
0xb0ca: movsxd  rdx, dword ptr [rip + 0x1f8c7]
0xb0d1: mov     esi, 0x20
0xb0d6: mov     rdi, rax
0xb0d9: mov     rbp, rax
0xb0dc: call    0x4a50
0xb0e1: movsxd  rax, dword ptr [rip + 0x1f8b0]
0xb0e8: mov     rdi, rbp
0xb0eb: mov     byte ptr [rbp + rax], 0
0xb0f0: mov     esi, dword ptr [rip + 0x1f8a2]
0xb0f6: call    0xa1a0
0xb0fb: mov     rdi, rbp
0xb0fe: lea     rbp, [rip + 0x1f8bb]
0xb105: call    0x18d10
0xb10a: mov     eax, dword ptr [rip + 0x1f88c]
0xb110: mov     rdi, r12
0xb113: mov     dword ptr [rip + 0x2023b], 0
0xb11d: sub     eax, 1

```

```

0xb120: mov     dword ptr [rip + 0x2022e], eax
0xb126: call    0x9fe0
0xb12b: mov     esi, 0x13
0xb130: lea     rdi, [rip + 0x15ad5]
0xb137: call    0xa1a0
0xb13c: mov     rdi, rbp
0xb13f: call    0x49b0
0xb144: mov     rdi, rbp
0xb147: mov     esi, eax
0xb149: call    0xa1a0
0xb14e: mov     esi, 1
0xb153: lea     rdi, [rip + 0x155eb]
0xb15a: call    0xa1a0
0xb15f: xor     eax, eax
0xb161: call    0xa820
0xb166: mov     byte ptr [rip + 0x201e5], bl
0xb16c: pop     rbx
0xb16d: pop     rbp
0xb16e: pop     r12
0xb170: ret
0xb171: nop     word ptr cs:[rax + rax]
0xb17c: nop     dword ptr [rax]
0xb180: endbr64
0xb184: push    r15
0xb186: push    r14
0xb188: mov     r14, rdi
0xb18b: push    r13
0xb18d: push    r12
0xb18f: push    rbp
0xb190: lea     rbp, [rip + 0x1f969]
0xb197: push    rbx
0xb198: sub     rsp, 0x28
0xb19c: jmp     0xb1ae
0xb19e: nop
0xb1a0: xor     eax, eax
0xb1a2: call    0xad50
0xb1a7: mov     ebx, eax
0xb1a9: cmp     ebx, -1
0xb1ac: jne     0xb1ef
0xb1ae: xor     eax, eax
0xb1b0: call    0x18ab0
0xb1b5: mov     eax, dword ptr [rip + 0x1fd49]
0xb1bb: test    eax, eax
0xb1bd: je      0xb1a0
0xb1bf: movsxd  rsi, dword ptr [rip + 0x1fd3a]
0xb1c6: movsx   ebx, byte ptr [rbp + rsi]
0xb1cb: mov     rdx, rsi
0xb1ce: mov     ecx, ebx

```

```

0xb1d0: test    bl, bl
0xb1d2: js      0xb4a0
0xb1d8: sub     eax, 1
0xb1db: add     edx, 1
0xb1de: mov     dword ptr [rip + 0x1fd20], eax
0xb1e4: mov     dword ptr [rip + 0x1fd16], edx
0xb1ea: cmp     ebx, -1
0xb1ed: je      0xb1ae
0xb1ef: test    r14, r14
0xb1f2: je      0xb1fb
0xb1f4: mov     dword ptr [r14], 1
0xb1fb: mov     eax, dword ptr [rip + 0x201fb]
0xb201: test    ebx, ebx
0xb203: jne     0xb445
0xb209: mov     ebx, 0xff
0xb20e: test    eax, eax
0xb210: jne     0xb240
0xb212: xor     r12d, r12d
0xb215: mov     ebp, 0x1fe
0xb21a: mov     ebx, 0xff
0xb21f: call    0x4f00
0xb224: mov     rax, qword ptr [rax]
0xb227: movzx   eax, word ptr [rax + rbp]
0xb22b: test    ah, 0x40
0xb22e: jne     0xb47d
0xb234: test    r12b, r12b
0xb237: jne     0xb47d
0xb23d: nop     dword ptr [rax]
0xb240: mov     rax, qword ptr [rip + 0x1f739]
0xb247: xor     r12d, r12d
0xb24a: lea     r13, [rip + 0x1f8af]
0xb251: mov     dword ptr [rip + 0x1f751], 0
0xb25b: mov     dword ptr [rip + 0x1f743], 0
0xb265: mov     qword ptr [rip + 0x1f70c], rax
0xb26c: test    ebx, ebx
0xb26e: mov     eax, 0xffffffff
0xb273: movsxd  rbp, r12d
0xb276: cmovne  eax, ebx
0xb279: mov     rbx, qword ptr [rip + 0x200e0]
0xb280: mov     byte ptr [rbx + rbp], al
0xb283: add     rbp, 1
0xb287: mov     byte ptr [rbx + rbp], 0
0xb28b: cmp     qword ptr [rip + 0x1f6e5], 0
0xb293: je      0xb2b3
0xb295: mov     rdi, rbx
0xb298: call    0x99e0
0xb29d: mov     r15, rax
0xb2a0: cmp     rax, -1

```

```

0xb2a4: je      0xb33a
0xb2aa: test   rax, rax
0xb2ad: jne     0xb38f
0xb2b3: mov     ecx, dword ptr [rip + 0x1fc4b]
0xb2b9: test    ecx, ecx
0xb2bb: jne     0xb2ee
0xb2bd: mov     edx, dword ptr [rip + 0x1f6e5]
0xb2c3: test    edx, edx
0xb2c5: je      0xb380
0xb2cb: mov     rdi, rbx
0xb2ce: call    0xb080
0xb2d3: mov     dword ptr [rip + 0x1f6cf], 1
0xb2dd: jmp     0xb2ee
0xb2df: nop
0xb2e0: xor     eax, eax
0xb2e2: call    0xad50
0xb2e7: mov     ebx, eax
0xb2e9: cmp     ebx, -1
0xb2ec: jne     0xb32c
0xb2ee: xor     eax, eax
0xb2f0: call    0x18ab0
0xb2f5: mov     eax, dword ptr [rip + 0x1fc09]
0xb2fb: test    eax, eax
0xb2fd: je      0xb2e0
0xb2ff: movsxd  rsi, dword ptr [rip + 0x1fbfa]
0xb306: movsx   ebx, byte ptr [r13 + rsi]
0xb30c: mov     rdx, rsi
0xb30f: mov     ecx, ebx
0xb311: test    bl, bl
0xb313: js      0xb360
0xb315: sub     eax, 1
0xb318: add     edx, 1
0xb31b: mov     dword ptr [rip + 0x1fbe3], eax
0xb321: mov     dword ptr [rip + 0x1fbd9], edx
0xb327: cmp     ebx, -1
0xb32a: je      0xb2ee
0xb32c: add     r12d, 1
0xb330: cmp     r12d, 0xf
0xb334: jne     0xb26c
0xb33a: mov     edi, 1
0xb33f: xor     r15d, r15d
0xb342: call    0x4aa0
0xb347: mov     dword ptr [rip + 0x1f65b], 0
0xb351: jmp     0xb433
0xb356: nop     word ptr cs:[rax + rax]
0xb360: and     ecx, 0x7f
0xb363: add     r12d, 1
0xb367: mov     byte ptr [r13 + rsi], cl

```

```

0xb36c: cmp     r12d, 0xf
0xb370: je        0xb33a
0xb372: mov     eax, 0x1b
0xb377: jmp     0xb279
0xb37c: nop     dword ptr [rax]
0xb380: mov     edi, 1
0xb385: call    0x4aa0
0xb38a: jmp     0xb2d3
0xb38f: test    r14, r14
0xb392: je        0xb41f
0xb398: mov     ecx, dword ptr [rip + 0x1fb66]
0xb39e: cmp     r12d, ecx
0xb3a1: jge     0xb41f
0xb3a3: movsxd  rax, dword ptr [rip + 0x1fb56]
0xb3aa: lea     r9d, [r12 + 1]
0xb3af: lea     rdx, [rip + 0x1f74a]
0xb3b6: movsxd  rdi, r9d
0xb3b9: mov     qword ptr [rsp + 0x18], rdi
0xb3be: lea     r8d, [r9 + rax]
0xb3c2: lea     r13, [rax + rdx]
0xb3c6: jmp     0xb3f1
0xb3c8: nop     dword ptr [rax + rax]
0xb3d0: sub     ecx, r9d
0xb3d3: add     dword ptr [r14], 1
0xb3d7: add     r13, qword ptr [rsp + 0x18]
0xb3dc: mov     dword ptr [rip + 0x1fb1d], r8d
0xb3e3: add     r8d, r9d
0xb3e6: mov     dword ptr [rip + 0x1fb18], ecx
0xb3ec: cmp     r12d, ecx
0xb3ef: jge     0xb41f
0xb3f1: mov     rdx, rbp
0xb3f4: mov     rsi, r13
0xb3f7: mov     rdi, rbx
0xb3fa: mov     dword ptr [rsp + 0x14], r8d
0xb3ff: mov     dword ptr [rsp + 0x10], r9d
0xb404: mov     dword ptr [rsp + 0xc], ecx
0xb408: call    0x4af0
0xb40d: mov     ecx, dword ptr [rsp + 0xc]
0xb411: mov     r9d, dword ptr [rsp + 0x10]
0xb416: test    eax, eax
0xb418: mov     r8d, dword ptr [rsp + 0x14]
0xb41d: je        0xb3d0
0xb41f: mov     edi, 1
0xb424: call    0x4aa0
0xb429: mov     dword ptr [rip + 0x1f579], 0
0xb433: add     rsp, 0x28
0xb437: mov     rax, r15
0xb43a: pop     rbx

```

```

0xb43b: pop    rbp
0xb43c: pop    r12
0xb43e: pop    r13
0xb440: pop    r14
0xb442: pop    r15
0xb444: ret
0xb445: test   eax, eax
0xb447: jne    0xb240
0xb44d: cmp    ebx, 0xa
0xb450: sete   bpl
0xb454: cmp    ebx, 0xd
0xb457: sete   al
0xb45a: or     ebp, eax
0xb45c: call   0x4f00
0xb461: mov    rax, qword ptr [rax]
0xb464: test   bpl, bpl
0xb467: je     0xb4c6
0xb469: mov    ebx, 0xa
0xb46e: test   byte ptr [rax + 0x15], 0x40
0xb472: je     0xb240
0xb478: mov    ebx, 0xa
0xb47d: mov    rax, qword ptr [rip + 0x1f4dc]
0xb484: lea    r15, [rip + 0x1f4d5]
0xb48b: mov    byte ptr [rax], bl
0xb48d: mov    rax, qword ptr [rip + 0x1f4cc]
0xb494: mov    byte ptr [rax + 1], 0
0xb498: jmp    0xb433
0xb49a: nop    word ptr [rax + rax]
0xb4a0: and    ecx, 0x7f
0xb4a3: mov    byte ptr [rbp + rsi], cl
0xb4a7: test   r14, r14
0xb4aa: je     0xb4b3
0xb4ac: mov    dword ptr [r14], 1
0xb4b3: mov    eax, dword ptr [rip + 0x1ff43]
0xb4b9: mov    ebx, 0x1b
0xb4be: test   eax, eax
0xb4c0: jne    0xb240
0xb4c6: movsxd rbp, ebx
0xb4c9: add    rbp, rbp
0xb4cc: cmp    ebx, 7
0xb4cf: sete   r12b
0xb4d3: jmp    0xb21f
0xb4d8: nop    dword ptr [rax + rax]
0xb4e0: endbr64
0xb4e4: mov    eax, dword ptr [rip + 0x1f4c2]
0xb4ea: test   eax, eax
0xb4ec: jne    0xb4f0
0xb4ee: ret

```



```

0xb4ef: nop
0xb4f0: sub     rsp, 8
0xb4f4: mov     rdi, qword ptr [rip + 0x1fe65]
0xb4fb: call    0xb080
0xb500: mov     dword ptr [rip + 0x1f49e], 1
0xb50a: add     rsp, 8
0xb50e: ret
0xb50f: nop
0xb510: endbr64
0xb514: mov     rax, qword ptr [rip + 0x1fe45]
0xb51b: ret
0xb51c: nop     dword ptr [rax]
0xb520: endbr64
0xb524: push    r15
0xb526: mov     esi, 0x5413
0xb52b: mov     edi, 1
0xb530: push    r14
0xb532: push    r13
0xb534: push    r12
0xb536: push    rbp
0xb537: push    rbx
0xb538: sub     rsp, 0x58
0xb53c: mov     rax, qword ptr fs:[0x28]
0xb545: mov     qword ptr [rsp + 0x48], rax
0xb54a: xor     eax, eax
0xb54c: lea     rdx, [rsp + 0x18]
0xb551: mov     dword ptr [rsp + 0x10], 0
0xb559: mov     dword ptr [rsp + 0x14], 0
0xb561: call    0x4a70
0xb566: test    eax, eax
0xb568: jne     0xb5a0
0xb56a: mov     ebx, eax
0xb56c: movzx   eax, word ptr [rsp + 0x1a]
0xb571: test    ax, ax
0xb574: je      0xb790
0xb57a: movzx   ebp, word ptr [rsp + 0x18]
0xb57f: test    bp, bp
0xb582: je      0xb790
0xb588: movzx   ebx, ax
0xb58b: lea     r12d, [rbp - 1]
0xb58f: lea     eax, [rbx - 1]
0xb592: mov     dword ptr [rsp + 0xc], eax
0xb596: jmp     0xb5b2
0xb598: nop     dword ptr [rax + rax]
0xb5a0: mov     dword ptr [rsp + 0xc], 0xffffffff
0xb5a8: mov     r12d, 0xffffffff
0xb5ae: xor     ebx, ebx
0xb5b0: xor     ebp, ebp

```

```

0xb5b2: lea    r13, [rip + 0x15667]
0xb5b9: lea    r15, [rip + 0x15a13]
0xb5c0: mov     rdi, r13
0xb5c3: call    0x47e0
0xb5c8: mov     rdi, rax
0xb5cb: test    rax, rax
0xb5ce: je      0xb5df
0xb5d0: lea     rdx, [rsp + 0x14]
0xb5d5: mov     rsi, r15
0xb5d8: xor     eax, eax
0xb5da: call    0x4c60
0xb5df: lea     r14, [rip + 0x15642]
0xb5e6: mov     rdi, r14
0xb5e9: call    0x47e0
0xb5ee: mov     rdi, rax
0xb5f1: test    rax, rax
0xb5f4: je      0xb605
0xb5f6: lea     rdx, [rsp + 0x10]
0xb5fb: mov     rsi, r15
0xb5fe: xor     eax, eax
0xb600: call    0x4c60
0xb605: cmp     dword ptr [rsp + 0xc], 0x3ff
0xb60d: mov     eax, dword ptr [rip + 0x1e8fd]
0xb613: jbe     0xb628
0xb615: mov     ebx, dword ptr [rsp + 0x14]
0xb619: lea     edx, [rbx - 1]
0xb61c: cmp     edx, 0x3ff
0xb622: ja      0xb7c0
0xb628: mov     dword ptr [rip + 0x1f36a], ebx
0xb62e: cmp     r12d, 0x3ff
0xb635: jbe     0xb64a
0xb637: mov     ebp, dword ptr [rsp + 0x10]
0xb63b: lea     edx, [rbp - 1]
0xb63e: cmp     edx, 0x3ff
0xb644: ja      0xb7a8
0xb64a: lea     r12, [rsp + 0x20]
0xb64f: mov     r8d, ebp
0xb652: mov     rcx, r15
0xb655: mov     edx, 0x20
0xb65a: mov     esi, 1
0xb65f: mov     rdi, r12
0xb662: xor     eax, eax
0xb664: mov     dword ptr [rip + 0x1f332], ebp
0xb66a: call    0x4f10
0xb66f: mov     rsi, r12
0xb672: mov     rdi, r14
0xb675: call    0x18380
0xb67a: mov     rcx, r15

```

```

0xb67d: mov     edx, 0x20
0xb682: mov     rdi, r12
0xb685: mov     r8d, dword ptr [rip + 0x1f30c]
0xb68c: mov     esi, 1
0xb691: xor     eax, eax
0xb693: call    0x4f10
0xb698: mov     rdi, r13
0xb69b: mov     rsi, r12
0xb69e: call    0x18380
0xb6a3: mov     rdi, qword ptr [rip + 0x1fc8e]
0xb6aa: test    rdi, rdi
0xb6ad: je      0xb6b4
0xb6af: call    0x18d10
0xb6b4: mov     rdi, qword ptr [rip + 0x1fc75]
0xb6bb: test    rdi, rdi
0xb6be: je      0xb6c5
0xb6c0: call    0x18d10
0xb6c5: mov     rdi, qword ptr [rip + 0x1fc5c]
0xb6cc: test    rdi, rdi
0xb6cf: je      0xb6d6
0xb6d1: call    0x18d10
0xb6d6: mov     rdi, qword ptr [rip + 0x1fc43]
0xb6dd: test    rdi, rdi
0xb6e0: je      0xb6e7
0xb6e2: call    0x18d10
0xb6e7: mov     edi, dword ptr [rip + 0x1f2ab]
0xb6ed: imul    edi, dword ptr [rip + 0x1f2a8]
0xb6f4: mov     esi, 1
0xb6f9: movsxd  rdi, edi
0xb6fc: call    0x1d740
0xb701: mov     edi, dword ptr [rip + 0x1f291]
0xb707: mov     esi, 1
0xb70c: imul    edi, dword ptr [rip + 0x1f289]
0xb713: mov     qword ptr [rip + 0x1fc1e], rax
0xb71a: movsxd  rdi, edi
0xb71d: call    0x1d740
0xb722: mov     edi, dword ptr [rip + 0x1f270]
0xb728: mov     esi, 1
0xb72d: imul    edi, dword ptr [rip + 0x1f268]
0xb734: mov     qword ptr [rip + 0x1fbf5], rax
0xb73b: movsxd  rdi, edi
0xb73e: call    0x1d740
0xb743: mov     edi, dword ptr [rip + 0x1f24f]
0xb749: mov     esi, 1
0xb74e: imul    edi, dword ptr [rip + 0x1f247]
0xb755: mov     qword ptr [rip + 0x1fbcc], rax
0xb75c: movsxd  rdi, edi
0xb75f: call    0x1d740

```

```

0xb764: mov     qword ptr [rip + 0x1fbb5], rax
0xb76b: mov     rax, qword ptr [rsp + 0x48]
0xb770: sub     rax, qword ptr fs:[0x28]
0xb779: jne     0xb7dc
0xb77b: add     rsp, 0x58
0xb77f: pop     rbx
0xb780: pop     rbp
0xb781: pop     r12
0xb783: pop     r13
0xb785: pop     r14
0xb787: pop     r15
0xb789: ret
0xb78a: nop     word ptr [rax + rax]
0xb790: mov     dword ptr [rsp + 0xc], 0xffffffff
0xb798: xor     ebp, ebp
0xb79a: mov     r12d, 0xffffffff
0xb7a0: jmp     0xb5b2
0xb7a5: nop     dword ptr [rax]
0xb7a8: lea     edx, [rax - 1]
0xb7ab: mov     ebp, 0x18
0xb7b0: cmp     edx, 0x400
0xb7b6: cmovb   ebp, eax
0xb7b9: jmp     0xb64a
0xb7be: nop
0xb7c0: mov     ebx, dword ptr [rip + 0x1e72a]
0xb7c6: lea     edx, [rbx - 1]
0xb7c9: cmp     edx, 0x3ff
0xb7cf: mov     edx, 0x50
0xb7d4: cmova   ebx, edx
0xb7d7: jmp     0xb628
0xb7dc: call    0x49d0
0xb7e1: nop     word ptr cs:[rax + rax]
0xb7ec: nop     dword ptr [rax]
0xb7f0: endbr64
0xb7f4: push    r15
0xb7f6: push    r14
0xb7f8: push    r13
0xb7fa: push    r12
0xb7fc: mov     r12, rdi
0xb7ff: mov     edi, 0x5858
0xb804: push    rbp
0xb805: push    rbx
0xb806: mov     ebx, esi
0xb808: sub     rsp, 0x38
0xb80c: mov     rax, qword ptr fs:[0x28]
0xb815: mov     qword ptr [rsp + 0x28], rax
0xb81a: movabs  rax, 0x7363762f7665642f
0xb824: mov     word ptr [rsp + 8], di

```

```

0xb829: mov     byte ptr [rsp + 0xa], 0
0xb82e: mov     dword ptr [rsp + 0x18], 0x585861
0xb836: mov     qword ptr [rsp], rax
0xb83a: mov     qword ptr [rsp + 0x10], rax
0xb83f: cmp     esi, 1
0xb842: je      0xb978
0xb848: mov     esi, dword ptr [rip + 0x1fbb2]
0xb84e: mov     r13, qword ptr [rip + 0x1d783]
0xb855: mov     r14d, 1
0xb85b: test    esi, esi
0xb85d: jne     0xb90c
0xb863: mov     rax, qword ptr [rip + 0x1f126]
0xb86a: lea     r15, [rsp + 0x10]
0xb86f: mov     esi, r14d
0xb872: mov     dword ptr [rip + 0x1fb84], 0
0xb87c: mov     rdi, r15
0xb87f: movzx   edx, byte ptr [rax + 8]
0xb883: mov     byte ptr [rsp + 0x19], dl
0xb887: movzx   eax, byte ptr [rax + 9]
0xb88b: mov     byte ptr [rsp + 0x1a], al
0xb88f: call    0x4930
0xb894: mov     ebp, eax
0xb896: cmp     eax, -1
0xb899: je      0xb8f0
0xb89b: mov     eax, dword ptr [rip + 0x1f0fb]
0xb8a1: imul    eax, dword ptr [rip + 0x1f0f0]
0xb8a8: mov     edi, ebp
0xb8aa: mov     rsi, r12
0xb8ad: lea     edx, [rax + rax + 4]
0xb8b1: xor     eax, eax
0xb8b3: call    r13
0xb8b6: mov     edi, ebp
0xb8b8: call    0x4ab0
0xb8bd: cmp     ebx, 2
0xb8c0: je      0xb990
0xb8c6: mov     eax, 1
0xb8cb: mov     rdx, qword ptr [rsp + 0x28]
0xb8d0: sub     rdx, qword ptr fs:[0x28]
0xb8d9: jne     0xbb07
0xb8df: add     rsp, 0x38
0xb8e3: pop     rbx
0xb8e4: pop     rbp
0xb8e5: pop     r12
0xb8e7: pop     r13
0xb8e9: pop     r14
0xb8eb: pop     r15
0xb8ed: ret
0xb8ee: nop

```

```

0xb8f0: mov     ecx, 0x30
0xb8f5: mov     esi, r14d
0xb8f8: mov     rdi, r15
0xb8fb: mov     word ptr [rsp + 0x19], cx
0xb900: call    0x4930
0xb905: mov     ebp, eax
0xb907: cmp     eax, -1
0xb90a: jne     0xb89b
0xb90c: mov     dword ptr [rip + 0x1faea], 1
0xb916: cmp     ebx, 2
0xb919: je      0xba30
0xb91f: mov     rax, qword ptr [rip + 0x1f06a]
0xb926: mov     r15, rsp
0xb929: mov     esi, r14d
0xb92c: mov     rdi, r15
0xb92f: movzx   edx, byte ptr [rax + 8]
0xb933: movzx   eax, byte ptr [rax + 9]
0xb937: mov     byte ptr [rsp + 8], dl
0xb93b: mov     byte ptr [rsp + 9], al
0xb93f: call    0x4930
0xb944: mov     ebp, eax
0xb946: cmp     eax, -1
0xb949: je      0xba00
0xb94f: mov     edx, dword ptr [rip + 0x1f043]
0xb955: mov     edi, ebp
0xb957: mov     rsi, r12
0xb95a: xor     eax, eax
0xb95c: imul    edx, dword ptr [rip + 0x1f039]
0xb963: add     edx, 4
0xb966: call    r13
0xb969: mov     edi, ebp
0xb96b: call    0x4ab0
0xb970: jmp     0xb8c6
0xb975: nop     dword ptr [rax]
0xb978: mov     r13, qword ptr [rip + 0x1d661]
0xb97f: xor     r14d, r14d
0xb982: jmp     0xb863
0xb987: nop     word ptr [rax + rax]
0xb990: movsx   esi, byte ptr [r12 + 2]
0xb996: movsx   edi, byte ptr [r12 + 3]
0xb99c: call    0xa7a0
0xb9a1: mov     edx, dword ptr [rip + 0x1f561]
0xb9a7: test    edx, edx
0xb9a9: jle     0xb9e5
0xb9ab: xor     ebp, ebp
0xb9ad: lea     rbx, [rip + 0x1f56c]
0xb9b4: jmp     0xb9cc
0xb9b6: nop     word ptr cs:[rax + rax]

```

```

0xb9c0: mov     edx, dword ptr [rip + 0x1f542]
0xb9c6: add     ebp, eax
0xb9c8: cmp     edx, ebp
0xb9ca: jle     0xb9e5
0xb9cc: sub     edx, ebp
0xb9ce: movsxd  rsi, ebp
0xb9d1: mov     edi, 1
0xb9d6: add     rsi, rbx
0xb9d9: movsxd  rdx, edx
0xb9dc: call    0x18da0
0xb9e1: test    eax, eax
0xb9e3: jns     0xb9c0
0xb9e5: mov     dword ptr [rip + 0x1f519], 0
0xb9ef: mov     eax, 1
0xb9f4: jmp     0xb8cb
0xb9f9: nop     dword ptr [rax]
0xba00: mov     edx, 0x30
0xba05: mov     esi, r14d
0xba08: mov     rdi, r15
0xba0b: mov     word ptr [rsp + 8], dx
0xba10: call    0x4930
0xba15: mov     ebp, eax
0xba17: cmp     eax, -1
0xba1a: jne     0xb94f
0xba20: xor     eax, eax
0xba22: jmp     0xb8cb
0xba27: nop     word ptr [rax + rax]
0xba30: xor     eax, eax
0xba32: mov     r15, rsp
0xba35: call    0xa300
0xba3a: mov     rax, qword ptr [rip + 0x1ef4f]
0xba41: mov     esi, r14d
0xba44: mov     rdi, r15
0xba47: movzx   edx, byte ptr [rax + 8]
0xba4b: movzx   eax, byte ptr [rax + 9]
0xba4f: mov     byte ptr [rsp + 8], dl
0xba53: mov     byte ptr [rsp + 9], al
0xba57: call    0x4930
0xba5c: mov     ebp, eax
0xba5e: cmp     eax, -1
0xba61: je      0xbae0
0xba63: mov     edx, dword ptr [rip + 0x1ef2f]
0xba69: mov     rsi, r12
0xba6c: mov     edi, ebp
0xba6e: xor     eax, eax
0xba70: imul    edx, dword ptr [rip + 0x1ef25]
0xba77: add     edx, 4
0xba7a: call    r13

```

```

0xba7d: mov     edi, ebp
0xba7f: call    0x4ab0
0xba84: mov     eax, dword ptr [rip + 0x1ef12]
0xba8a: xor     esi, esi
0xba8c: lea     edi, [rax - 1]
0xba8f: call    0xa7a0
0xba94: mov     edx, dword ptr [rip + 0x1f46e]
0xba9a: test    edx, edx
0xba9c: jle     0xb9e5
0xbaa2: xor     ebp, ebp
0xbaa4: lea     rbx, [rip + 0x1f475]
0xbaab: jmp     0xbac0
0xbaad: nop     dword ptr [rax]
0xbab0: mov     edx, dword ptr [rip + 0x1f452]
0xbab6: add     ebp, eax
0xbab8: cmp     edx, ebp
0xbaba: jle     0xb9e5
0xbac0: sub     edx, ebp
0xbac2: movsxd  rsi, ebp
0xbac5: mov     edi, 1
0xbaca: add     rsi, rbx
0xbacd: movsxd  rdx, edx
0xbad0: call    0x18da0
0xbad5: test    eax, eax
0xbad7: jns     0xbab0
0xbad9: jmp     0xb9e5
0xbade: nop
0xbae0: mov     eax, 0x30
0xbae5: mov     esi, r14d
0xbae8: mov     rdi, r15
0xbaeb: mov     word ptr [rsp + 8], ax
0xbaf0: call    0x4930
0xbaf5: mov     ebp, eax
0xbaf7: cmp     eax, -1
0xbafa: jne     0xba63
0xbb00: xor     eax, eax
0xbb02: jmp     0xb8cb
0xbb07: call    0x49d0
0xbb0c: nop     dword ptr [rax]
0xbb10: endbr64
0xbb14: mov     eax, dword ptr [rip + 0x1ee6e]
0xbb1a: test    eax, eax
0xbb1c: jne     0xbb20
0xbb1e: ret
0xbb1f: nop
0xbb20: sub     rsp, 8
0xbb24: mov     esi, 1
0xbb29: xor     eax, eax

```



```

0xbb2b: call    0xb7f0
0xbb30: mov     dword ptr [rip + 0x1f8ce], eax
0xbb36: add     rsp, 8
0xbb3a: ret
0xbb3b: nop     dword ptr [rax + rax]
0xbb40: endbr64
0xbb44: push    rbp
0xbb45: xor     eax, eax
0xbb47: mov     rbp, rdi
0xbb4a: call    0xa6c0
0xbb4f: mov     edx, dword ptr [rip + 0x1ee33]
0xbb55: test    edx, edx
0xbb57: je      0xbb94
0xbb59: mov     eax, dword ptr [rip + 0x1f8a5]
0xbb5f: test    eax, eax
0xbb61: je      0xbb94
0xbb63: mov     edx, dword ptr [rip + 0x1ee33]
0xbb69: imul    edx, dword ptr [rip + 0x1ee28]
0xbb70: xor     esi, esi
0xbb72: mov     rdi, qword ptr [rip + 0x1f7af]
0xbb79: movsxd  rdx, edx
0xbb7c: call    0x4a50
0xbb81: xor     eax, eax
0xbb83: mov     esi, 2
0xbb88: mov     rdi, rbp
0xbb8b: call    0xb7f0
0xbb90: test    eax, eax
0xbb92: jne     0xbba0
0xbb94: xor     eax, eax
0xbb96: pop     rbp
0xbb97: jmp     0xa300
0xbb9c: nop     dword ptr [rax]
0xbba0: mov     edx, dword ptr [rip + 0x1edf6]
0xbba6: imul    edx, dword ptr [rip + 0x1edeb]
0xbbad: xor     esi, esi
0xbbae: mov     rdi, qword ptr [rip + 0x1f782]
0xbbb6: pop     rbp
0xbbb7: movsxd  rdx, edx
0xbbba: jmp     0x4a50
0xbbbf: nop
0xbbc0: endbr64
0xbbc4: push    rbp
0xbbc5: mov     rbp, rdi
0xbbc8: push    rbx
0xbbc9: sub     rsp, 8
0xbbcd: cmp     dword ptr [rip + 0x1ed84], 1
0xbbd4: je      0xbcd0
0xbbda: xor     eax, eax

```

```

0xbddc: call    0xa6c0
0xbbe1: test    rbp, rbp
0xbbe4: je      0xbbf4
0xbbe6: mov     eax, dword ptr [rip + 0x1ed9c]
0xbbec: test    eax, eax
0xbbee: jne     0xbcc0
0xbbf4: mov     eax, dword ptr [rip + 0x1eda2]
0xbbfa: mov     rdi, qword ptr [rip + 0x1e347]
0xbc01: lea     rdx, [rip - 0x2198]
0xbc08: lea     esi, [rax - 1]
0xbc0b: call    0x4820
0xbc10: mov     edx, dword ptr [rip + 0x1f2f2]
0xbc16: test    edx, edx
0xbc18: jle     0xbc4d
0xbc1a: xor     ebp, ebp
0xbc1c: lea     rbx, [rip + 0x1f2fd]
0xbc23: jmp     0xbc34
0xbc25: nop     dword ptr [rax]
0xbc28: mov     edx, dword ptr [rip + 0x1f2da]
0xbc2e: add     ebp, eax
0xbc30: cmp     edx, ebp
0xbc32: jle     0xbc4d
0xbc34: sub     edx, ebp
0xbc36: movsxd  rsi, ebp
0xbc39: mov     edi, 1
0xbc3e: add     rsi, rbx
0xbc41: movsxd  rdx, edx
0xbc44: call    0x18da0
0xbc49: test    eax, eax
0xbc4b: jns     0xbc28
0xbc4d: mov     edi, dword ptr [rip + 0x1ed49]
0xbc53: xor     esi, esi
0xbc55: mov     dword ptr [rip + 0x1f2a9], 0
0xbc5f: call    0xa7a0
0xbc64: mov     edx, dword ptr [rip + 0x1f29e]
0xbc6a: test    edx, edx
0xbc6c: jle     0xbca5
0xbc6e: xor     ebp, ebp
0xbc70: lea     rbx, [rip + 0x1f2a9]
0xbc77: jmp     0xbc8c
0xbc79: nop     dword ptr [rax]
0xbc80: mov     edx, dword ptr [rip + 0x1f282]
0xbc86: add     ebp, eax
0xbc88: cmp     edx, ebp
0xbc8a: jle     0xbca5
0xbc8c: sub     edx, ebp
0xbc8e: movsxd  rsi, ebp
0xbc91: mov     edi, 1

```

```

0xbc96: add     rsi, rbx
0xbc99: movsxd  rdx, edx
0xbc9c: call    0x18da0
0xbca1: test    eax, eax
0xbca3: jns     0xbc80
0xbca5: mov     dword ptr [rip + 0x1f259], 0
0xbcaf: add     rsp, 8
0xbcb3: mov     edi, 0xa
0xbcb8: pop     rbx
0xbcb9: pop     rbp
0xbcba: jmp     0x4810
0xbcbf: nop
0xbcc0: mov     rdi, rbp
0xbcc3: call    0xbb40
0xbcc8: jmp     0xbbf4
0xbccd: nop     dword ptr [rax]
0xbcd0: lea     rdx, [rip + 0x1f6e9]
0xbcd7: mov     esi, 1
0xbcdc: mov     edi, 1
0xbce1: call    0x4d70
0xbce6: mov     dword ptr [rip + 0x1ec68], 0
0xbcf0: jmp     0xbbda
0xbcf5: nop     word ptr cs:[rax + rax]
0xbd00: endbr64
0xbd04: push    r13
0xbd06: lea     r13, [rip + 0x1cc33]
0xbd0d: push    r12
0xbd0f: push    rbp
0xbd10: mov     rbp, rdi
0xbd13: lea     rdi, [rip + 0x14f14]
0xbd1a: push    rbx
0xbd1b: xor     ebx, ebx
0xbd1d: sub     rsp, 8
0xbd21: jmp     0xbd2d
0xbd23: nop     dword ptr [rax + rax]
0xbd28: mov     rdi, qword ptr [r13 + rbx*8]
0xbd2d: mov     rsi, rbp
0xbd30: mov     r12d, ebx
0xbd33: call    0x4b30
0xbd38: test    eax, eax
0xbd3a: je      0xbd60
0xbd3c: add     rbx, 1
0xbd40: cmp     rbx, 0xa
0xbd44: jne     0xbd28
0xbd46: add     rsp, 8
0xbd4a: mov     r12d, 0xffffffff
0xbd50: pop     rbx
0xbd51: mov     eax, r12d

```

```

0xbd54: pop      rbp
0xbd55: pop      r12
0xbd57: pop      r13
0xbd59: ret
0xbd5a: nop      word ptr [rax + rax]
0xbd60: lea      eax, [rbx - 8]
0xbd63: cmp      ebx, 7
0xbd66: cmovg    r12d, eax
0xbd6a: add      rsp, 8
0xbd6e: pop      rbx
0xbd6f: pop      rbp
0xbd70: mov      eax, r12d
0xbd73: pop      r12
0xbd75: pop      r13
0xbd77: ret
0xbd78: nop      dword ptr [rax + rax]
0xbd80: endbr64
0xbd84: push     r12
0xbd86: mov      r12, rdi
0xbd89: push     rbp
0xbd8a: lea      rbp, [rip + 0x1e1e7]
0xbd91: push     rbx
0xbd92: mov      ebx, 0x11
0xbd97: jmp      0xbdac
0xbd99: nop      dword ptr [rax]
0xbda0: add      ebx, 1
0xbda3: add      rbp, 0x20
0xbda7: cmp      ebx, 0x26
0xbdaa: je       0xbdd8
0xbdac: mov      rdi, qword ptr [rbp]
0xbdb0: mov      rsi, r12
0xbdb3: call     0x4b30
0xbdb8: test     eax, eax
0xbdba: jne      0xbda0
0xbdbc: movsxd   rbx, ebx
0xbdbf: lea      rax, [rip + 0x1df7a]
0xbdc6: shl      rbx, 5
0xbdca: mov      rax, qword ptr [rax + rbx + 8]
0xbdcf: pop      rbx
0xbdd0: pop      rbp
0xbdd1: pop      r12
0xbdd3: ret
0xbdd4: nop      dword ptr [rax]
0xbdd8: pop      rbx
0xbdd9: xor      eax, eax
0xbddb: pop      rbp
0xbddc: pop      r12
0xbdde: ret

```

```

0xbddf: nop
0xbde0: endbr64
0xbde4: push    r14
0xbde6: lea     rdi, [rip + 0x1479d]
0xbded: push    r13
0xbdef: push    r12
0xbdf1: push    rbp
0xbdf2: push    rbx
0xbdf3: sub     rsp, 0xb0
0xbdfa: mov     rax, qword ptr fs:[0x28]
0xbe03: mov     qword ptr [rsp + 0xa8], rax
0xbe0b: xor     eax, eax
0xbe0d: call    0x47e0
0xbe12: lea     rsi, [rsp + 0x10]
0xbe17: mov     edi, 1
0xbe1c: mov     rbp, rax
0xbe1f: call    0x4eb0
0xbe24: mov     rax, qword ptr [rsp + 0x38]
0xbe29: shr     rax, 8
0xbe2d: cmp     rax, 4
0xbe31: sete    al
0xbe34: movzx   eax, al
0xbe37: mov     dword ptr [rip + 0x1eb4b], eax
0xbe3d: test    rbp, rbp
0xbe40: je      0xc36d
0xbe46: mov     rdi, rbp
0xbe49: call    0x49b0
0xbe4e: cmp     rax, 0x3f
0xbe52: ja      0xc308
0xbe58: lea     r12, [rip + 0x14ff1]
0xbe5f: lea     rbx, [rip + 0x14d6f]
0xbe66: lea     r14, [rip + 0x14e02]
0xbe6d: lea     r13, [rip + 0x14e07]
0xbe74: nop     dword ptr [rax]
0xbe78: xor     edi, edi
0xbe7a: mov     rsi, rbp
0xbe7d: call    0x4e90
0xbe82: cmp     eax, -1
0xbe85: je      0xbf38
0xbe8b: test    eax, eax
0xbe8d: jne     0xbff8
0xbe93: mov     rdi, qword ptr [rip + 0x1e9e6]
0xbe9a: mov     esi, 1
0xbe9f: mov     r8, rbp
0xbea2: mov     rdx, r12
0xbea5: mov     rcx, qword ptr [rip + 0x1ea3c]
0xbeac: lea     r9, [rip + 0x1de5d]
0xbeb3: call    0x4e60

```

```

0xbeb8: mov     rsi, rbx
0xbebb: mov     rdi, rbp
0xbebe: call    0x4b30
0xbec3: test    eax, eax
0xbec5: je      0xbf98
0xbecb: mov     rdi, rbp
0xbece: call    0x9940
0xbed3: test    eax, eax
0xbed5: jne     0xbf01
0xbedb: mov     rsi, r14
0xbede: mov     rdi, rbp
0xbee1: call    0x4b30
0xbee6: test    eax, eax
0xbee8: je      0xbf01
0xbeea: lea     rsi, [rip + 0x14d84]
0xbef1: mov     rdi, rbp
0xbef4: call    0x4b30
0xbef9: test    eax, eax
0xbefb: jne     0xc3b4
0xbf01: mov     rdi, qword ptr [rip + 0x1e978]
0xbf08: mov     rcx, qword ptr [rip + 0x1e9d9]
0xbf0f: mov     esi, 1
0xbf14: xor     eax, eax
0xbf16: mov     rdx, r13
0xbf19: lea     rbp, [rip + 0x14cba]
0xbf20: call    0x4e60
0xbf25: xor     edi, edi
0xbf27: mov     rsi, rbp
0xbf2a: call    0x4e90
0xbf2f: cmp     eax, -1
0xbf32: jne     0xbe8b
0xbf38: mov     rcx, qword ptr [rip + 0x1e9a9]
0xbf3f: mov     rdi, qword ptr [rip + 0x1e93a]
0xbf46: xor     eax, eax
0xbf48: lea     r8, [rip + 0x1ddc1]
0xbf4f: lea     rdx, [rip + 0x14ea2]
0xbf56: mov     esi, 1
0xbf5b: call    0x4e60
0xbf60: mov     rdi, qword ptr [rip + 0x1e919]
0xbf67: xor     eax, eax
0xbf69: mov     rcx, qword ptr [rip + 0x1e978]
0xbf70: lea     r8, [rip + 0x1dd89]
0xbf77: lea     rdx, [rip + 0x14ea2]
0xbf7e: mov     esi, 1
0xbf83: call    0x4e60
0xbf88: mov     edi, 1
0xbf8d: call    0x4e20
0xbf92: nop     word ptr [rax + rax]

```

```

0xbf98: mov     rcx, qword ptr [rip + 0x1e949]
0xbf9f: mov     rdi, qword ptr [rip + 0x1e8da]
0xbfa6: lea     rdx, [rip + 0x14c99]
0xbfad: mov     esi, 1
0xbfb2: lea     rbp, [rip + 0x14c21]
0xbfb9: call    0x4e60
0xbfbe: jmp     0xbe78
0xbfc3: nop     dword ptr [rax + rax]
0xbfc8: mov     rcx, qword ptr [rip + 0x1e919]
0xbfcf: mov     rdi, qword ptr [rip + 0x1e8aa]
0xbfd6: lea     rdx, [rip + 0x14c7d]
0xbfdd: xor     eax, eax
0xbfdf: mov     esi, 1
0xbfe4: lea     rbp, [rip + 0x14bf6]
0xbfeb: call    0x4e60
0xbff0: jmp     0xbe78
0xbff5: nop     dword ptr [rax]
0xbff8: mov     rdi, rbp
0xbffb: lea     r13, [rsp + 8]
0xc000: call    0x1d7a0
0xc005: mov     edi, 0x800
0xc00a: mov     qword ptr [rip + 0x1e93f], rax
0xc011: call    0x1d670
0xc016: mov     rdi, qword ptr [rip + 0x1de03]
0xc01d: mov     rsi, r13
0xc020: mov     qword ptr [rsp + 8], rax
0xc025: call    0x4ca0
0xc02a: xor     edx, edx
0xc02c: test    rax, rax
0xc02f: je      0xc034
0xc031: movzx   edx, byte ptr [rax]
0xc034: mov     rdi, qword ptr [rip + 0x1de25]
0xc03b: mov     rsi, r13
0xc03e: mov     byte ptr [rip + 0x1e904], dl
0xc044: call    0x4ca0
0xc049: mov     rdi, qword ptr [rip + 0x1ddf0]
0xc050: mov     rsi, r13
0xc053: mov     qword ptr [rip + 0x1e8e6], rax
0xc05a: call    0x4ca0
0xc05f: cmp     qword ptr [rip + 0x1e8d9], 0
0xc067: mov     qword ptr [rip + 0x1e8ca], rax
0xc06e: je      0xc2c0
0xc074: test    rax, rax
0xc077: je      0xc2c0
0xc07d: mov     rdi, qword ptr [rip + 0x1dcbc]
0xc084: mov     rsi, r13
0xc087: call    0x4ca0
0xc08c: mov     rdi, qword ptr [rip + 0x1dced]

```

```

0xc093: mov     rsi, r13
0xc096: mov     qword ptr [rip + 0x1dcab], rax
0xc09d: call    0x4ca0
0xc0a2: mov     rdi, qword ptr [rip + 0x1dcb7]
0xc0a9: mov     rsi, r13
0xc0ac: mov     qword ptr [rip + 0x1dcd5], rax
0xc0b3: call    0x4ca0
0xc0b8: cmp     qword ptr [rip + 0x1dc88], 0
0xc0c0: mov     qword ptr [rip + 0x1dca1], rax
0xc0c7: je      0xc352
0xc0cd: mov     rdi, qword ptr [rip + 0x1ddac]
0xc0d4: mov     rsi, r13
0xc0d7: call    0x4ca0
0xc0dc: xor     edx, edx
0xc0de: mov     qword ptr [rip + 0x1dda3], rax
0xc0e5: test    rax, rax
0xc0e8: je      0xc0ff
0xc0ea: mov     rdi, qword ptr [rip + 0x1ddcf]
0xc0f1: mov     qword ptr [rip + 0x1dc70], rax
0xc0f8: call    0x4c90
0xc0fd: mov     edx, eax
0xc0ff: mov     rdi, qword ptr [rip + 0x1dd9a]
0xc106: mov     dword ptr [rip + 0x1ddc4], edx
0xc10c: call    0x4c50
0xc111: mov     dword ptr [rip + 0x1dd99], eax
0xc117: test    eax, eax
0xc119: js      0xc13c
0xc11b: mov     qword ptr [rip + 0x1dc62], 0
0xc126: mov     qword ptr [rip + 0x1dc37], 0
0xc131: mov     qword ptr [rip + 0x1dc0c], 0
0xc13c: mov     rdi, qword ptr [rip + 0x1dd9d]
0xc143: call    0x4c50
0xc148: mov     rdi, qword ptr [rip + 0x1ddb1]
0xc14f: mov     dword ptr [rip + 0x1dd9b], eax
0xc155: call    0x4c50
0xc15a: mov     rdi, qword ptr [rip + 0x1dc3f]
0xc161: mov     rsi, r13
0xc164: mov     dword ptr [rip + 0x1dda6], eax
0xc16a: call    0x4ca0
0xc16f: mov     rdi, qword ptr [rip + 0x1dc4a]
0xc176: mov     rsi, r13
0xc179: mov     qword ptr [rip + 0x1dc28], rax
0xc180: call    0x4ca0
0xc185: cmp     qword ptr [rip + 0x1dc1b], 0
0xc18d: mov     qword ptr [rip + 0x1dc34], rax
0xc194: je      0xc2a0
0xc19a: test    rax, rax
0xc19d: je      0xc2a0

```



```

0xc1a3: mov     rdi, qword ptr [rip + 0x1dc36]
0xc1aa: mov     rsi, r13
0xc1ad: lea     rbp, [rip + 0x1ddac]
0xc1b4: lea     rbx, [rbp - 0x220]
0xc1bb: call    0x4ca0
0xc1c0: mov     rdi, qword ptr [rip + 0x1dc39]
0xc1c7: mov     rsi, r13
0xc1ca: lea     r12, [rbx + 0x4c0]
0xc1d1: mov     qword ptr [rip + 0x1dc10], rax
0xc1d8: call    0x4ca0
0xc1dd: mov     rdi, qword ptr [rip + 0x1dd3c]
0xc1e4: mov     rsi, r13
0xc1e7: mov     qword ptr [rip + 0x1dc1a], rax
0xc1ee: call    0x4ca0
0xc1f3: mov     rdi, qword ptr [rip + 0x1dd46]
0xc1fa: mov     rsi, r13
0xc1fd: mov     qword ptr [rip + 0x1dd24], rax
0xc204: call    0x4ca0
0xc209: mov     qword ptr [rip + 0x1dd38], rax
0xc210: mov     rdi, qword ptr [rbp]
0xc214: mov     rsi, r13
0xc217: add     rbp, 0x20
0xc21b: call    0x4ca0
0xc220: mov     qword ptr [rbp - 0x18], rax
0xc224: cmp     rbp, r12
0xc227: jne     0xc210
0xc229: xor     ebp, ebp
0xc22b: lea     r13, [rip + 0x161b3]
0xc232: lea     r14, [rip + 0x14c57]
0xc239: jmp     0xc249
0xc23b: nop     dword ptr [rax + rax]
0xc240: add     rbx, 0x20
0xc244: cmp     rbx, r12
0xc247: je      0xc268
0xc249: cmp     qword ptr [rbx + 8], 0
0xc24e: jne     0xc240
0xc250: mov     eax, dword ptr [rbx + 0x14]
0xc253: test    eax, eax
0xc255: jne     0xc2e0
0xc25b: mov     qword ptr [rbx + 8], r13
0xc25f: add     rbx, 0x20
0xc263: cmp     rbx, r12
0xc266: jne     0xc249
0xc268: test    ebp, ebp
0xc26a: jne     0xc390
0xc270: mov     rax, qword ptr [rsp + 0xa8]
0xc278: sub     rax, qword ptr fs:[0x28]
0xc281: jne     0xc3be

```

```

0xc287: add    rsp, 0xb0
0xc28e: pop     rbx
0xc28f: pop     rbp
0xc290: pop     r12
0xc292: pop     r13
0xc294: pop     r14
0xc296: ret
0xc297: nop     word ptr [rax + rax]
0xc2a0: mov     qword ptr [rip + 0x1dafd], 0
0xc2ab: mov     qword ptr [rip + 0x1db12], 0
0xc2b6: jmp     0xc1a3
0xc2bb: nop     dword ptr [rax + rax]
0xc2c0: mov     qword ptr [rip + 0x1e66d], 0
0xc2cb: mov     qword ptr [rip + 0x1e66a], 0
0xc2d6: jmp     0xc07d
0xc2db: nop     dword ptr [rax + rax]
0xc2e0: mov     r8, qword ptr [rbx]
0xc2e3: mov     rcx, qword ptr [rip + 0x1e5fe]
0xc2ea: mov     rdx, r14
0xc2ed: xor     eax, eax
0xc2ef: mov     rdi, qword ptr [rip + 0x1e58a]
0xc2f6: mov     esi, 1
0xc2fb: add     ebp, 1
0xc2fe: call    0x4e60
0xc303: jmp     0xc240
0xc308: mov     rcx, qword ptr [rip + 0x1e5d9]
0xc30f: mov     rdi, qword ptr [rip + 0x1e56a]
0xc316: lea     rdx, [rip + 0x14aab]
0xc31d: xor     eax, eax
0xc31f: mov     esi, 1
0xc324: call    0x4e60
0xc329: mov     rcx, qword ptr [rip + 0x1e550]
0xc330: mov     edx, 0x11
0xc335: mov     esi, 1
0xc33a: lea     rdi, [rip + 0x148f3]
0xc341: lea     rbp, [rip + 0x1d9d1]
0xc348: call    0x4e40
0xc34d: jmp     0xbe58
0xc352: mov     qword ptr [rip + 0x1da2b], 0
0xc35d: mov     qword ptr [rip + 0x1da00], 0
0xc368: jmp     0xc0cd
0xc36d: mov     rcx, qword ptr [rip + 0x1e574]
0xc374: mov     rdi, qword ptr [rip + 0x1e505]
0xc37b: lea     rdx, [rip + 0x14a16]
0xc382: xor     eax, eax
0xc384: mov     esi, 1
0xc389: call    0x4e60
0xc38e: jmp     0xc329

```

```

0xc390: mov     rcx, qword ptr [rip + 0x1e551]
0xc397: mov     rdi, qword ptr [rip + 0x1e4e2]
0xc39e: mov     r8d, ebp
0xc3a1: xor     eax, eax
0xc3a3: lea     rdx, [rip + 0x14b16]
0xc3aa: mov     esi, 1
0xc3af: call    0x4e60
0xc3b4: mov     edi, 1
0xc3b9: call    0x4e20
0xc3be: call    0x49d0
0xc3c3: nop     word ptr cs:[rax + rax]
0xc3ce: nop
0xc3d0: endbr64
0xc3d4: push    rbx
0xc3d5: mov     ebx, edi
0xc3d7: xor     edi, edi
0xc3d9: call    0x48f0
0xc3de: test    eax, eax
0xc3e0: je      0xc453
0xc3e2: mov     edi, 1
0xc3e7: call    0x48f0
0xc3ec: test    eax, eax
0xc3ee: je      0xc453
0xc3f0: mov     edi, 1
0xc3f5: call    0x4cd0
0xc3fa: mov     qword ptr [rip + 0x1e58f], rax
0xc401: test    rax, rax
0xc404: je      0xc47e
0xc406: lea     rsi, [rip + 0x1efb3]
0xc40d: mov     edi, 1
0xc412: call    0x4d60
0xc417: mov     edi, 0x40
0xc41c: call    0x1d670
0xc421: mov     dword ptr [rip + 0x1efd5], ebx
0xc427: pop     rbx
0xc428: mov     qword ptr [rip + 0x1ef31], rax
0xc42f: mov     qword ptr [rip + 0x1e52a], rax
0xc436: xor     eax, eax
0xc438: mov     qword ptr [rip + 0x1e52d], 0
0xc443: mov     qword ptr [rip + 0x1e51a], 0
0xc44e: jmp     0xbde0
0xc453: mov     rdi, qword ptr [rip + 0x1e426]
0xc45a: mov     rcx, qword ptr [rip + 0x1e487]
0xc461: lea     rdx, [rip + 0x14a88]
0xc468: xor     eax, eax
0xc46a: mov     esi, 1
0xc46f: call    0x4e60
0xc474: mov     edi, 1

```

```

0xc479: call    0x4e20
0xc47e: mov     rdi, qword ptr [rip + 0x1e3fb]
0xc485: mov     rcx, qword ptr [rip + 0x1e45c]
0xc48c: mov     esi, 1
0xc491: lea     rdx, [rip + 0x147f8]
0xc498: call    0x4e60
0xc49d: mov     edi, 1
0xc4a2: call    0x4e20
0xc4a7: nop     word ptr [rax + rax]
0xc4b0: endbr64
0xc4b4: push    r12
0xc4b6: push    rbp
0xc4b7: mov     rbp, rdi
0xc4ba: mov     rdi, qword ptr [rip + 0x1e48f]
0xc4c1: push    rbx
0xc4c2: call    0x9940
0xc4c7: test    eax, eax
0xc4c9: jne     0xc4d0
0xc4cb: pop     rbx
0xc4cc: pop     rbp
0xc4cd: pop     r12
0xc4cf: ret
0xc4d0: mov     rdi, rbp
0xc4d3: call    0x49b0
0xc4d8: lea     rdi, [rax + 0x81]
0xc4df: mov     rbx, rax
0xc4e2: call    0x1d670
0xc4e7: mov     rdi, rbp
0xc4ea: mov     r12, rax
0xc4ed: call    0x1d7a0
0xc4f2: mov     rsi, rbx
0xc4f5: add     ebx, 0x7f
0xc4f8: mov     rbp, rax
0xc4fb: mov     rdi, rax
0xc4fe: call    0x184d0
0xc503: push    7
0xc505: lea     rcx, [rip + 0x147a2]
0xc50c: xor     eax, eax
0xc50e: push    rbp
0xc50f: mov     rdx, -1
0xc516: mov     esi, 1
0xc51b: mov     rdi, r12
0xc51e: lea     r9, [rip + 0x14040]
0xc525: mov     r8d, 0x1b
0xc52b: call    0x4f10
0xc530: mov     eax, dword ptr [rip + 0x1e462]
0xc536: pop     rdx
0xc537: pop     rcx

```

```

0xc538: add     eax, eax
0xc53a: cmp     ebx, eax
0xc53c: jl      0xc555
0xc53e: cdqe
0xc540: mov     byte ptr [r12 + rax], 7
0xc545: mov     eax, dword ptr [rip + 0x1e44d]
0xc54b: add     eax, eax
0xc54d: cdqe
0xc54f: mov     byte ptr [r12 + rax + 1], 0
0xc555: mov     rdi, r12
0xc558: call    0x49b0
0xc55d: mov     rsi, r12
0xc560: mov     edi, 1
0xc565: mov     rdx, rax
0xc568: call    0x18da0
0xc56d: mov     rdi, rbp
0xc570: call    0x18d10
0xc575: mov     rdi, r12
0xc578: call    0x18d10
0xc57d: mov     rdi, qword ptr [rip + 0x1e2bc]
0xc584: pop     rbx
0xc585: pop     rbp
0xc586: pop     r12
0xc588: jmp     0x4c40
0xc58d: nop     dword ptr [rax]
0xc590: endbr64
0xc594: sub     rsp, 8
0xc598: mov     edi, 0x18
0xc59d: call    0x1d670
0xc5a2: mov     qword ptr [rax], 0
0xc5a9: mov     qword ptr [rax + 8], 0
0xc5b1: add     rsp, 8
0xc5b5: ret
0xc5b6: nop     word ptr cs:[rax + rax]
0xc5c0: endbr64
0xc5c4: test    rdi, rdi
0xc5c7: je      0xc5d0
0xc5c9: jmp     0x18d10
0xc5ce: nop
0xc5d0: ret
0xc5d1: nop     word ptr cs:[rax + rax]
0xc5dc: nop     dword ptr [rax]
0xc5e0: endbr64
0xc5e4: mov     dword ptr [rdi], esi
0xc5e6: mov     dword ptr [rdi + 4], edx
0xc5e9: mov     dword ptr [rdi + 8], ecx
0xc5ec: mov     dword ptr [rdi + 0xc], r8d
0xc5f0: ret

```

```

0xc5f1: nop      word ptr cs:[rax + rax]
0xc5fc: nop      dword ptr [rax]
0xc600: endbr64
0xc604: mov      rax, rdi
0xc607: mov      rdi, rsi
0xc60a: mov      esi, edx
0xc60c: mov      ecx, dword ptr [rax + 0x10]
0xc60f: lea      r8d, [rcx + rdx]
0xc613: mov      edx, dword ptr [rax + 0xc]
0xc616: mov      dword ptr [rax + 0x10], r8d
0xc61a: cmp      edx, ecx
0xc61c: jle      0xc640
0xc61e: mov      r9d, dword ptr [rax + 8]
0xc622: cmp      dword ptr [rax + 0x14], r9d
0xc626: jge      0xc640
0xc628: cmp      r8d, edx
0xc62b: jle      0xc631
0xc62d: sub      edx, ecx
0xc62f: mov      esi, edx
0xc631: jmp      0xa1a0
0xc636: nop      word ptr cs:[rax + rax]
0xc640: xor      eax, eax
0xc642: ret
0xc643: nop      word ptr cs:[rax + rax]
0xc64e: nop
0xc650: endbr64
0xc654: mov      ecx, dword ptr [rdi + 0x10]
0xc657: lea      edx, [rcx + 1]
0xc65a: mov      dword ptr [rdi + 0x10], edx
0xc65d: cmp      edx, dword ptr [rdi + 0xc]
0xc660: jg       0xc66a
0xc662: mov      ecx, dword ptr [rdi + 8]
0xc665: cmp      dword ptr [rdi + 0x14], ecx
0xc668: jl       0xc670
0xc66a: xor      eax, eax
0xc66c: ret
0xc66d: nop      dword ptr [rax]
0xc670: mov      edi, esi
0xc672: jmp      0xa230
0xc677: nop      word ptr [rax + rax]
0xc680: endbr64
0xc684: mov      rax, rdi
0xc687: mov      edi, esi
0xc689: mov      dword ptr [rax + 0x10], edx
0xc68c: add      edx, dword ptr [rax]
0xc68e: mov      dword ptr [rax + 0x14], esi
0xc691: add      edi, dword ptr [rax + 4]
0xc694: mov      esi, edx

```

```

0xc696: jmp      0xa4b0
0xc69b: nop      dword ptr [rax + rax]
0xc6a0: endbr64
0xc6a4: mov      eax, dword ptr [rdi]
0xc6a6: ret
0xc6a7: nop      word ptr [rax + rax]
0xc6b0: endbr64
0xc6b4: mov      eax, dword ptr [rdi + 4]
0xc6b7: ret
0xc6b8: nop      dword ptr [rax + rax]
0xc6c0: endbr64
0xc6c4: mov      eax, dword ptr [rdi + 8]
0xc6c7: ret
0xc6c8: nop      dword ptr [rax + rax]
0xc6d0: endbr64
0xc6d4: mov      eax, dword ptr [rdi + 0xc]
0xc6d7: ret
0xc6d8: nop      dword ptr [rax + rax]
0xc6e0: push     r14
0xc6e2: push     r13
0xc6e4: push     r12
0xc6e6: mov      r12, qword ptr [rip + 0x1ed1b]
0xc6ed: push     rbp
0xc6ee: push     rbx
0xc6ef: mov      r13, qword ptr [r12 + 0x20]
0xc6f4: mov      rbx, qword ptr [r12 + 0x18]
0xc6f9: cmp      r13, rbx
0xc6fc: je       0xc761
0xc6fe: mov      ebp, edi
0xc700: jbe      0xc770
0xc702: test     bpl, 1
0xc706: je       0xc75c
0xc708: mov      rdi, qword ptr [r12 + 0x60]
0xc70d: test     rdi, rdi
0xc710: je       0xc71e
0xc712: call     0x18d10
0xc717: mov      r12, qword ptr [rip + 0x1ecea]
0xc71e: mov      r14, r13
0xc721: sub      r14, rbx
0xc724: lea      rdi, [r14 + 1]
0xc728: call     0x1d670
0xc72d: mov      rdx, r14
0xc730: mov      qword ptr [r12 + 0x60], rax
0xc735: mov      rax, qword ptr [rip + 0x1eccc]
0xc73c: mov      rsi, qword ptr [rax + 0x58]
0xc740: mov      rdi, qword ptr [rax + 0x60]
0xc744: add      rsi, rbx
0xc747: call     0x4bb0

```

```

0xc74c: mov     rax, qword ptr [rip + 0x1ecb5]
0xc753: mov     rax, qword ptr [rax + 0x60]
0xc757: mov     byte ptr [rax + r14], 0
0xc75c: and     ebp, 2
0xc75f: jne     0xc788
0xc761: pop     rbx
0xc762: pop     rbp
0xc763: pop     r12
0xc765: pop     r13
0xc767: pop     r14
0xc769: ret
0xc76a: nop     word ptr [rax + rax]
0xc770: mov     rax, rbx
0xc773: mov     rbx, r13
0xc776: mov     r13, rax
0xc779: test    bpl, 1
0xc77d: je      0xc75c
0xc77f: jmp     0xc708
0xc781: nop     dword ptr [rax]
0xc788: mov     rax, qword ptr [rip + 0x1ec79]
0xc78f: mov     rdx, rbx
0xc792: sub     rdx, r13
0xc795: add     qword ptr [rax + 0x48], rdx
0xc799: mov     rdi, qword ptr [rax + 0x58]
0xc79d: add     qword ptr [rax + 0x38], rdx
0xc7a1: mov     qword ptr [rax + 0x18], rbx
0xc7a5: lea     rsi, [rdi + r13]
0xc7a9: add     rdi, rbx
0xc7ac: mov     qword ptr [rax + 0x20], rbx
0xc7b0: call    0x48b0
0xc7b5: mov     rbx, qword ptr [rip + 0x1ec4c]
0xc7bc: mov     rax, qword ptr [rbx + 0x38]
0xc7c0: mov     rdi, qword ptr [rbx + 0x58]
0xc7c4: lea     rsi, [rax + 1]
0xc7c8: mov     qword ptr [rbx + 0x50], rsi
0xc7cc: call    0x1d690
0xc7d1: mov     qword ptr [rbx + 0x58], rax
0xc7d5: pop     rbx
0xc7d6: pop     rbp
0xc7d7: pop     r12
0xc7d9: pop     r13
0xc7db: pop     r14
0xc7dd: ret
0xc7de: nop
0xc7e0: endbr64
0xc7e4: test    rdi, rdi
0xc7e7: je      0xc818
0xc7e9: push    rbp

```



```

0xc7ea: mov     rbp, rdi
0xc7ed: mov     rdi, qword ptr [rdi + 0x58]
0xc7f1: test    rdi, rdi
0xc7f4: je      0xc7fb
0xc7f6: call    0x18d10
0xc7fb: mov     rdi, qword ptr [rbp + 0x60]
0xc7ff: test    rdi, rdi
0xc802: je      0xc809
0xc804: call    0x18d10
0xc809: mov     rdi, rbp
0xc80c: pop     rbp
0xc80d: jmp     0x18d10
0xc812: nop     word ptr [rax + rax]
0xc818: ret
0xc819: nop     dword ptr [rax]
0xc820: endbr64
0xc824: mov     rax, qword ptr [rip + 0x1ebdd]
0xc82b: push    rbp
0xc82c: mov     rdi, qword ptr [rax + 0x68]
0xc830: xor     eax, eax
0xc832: call    0x1aac0
0xc837: mov     rax, qword ptr [rip + 0x1ebca]
0xc83e: mov     rdi, qword ptr [rax]
0xc841: call    0xc5c0
0xc846: mov     rbp, qword ptr [rip + 0x1ebbb]
0xc84d: test    rbp, rbp
0xc850: je      0xc876
0xc852: mov     rdi, qword ptr [rbp + 0x58]
0xc856: test    rdi, rdi
0xc859: je      0xc860
0xc85b: call    0x18d10
0xc860: mov     rdi, qword ptr [rbp + 0x60]
0xc864: test    rdi, rdi
0xc867: je      0xc86e
0xc869: call    0x18d10
0xc86e: mov     rdi, rbp
0xc871: call    0x18d10
0xc876: mov     qword ptr [rip + 0x1eb87], 0
0xc881: pop     rbp
0xc882: ret
0xc883: nop     word ptr cs:[rax + rax]
0xc88e: nop
0xc890: endbr64
0xc894: mov     rax, qword ptr [rip + 0x1eb6d]
0xc89b: movsxd  rcx, edi
0xc89e: mov     edx, esi
0xc8a0: mov     r8, rcx
0xc8a3: mov     qword ptr [rax + 0x28], rcx

```

```

0xc8a7: movsxd rcx, esi
0xc8aa: mov     rdi, qword ptr [rax]
0xc8ad: xor     esi, esi
0xc8af: mov     qword ptr [rax + 0x30], rcx
0xc8b3: mov     ecx, 1
0xc8b8: jmp     0xc5e0
0xc8bd: nop     dword ptr [rax]
0xc8c0: endbr64
0xc8c4: push    r12
0xc8c6: mov     edi, 0x70
0xc8cb: push    rbx
0xc8cc: sub     rsp, 8
0xc8d0: call    0x1d670
0xc8d5: mov     rbx, qword ptr [rip + 0x1eb2c]
0xc8dc: mov     r12, rax
0xc8df: movdqu  xmm0, xmmword ptr [rbx]
0xc8e3: movups  xmmword ptr [rax], xmm0
0xc8e6: movdqu  xmm1, xmmword ptr [rbx + 0x10]
0xc8eb: movups  xmmword ptr [rax + 0x10], xmm1
0xc8ef: movdqu  xmm2, xmmword ptr [rbx + 0x20]
0xc8f4: movups  xmmword ptr [rax + 0x20], xmm2
0xc8f8: movdqu  xmm3, xmmword ptr [rbx + 0x30]
0xc8fd: movups  xmmword ptr [rax + 0x30], xmm3
0xc901: movdqu  xmm4, xmmword ptr [rbx + 0x40]
0xc906: movups  xmmword ptr [rax + 0x40], xmm4
0xc90a: movdqu  xmm5, xmmword ptr [rbx + 0x50]
0xc90f: movups  xmmword ptr [rax + 0x50], xmm5
0xc913: movdqu  xmm6, xmmword ptr [rbx + 0x60]
0xc918: cmp     qword ptr [rax + 0x58], 0
0xc91d: movups  xmmword ptr [rax + 0x60], xmm6
0xc921: je      0xc94a
0xc923: mov     rdi, qword ptr [rax + 0x50]
0xc927: call    0x1d670
0xc92c: mov     rdx, qword ptr [r12 + 0x50]
0xc931: mov     qword ptr [rbx + 0x58], rax
0xc935: mov     rax, qword ptr [rip + 0x1eacc]
0xc93c: mov     rsi, qword ptr [r12 + 0x58]
0xc941: mov     rdi, qword ptr [rax + 0x58]
0xc945: call    0x4bb0
0xc94a: mov     rdi, qword ptr [r12 + 0x60]
0xc94f: test    rdi, rdi
0xc952: je      0xc964
0xc954: mov     rbx, qword ptr [rip + 0x1eaad]
0xc95b: call    0x1d7a0
0xc960: mov     qword ptr [rbx + 0x60], rax
0xc964: add     rsp, 8
0xc968: mov     rax, r12
0xc96b: pop     rbx

```

```

0xc96c: pop      r12
0xc96e: ret
0xc96f: nop
0xc970: endbr64
0xc974: test     rdi, rdi
0xc977: je       0xc9f0
0xc979: push     r13
0xc97b: push     r12
0xc97d: push     rbp
0xc97e: push     rbx
0xc97f: mov      rbx, rdi
0xc982: sub      rsp, 8
0xc986: mov      rbp, qword ptr [rip + 0x1ea7b]
0xc98d: mov      rdi, qword ptr [rbp + 0x58]
0xc991: mov      r13, qword ptr [rbp + 0x28]
0xc995: mov      r12, qword ptr [rbp + 0x30]
0xc999: test     rdi, rdi
0xc99c: je       0xc9a3
0xc99e: call     0x18d10
0xc9a3: mov      rdi, qword ptr [rbp + 0x60]
0xc9a7: test     rdi, rdi
0xc9aa: je       0xc9b1
0xc9ac: call     0x18d10
0xc9b1: mov      rdi, rbp
0xc9b4: call     0x18d10
0xc9b9: movsxd   rax, r13d
0xc9bc: mov      edx, r12d
0xc9bf: movsxd   r12, r12d
0xc9c2: mov      qword ptr [rbx + 0x28], rax
0xc9c6: mov      rdi, qword ptr [rbx]
0xc9c9: mov      r8d, r13d
0xc9cc: mov      ecx, 1
0xc9d1: mov      qword ptr [rbx + 0x30], r12
0xc9d5: xor      esi, esi
0xc9d7: mov      qword ptr [rip + 0x1ea2a], rbx
0xc9de: add      rsp, 8
0xc9e2: pop      rbx
0xc9e3: pop      rbp
0xc9e4: pop      r12
0xc9e6: pop      r13
0xc9e8: jmp      0xc5e0
0xc9ed: nop      dword ptr [rax]
0xc9f0: ret
0xc9f1: nop      word ptr cs:[rax + rax]
0xc9fc: nop      dword ptr [rax]
0xca00: endbr64
0xca04: mov      rdx, qword ptr [rip + 0x1e9fd]
0xca0b: mov      rax, qword ptr [rdx + 0x18]

```

```

0xca0f: sub     rax, qword ptr [rdx + 0x40]
0xca13: ret
0xca14: nop     word ptr cs:[rax + rax]
0xca1f: nop
0xca20: endbr64
0xca24: mov     rax, qword ptr [rip + 0x1e9dd]
0xca2b: mov     r8d, dword ptr [rax + 8]
0xca2f: mov     dword ptr [rax + 8], edi
0xca32: mov     eax, r8d
0xca35: ret
0xca36: nop     word ptr cs:[rax + rax]
0xca40: endbr64
0xca44: mov     rax, qword ptr [rip + 0x1e9bd]
0xca4b: cmp     qword ptr [rax + 0x48], 0
0xca50: sete    al
0xca53: movzx   eax, al
0xca56: ret
0xca57: nop     word ptr [rax + rax]
0xca60: endbr64
0xca64: mov     rax, qword ptr [rip + 0x1e99d]
0xca6b: mov     rdx, qword ptr [rax + 0x18]
0xca6f: mov     qword ptr [rax + 0x20], rdx
0xca73: ret
0xca74: nop     word ptr cs:[rax + rax]
0xca7f: nop
0xca80: endbr64
0xca84: sub     rsp, 8
0xca88: xor     eax, eax
0xca8a: mov     edi, 3
0xca8f: call    0xc6e0
0xca94: mov     rax, qword ptr [rip + 0x1e96d]
0xca9b: mov     dword ptr [rax + 0x10], 0x18
0xcaa2: add     rsp, 8
0xcaa6: ret
0xcaa7: nop     word ptr [rax + rax]
0xcab0: endbr64
0xcab4: sub     rsp, 8
0xcab8: xor     eax, eax
0xcaba: mov     edi, 1
0xcabf: call    0xc6e0
0xcac4: mov     rax, qword ptr [rip + 0x1e93d]
0xcacb: mov     dword ptr [rax + 0x10], 0x19
0xcad2: add     rsp, 8
0xcad6: ret
0xcad7: nop     word ptr [rax + rax]
0xcae0: endbr64
0xcae4: mov     rax, qword ptr [rip + 0x1e91d]
0xcaeb: mov     rdx, qword ptr [rax + 0x18]

```

```

0xcaef: mov     rcx, qword ptr [rax + 0x20]
0xcaf3: movsxd  rdx, edx
0xcaf6: mov     qword ptr [rax + 0x18], rcx
0xcafa: mov     qword ptr [rax + 0x20], rdx
0xcafe: ret
0xcaff: nop
0xcb00: endbr64
0xcb04: mov     rax, qword ptr [rip + 0x1e8fd]
0xcb0b: mov     rdx, qword ptr [rax + 0x18]
0xcb0f: cmp     rdx, qword ptr [rax + 0x40]
0xcb13: jbe     0xcb24
0xcb15: sub     rdx, 1
0xcb19: mov     dword ptr [rax + 0x10], 1
0xcb20: mov     qword ptr [rax + 0x18], rdx
0xcb24: ret
0xcb25: nop     word ptr cs:[rax + rax]
0xcb30: endbr64
0xcb34: mov     rax, qword ptr [rip + 0x1e8cd]
0xcb3b: mov     rdx, qword ptr [rax + 0x18]
0xcb3f: cmp     rdx, qword ptr [rax + 0x38]
0xcb43: jae     0xcb54
0xcb45: add     rdx, 1
0xcb49: mov     dword ptr [rax + 0x10], 2
0xcb50: mov     qword ptr [rax + 0x18], rdx
0xcb54: ret
0xcb55: nop     word ptr cs:[rax + rax]
0xcb60: endbr64
0xcb64: mov     rdx, qword ptr [rip + 0x1e89d]
0xcb6b: mov     rax, qword ptr [rdx + 0x18]
0xcb6f: mov     rdi, qword ptr [rdx + 0x40]
0xcb73: cmp     rax, rdi
0xcb76: jbe     0xcc20
0xcb7c: mov     r8, qword ptr [rdx + 0x58]
0xcb80: jmp     0xcba9
0xcb82: nop     word ptr [rax + rax]
0xcb88: lea     ecx, [rsi - 0x30]
0xcb8b: cmp     cl, 9
0xcb8e: jbe     0xcbbf
0xcb90: sub     esi, 0x24
0xcb93: cmp     sil, 1
0xcb97: jbe     0xcbbf
0xcb99: mov     qword ptr [rdx + 0x18], rax
0xcb9d: mov     dword ptr [rdx + 0x10], 1
0xcba4: cmp     rdi, rax
0xcba7: jae     0xcbef
0xcba9: sub     rax, 1
0xcbad: movzx  esi, byte ptr [r8 + rax]
0xcbb2: mov     ecx, esi

```

```

0xcbb4: and     ecx, 0xffffffff
0xcbb7: sub     ecx, 0x41
0xcbb8: cmp     cl, 0x19
0xcbbd: ja      0xcb88
0xcbbf: mov     rax, qword ptr [rdx + 0x18]
0xcbc3: cmp     rdi, rax
0xcbc6: jae     0xcbef
0xcbc8: sub     rax, 1
0xcbcc: movzx   esi, byte ptr [r8 + rax]
0xcbd1: mov     ecx, esi
0xcbd3: and     ecx, 0xffffffff
0xcbd6: sub     ecx, 0x41
0xcbd9: cmp     cl, 0x19
0xcbd9: jbe     0xcc00
0xcbde: lea     ecx, [rsi - 0x30]
0xcbe1: cmp     cl, 9
0xcbe4: jbe     0xcc00
0xcbe6: sub     esi, 0x24
0xcbe9: cmp     sil, 1
0xcbed: jbe     0xcc00
0xcbef: mov     dword ptr [rdx + 0x10], 3
0xcbf6: ret
0xcbf7: nop     word ptr [rax + rax]
0xcc00: mov     qword ptr [rdx + 0x18], rax
0xcc04: mov     dword ptr [rdx + 0x10], 1
0xcc0b: cmp     rdi, rax
0xcc0e: jb      0xcbc8
0xcc10: mov     dword ptr [rdx + 0x10], 3
0xcc17: ret
0xcc18: nop     dword ptr [rax + rax]
0xcc20: ret
0xcc21: nop     word ptr cs:[rax + rax]
0xcc2c: nop     dword ptr [rax]
0xcc30: endbr64
0xcc34: mov     rdx, qword ptr [rip + 0x1e7cd]
0xcc3b: mov     rax, qword ptr [rdx + 0x18]
0xcc3f: mov     rdi, qword ptr [rdx + 0x38]
0xcc43: cmp     rax, rdi
0xcc46: jae     0xcce8
0xcc4c: mov     r8, qword ptr [rdx + 0x58]
0xcc50: jmp     0xcc7d
0xcc52: nop     word ptr [rax + rax]
0xcc58: lea     ecx, [rsi - 0x30]
0xcc5b: cmp     cl, 9
0xcc5e: jbe     0xcc8f
0xcc60: sub     esi, 0x24
0xcc63: cmp     sil, 1
0xcc67: jbe     0xcc8f

```

```

0xcc69: add    rax, 1
0xcc6d: mov    dword ptr [rdx + 0x10], 2
0xcc74: mov    qword ptr [rdx + 0x18], rax
0xcc78: cmp    rdi, rax
0xcc7b: je     0xccbb
0xcc7d: movzx  esi, byte ptr [r8 + rax]
0xcc82: mov    ecx, esi
0xcc84: and    ecx, 0xffffffff
0xcc87: sub    ecx, 0x41
0xcc8a: cmp    cl, 0x19
0xcc8d: ja     0xcc58
0xcc8f: mov    rax, qword ptr [rdx + 0x18]
0xcc93: cmp    rdi, rax
0xcc96: jbe    0xccbb
0xcc98: movzx  esi, byte ptr [r8 + rax]
0xcc9d: mov    ecx, esi
0xcc9f: and    ecx, 0xffffffff
0xcca2: sub    ecx, 0x41
0xcca5: cmp    cl, 0x19
0xcca8: jbe    0xcc8
0xccaa: lea    ecx, [rsi - 0x30]
0ccad: cmp    cl, 9
0ccb0: jbe    0xcc8
0ccb2: sub    esi, 0x24
0ccb5: cmp    sil, 1
0ccb9: jbe    0xcc8
0ccbb: mov    dword ptr [rdx + 0x10], 4
0ccc2: ret
0ccc3: nop    dword ptr [rax + rax]
0ccc8: add    rax, 1
0cccc: mov    dword ptr [rdx + 0x10], 2
0ccd3: mov    qword ptr [rdx + 0x18], rax
0ccd7: cmp    rdi, rax
0ccda: jne    0xcc98
0ccdc: mov    dword ptr [rdx + 0x10], 4
0xcce3: ret
0xcce4: nop    dword ptr [rax]
0xcce8: ret
0xcce9: nop    dword ptr [rax]
0ccf0: endbr64
0ccf4: mov    rax, qword ptr [rip + 0x1e70d]
0ccfb: mov    rdx, qword ptr [rax + 0x40]
0ccff: mov    dword ptr [rax + 0x10], 5
0xcd06: mov    qword ptr [rax + 0x18], rdx
0xcd0a: ret
0xcd0b: nop    dword ptr [rax + rax]
0xcd10: endbr64
0xcd14: mov    rax, qword ptr [rip + 0x1e6ed]

```

```

0xcd1b: mov     rdx, qword ptr [rax + 0x38]
0xcd1f: mov     dword ptr [rax + 0x10], 6
0xcd26: mov     qword ptr [rax + 0x18], rdx
0xcd2a: ret
0xcd2b: nop     dword ptr [rax + rax]
0xcd30: endbr64
0xcd34: push    rbp
0xcd35: push    rbx
0xcd36: mov     ebx, edi
0xcd38: sub     rsp, 8
0xcd3c: call    0x4f00
0xcd41: movsxd  rdx, ebx
0xcd44: mov     rax, qword ptr [rax]
0xcd47: test    byte ptr [rax + rdx*2 + 1], 0x40
0xcd4c: je      0xcdab
0xcd4e: mov     rbp, qword ptr [rip + 0x1e6b3]
0xcd55: mov     rsi, qword ptr [rbp + 0x38]
0xcd59: lea     rdx, [rsi + 1]
0xcd5d: cmp     rdx, qword ptr [rbp + 0x50]
0xcd61: jae     0xcdb8
0xcd63: mov     rax, qword ptr [rbp + 0x18]
0xcd67: mov     rcx, qword ptr [rbp + 0x58]
0xcd6b: sub     rdx, rax
0xcd6e: lea     rsi, [rcx + rax]
0xcd72: lea     rdi, [rcx + rax + 1]
0xcd77: call    0x4d40
0xcd7c: mov     rax, qword ptr [rip + 0x1e685]
0xcd83: mov     rdx, qword ptr [rax + 0x58]
0xcd87: mov     rax, qword ptr [rax + 0x18]
0xcd8b: mov     byte ptr [rdx + rax], bl
0xcd8e: mov     rax, qword ptr [rip + 0x1e673]
0xcd95: add     qword ptr [rax + 0x18], 1
0xcd9a: add     qword ptr [rax + 0x38], 1
0xcd9f: add     qword ptr [rax + 0x48], 1
0xcda4: mov     dword ptr [rax + 0x10], 7
0xcdab: add     rsp, 8
0xcdaf: pop     rbx
0xcdb0: pop     rbp
0xcdb1: ret
0xcdb2: nop     word ptr [rax + rax]
0xcdb8: add     rsi, 0x21
0xcdbc: mov     rdi, qword ptr [rbp + 0x58]
0xcdc0: mov     qword ptr [rbp + 0x50], rsi
0xcdc4: call    0x1d690
0xcdc9: mov     qword ptr [rbp + 0x58], rax
0xcdcd: mov     rbp, qword ptr [rip + 0x1e634]
0xcdd4: mov     rax, qword ptr [rbp + 0x38]
0xcdd8: lea     rdx, [rax + 1]

```



```

0xcd6c: jmp      0xcd63
0xcdde: nop
0xcde0: endbr64
0xcde4: mov     rax, qword ptr [rip + 0x1e61d]
0xcdeb: mov     rcx, qword ptr [rax + 0x18]
0xcdef: mov     rdx, qword ptr [rax + 0x38]
0xcdf3: cmp     rcx, rdx
0xcdf6: jb      0xce00
0xcdf8: ret
0xcdf9: nop     dword ptr [rax]
0xce00: push    rbx
0xce01: mov     rax, qword ptr [rax + 0x58]
0xce05: add     rdx, 1
0xce09: sub     rdx, rcx
0xce0c: lea     rsi, [rax + rcx + 1]
0xce11: lea     rdi, [rax + rcx]
0xce15: call    0x4bb0
0xce1a: mov     rbx, qword ptr [rip + 0x1e5e7]
0xce21: mov     rsi, qword ptr [rbx + 0x38]
0xce25: sub     qword ptr [rbx + 0x48], 1
0xce2a: lea     rax, [rsi - 1]
0xce2e: mov     qword ptr [rbx + 0x38], rax
0xce32: test    al, 0xf
0xce34: je      0xce40
0xce36: mov     dword ptr [rbx + 0x10], 8
0xce3d: pop     rbx
0xce3e: ret
0xce3f: nop
0xce40: mov     qword ptr [rbx + 0x50], rsi
0xce44: mov     rdi, qword ptr [rbx + 0x58]
0xce48: call    0x1d690
0xce4d: mov     qword ptr [rbx + 0x58], rax
0xce51: mov     rbx, qword ptr [rip + 0x1e5b0]
0xce58: mov     dword ptr [rbx + 0x10], 8
0xce5f: pop     rbx
0xce60: ret
0xce61: nop     word ptr cs:[rax + rax]
0xce6c: nop     dword ptr [rax]
0xce70: endbr64
0xce74: mov     rax, qword ptr [rip + 0x1e58d]
0xce7b: mov     rcx, qword ptr [rax + 0x18]
0xce7f: cmp     rcx, qword ptr [rax + 0x40]
0xce83: ja      0xce90
0xce85: ret
0xce86: nop     word ptr cs:[rax + rax]
0xce90: push    rbx
0xce91: mov     rdx, qword ptr [rax + 0x38]
0xce95: mov     rsi, qword ptr [rax + 0x58]

```

```

0xce99: add     rdx, 1
0xce9d: lea     rdi, [rsi + rcx - 1]
0xcea2: sub     rdx, rcx
0xcea5: add     rsi, rcx
0xcea8: call    0x4bb0
0xcead: mov     rbx, qword ptr [rip + 0x1e554]
0xceb4: mov     rsi, qword ptr [rbx + 0x38]
0xceb8: sub     qword ptr [rbx + 0x18], 1
0xcebd: sub     qword ptr [rbx + 0x48], 1
0xcec2: lea     rax, [rsi - 1]
0xcec6: mov     qword ptr [rbx + 0x38], rax
0xceca: test    al, 0xf
0xcecc: je      0xcee0
0xcece: mov     dword ptr [rbx + 0x10], 9
0xcd5: pop     rbx
0xcd6: ret
0xcd7: nop     word ptr [rax + rax]
0xcee0: mov     qword ptr [rbx + 0x50], rsi
0xcee4: mov     rdi, qword ptr [rbx + 0x58]
0xcee8: call    0x1d690
0xceed: mov     qword ptr [rbx + 0x58], rax
0xcef1: mov     rbx, qword ptr [rip + 0x1e510]
0xcef8: mov     dword ptr [rbx + 0x10], 9
0xceff: pop     rbx
0xcf00: ret
0xcf01: nop     word ptr cs:[rax + rax]
0xcf0c: nop     dword ptr [rax]
0xcf10: endbr64
0xcf14: push    rbp
0xcf15: push    rbx
0xcf16: sub     rsp, 8
0xcf1a: mov     r9, qword ptr [rip + 0x1e4e7]
0xcf21: mov     rax, qword ptr [r9 + 0x18]
0xcf25: mov     rbx, qword ptr [r9 + 0x20]
0xcf29: mov     qword ptr [r9 + 0x20], rax
0xcf2d: xor     eax, eax
0xcf2f: call    0xcc30
0xcf34: xor     eax, eax
0xcf36: mov     edi, 3
0xcf3b: mov     rbp, qword ptr [r9 + 0x18]
0xcf3f: call    0xc6e0
0xcf44: mov     rax, qword ptr [rip + 0x1e4bd]
0xcf4b: mov     rdx, qword ptr [rax + 0x18]
0xcf4f: cmp     rdx, rbx
0xcf52: jae     0xcf65
0xcf54: lea     rcx, [rdx + rbx]
0xcf58: sub     rcx, rbp
0xcf5b: cmp     rbx, rbp

```

```

0xcf5e: cmovae    rdx, rcx
0xcf62: mov        rbx, rdx
0xcf65: mov        qword ptr [rax + 0x20], rbx
0xcf69: mov        dword ptr [rax + 0x10], 0xa
0xcf70: add        rsp, 8
0xcf74: pop        rbx
0xcf75: pop        rbp
0xcf76: ret
0xcf77: nop        word ptr [rax + rax]
0xcf80: endbr64
0xcf84: push       rbp
0xcf85: push       rbx
0xcf86: sub        rsp, 8
0xcf8a: mov        rax, qword ptr [rip + 0x1e477]
0xcf91: mov        rbp, qword ptr [rax + 0x18]
0xcf95: mov        rbx, qword ptr [rax + 0x20]
0xcf99: mov        qword ptr [rax + 0x20], rbp
0xcf9d: xor        eax, eax
0xcf9f: call       0xcb60
0xcfa4: xor        eax, eax
0xcfa6: mov        edi, 3
0xcfab: call       0xc6e0
0xcfb0: mov        rax, qword ptr [rip + 0x1e451]
0xcfb7: mov        rdx, qword ptr [rax + 0x18]
0xcfb9: cmp        rdx, rbx
0xcfbe: jae        0xcfd1
0xcfc0: lea        rcx, [rdx + rbx]
0xcfc4: sub        rcx, rbp
0xcfc7: cmp        rbx, rbp
0xcfca: cmovae    rdx, rcx
0xcfce: mov        rbx, rdx
0xcfd1: mov        qword ptr [rax + 0x20], rbx
0xcfd5: mov        dword ptr [rax + 0x10], 0xb
0xcfdc: add        rsp, 8
0xcfe0: pop        rbx
0xcfe1: pop        rbp
0xcfe2: ret
0xcfe3: nop        word ptr cs:[rax + rax]
0xcfee: nop
0xcff0: endbr64
0xcff4: push       rbx
0xcff5: mov        rbx, qword ptr [rip + 0x1e40c]
0xcffc: mov        esi, 1
0xd001: mov        qword ptr [rbx + 0x18], 0
0xd009: mov        rdi, qword ptr [rbx + 0x58]
0xd00d: mov        qword ptr [rbx + 0x20], 0
0xd015: mov        qword ptr [rbx + 0x38], 0
0xd01d: mov        qword ptr [rbx + 0x40], 0

```

```

0xd025: mov     qword ptr [rbx + 0x48], 0
0xd02d: mov     qword ptr [rbx + 0x50], 1
0xd035: call    0x1d690
0xd03a: mov     qword ptr [rbx + 0x58], rax
0xd03e: mov     rax, qword ptr [rip + 0x1e3c3]
0xd045: mov     rax, qword ptr [rax + 0x58]
0xd049: mov     byte ptr [rax], 0
0xd04c: mov     rax, qword ptr [rip + 0x1e3b5]
0xd053: mov     dword ptr [rax + 0x10], 0xc
0xd05a: pop     rbx
0xd05b: ret
0xd05c: nop     dword ptr [rax]
0xd060: endbr64
0xd064: push    rbx
0xd065: mov     edi, 0x70
0xd06a: sub     rsp, 0x10
0xd06e: mov     rax, qword ptr fs:[0x28]
0xd077: mov     qword ptr [rsp + 8], rax
0xd07c: xor     eax, eax
0xd07e: call    0x1d670
0xd083: mov     qword ptr [rax + 8], 1
0xd08b: mov     qword ptr [rax + 0x58], 0
0xd093: mov     qword ptr [rax + 0x60], 0
0xd09b: mov     qword ptr [rax + 0x28], 0
0xd0a3: mov     qword ptr [rax + 0x30], 0
0xd0ab: mov     qword ptr [rip + 0x1e356], rax
0xd0b2: xor     eax, eax
0xd0b4: call    0xcff0
0xd0b9: mov     rbx, qword ptr [rip + 0x1e348]
0xd0c0: xor     eax, eax
0xd0c2: call    0xc590
0xd0c7: lea     rdi, [rip + 0x135e4]
0xd0ce: mov     qword ptr [rbx], rax
0xd0d1: call    0x17c50
0xd0d6: mov     ecx, 1
0xd0db: mov     rsi, rsp
0xd0de: mov     edx, 1
0xd0e3: lea     rdi, [rip + 0x13e42]
0xd0ea: call    0x17320
0xd0ef: mov     rdi, qword ptr [rsp]
0xd0f3: lea     rax, [rip + 0x13e24]
0xd0fa: mov     rbx, qword ptr [rip + 0x1e307]
0xd101: test    rdi, rdi
0xd104: cmove   rdi, rax
0xd108: xor     eax, eax
0xd10a: call    0x19940
0xd10f: lea     rdi, [rip + 0x1cb32]
0xd116: mov     qword ptr [rbx + 0x68], rax

```

```

0xd11a: mov     eax, dword ptr [rip + 0x1cb40]
0xd120: test    eax, eax
0xd122: lea     rax, [rip + 0x1cb07]
0xd129: cmov     rdi, rax
0xd12d: call    0x17c50
0xd132: lea     rdi, [rip + 0x1d0c7]
0xd139: mov     edx, 6
0xd13e: lea     rsi, [rip + 0x1d0db]
0xd145: call    0x17f70
0xd14a: xor     eax, eax
0xd14c: call    0x1a310
0xd151: mov     rax, qword ptr [rip + 0x1e2b0]
0xd158: mov     rdi, qword ptr [rax + 0x68]
0xd15c: xor     eax, eax
0xd15e: call    0x1a900
0xd163: nop     dword ptr [rax + rax]
0xd168: xor     eax, eax
0xd16a: call    0x1ab40
0xd16f: test    rax, rax
0xd172: jne     0xd168
0xd174: mov     edi, 0x200
0xd179: call    0x1a620
0xd17e: mov     rax, qword ptr [rsp + 8]
0xd183: sub     rax, qword ptr fs:[0x28]
0xd18c: jne     0xd194
0xd18e: add     rsp, 0x10
0xd192: pop     rbx
0xd193: ret
0xd194: call    0x49d0
0xd199: nop     dword ptr [rax]
0xd1a0: endbr64
0xd1a4: sub     rsp, 8
0xd1a8: mov     rax, qword ptr [rip + 0x1e259]
0xd1af: or      edi, 2
0xd1b2: mov     rdx, qword ptr [rax + 0x40]
0xd1b6: mov     dword ptr [rax + 0x10], 6
0xd1bd: mov     qword ptr [rax + 0x20], rdx
0xd1c1: mov     rdx, qword ptr [rax + 0x38]
0xd1c5: mov     qword ptr [rax + 0x18], rdx
0xd1c9: xor     eax, eax
0xd1cb: call    0xc6e0
0xd1d0: mov     rax, qword ptr [rip + 0x1e231]
0xd1d7: mov     rdx, qword ptr [rax + 0x18]
0xd1db: mov     dword ptr [rax + 0x10], 0xd
0xd1e2: mov     qword ptr [rax + 0x20], rdx
0xd1e6: add     rsp, 8
0xd1ea: ret
0xd1eb: nop     dword ptr [rax + rax]

```

```

0xd1f0: endbr64
0xd1f4: mov     rax, qword ptr [rip + 0x1e20d]
0xd1fb: push    rbx
0xd1fc: mov     edi, 3
0xd201: mov     rdx, qword ptr [rax + 0x18]
0xd205: mov     rsi, qword ptr [rax + 0x20]
0xd209: mov     dword ptr [rax + 0x10], 5
0xd210: mov     rcx, qword ptr [rax + 0x40]
0xd214: mov     qword ptr [rax + 0x20], rdx
0xd218: mov     rbx, rsi
0xd21b: mov     qword ptr [rax + 0x18], rcx
0xd21f: sub     rbx, rdx
0xd222: cmp     rsi, rdx
0xd225: cmovbe  rbx, rcx
0xd229: xor     eax, eax
0xd22b: call    0xc6e0
0xd230: mov     rdx, qword ptr [rip + 0x1e1d1]
0xd237: mov     rax, qword ptr [rdx + 0x38]
0xd23b: mov     dword ptr [rdx + 0x10], 0xe
0xd242: cmp     rax, rbx
0xd245: cmova   rax, rbx
0xd249: mov     qword ptr [rdx + 0x20], rax
0xd24d: pop     rbx
0xd24e: ret
0xd24f: nop
0xd250: endbr64
0xd254: mov     rax, qword ptr [rip + 0x1e1ad]
0xd25b: push    rbx
0xd25c: mov     edi, 3
0xd261: mov     rdx, qword ptr [rax + 0x18]
0xd265: mov     rbx, qword ptr [rax + 0x20]
0xd269: mov     dword ptr [rax + 0x10], 6
0xd270: mov     qword ptr [rax + 0x20], rdx
0xd274: mov     rdx, qword ptr [rax + 0x38]
0xd278: mov     qword ptr [rax + 0x18], rdx
0xd27c: xor     eax, eax
0xd27e: call    0xc6e0
0xd283: mov     rdx, qword ptr [rip + 0x1e17e]
0xd28a: mov     rax, qword ptr [rdx + 0x38]
0xd28e: mov     dword ptr [rdx + 0x10], 0xf
0xd295: cmp     rax, rbx
0xd298: cmova   rax, rbx
0xd29c: mov     qword ptr [rdx + 0x20], rax
0xd2a0: pop     rbx
0xd2a1: ret
0xd2a2: nop     word ptr cs:[rax + rax]
0xd2ad: nop     dword ptr [rax]
0xd2b0: endbr64

```

```

0xd2b4: mov     rax, qword ptr [rip + 0x1e14d]
0xd2bb: mov     rdx, qword ptr [rax + 0x58]
0xd2bf: mov     rax, qword ptr [rax + 0x18]
0xd2c3: cmp     byte ptr [rdx + rax], 0x20
0xd2c7: je      0xd2d0
0xd2c9: ret
0xd2ca: nop     word ptr [rax + rax]
0xd2d0: sub     rsp, 8
0xd2d4: nop     dword ptr [rax]
0xd2d8: xor     eax, eax
0xd2da: call    0xcde0
0xd2df: mov     rax, qword ptr [rip + 0x1e122]
0xd2e6: mov     rdx, qword ptr [rax + 0x58]
0xd2ea: mov     rcx, qword ptr [rax + 0x18]
0xd2ee: cmp     byte ptr [rdx + rcx], 0x20
0xd2f2: je      0xd2d8
0xd2f4: cmp     qword ptr [rax + 0x48], 0
0xd2f9: jne     0xd31d
0xd2fb: jmp     0xd324
0xd2fd: nop     dword ptr [rax]
0xd300: xor     eax, eax
0xd302: call    0xce70
0xd307: mov     rax, qword ptr [rip + 0x1e0fa]
0xd30e: cmp     qword ptr [rax + 0x48], 0
0xd313: je      0xd324
0xd315: mov     rdx, qword ptr [rax + 0x58]
0xd319: mov     rcx, qword ptr [rax + 0x18]
0xd31d: cmp     byte ptr [rdx + rcx - 1], 0x20
0xd322: je      0xd300
0xd324: mov     dword ptr [rax + 0x10], 0x11
0xd32b: add     rsp, 8
0xd32f: ret
0xd330: endbr64
0xd334: mov     rax, qword ptr [rip + 0x1e0cd]
0xd33b: mov     rdx, qword ptr [rax + 0x58]
0xd33f: mov     rax, qword ptr [rax + 0x18]
0xd343: cmp     byte ptr [rdx + rax], 0x20
0xd347: je      0xd350
0xd349: ret
0xd34a: nop     word ptr [rax + rax]
0xd350: sub     rsp, 8
0xd354: xor     eax, eax
0xd356: call    0xd2b0
0xd35b: mov     edi, 0x20
0xd360: call    0xcd30
0xd365: mov     rax, qword ptr [rip + 0x1e09c]
0xd36c: mov     dword ptr [rax + 0x10], 0x10
0xd373: add     rsp, 8

```

```

0xd377: ret
0xd378: nop      dword ptr [rax + rax]
0xd380: endbr64
0xd384: push     rbp
0xd385: push     rbx
0xd386: sub      rsp, 8
0xd38a: mov      rbp, qword ptr [rip + 0x1e077]
0xd391: mov      rbx, qword ptr [rbp + 0x18]
0xd395: cmp      rbx, qword ptr [rbp + 0x38]
0xd399: jb       0xd3a8
0xd39b: add      rsp, 8
0xd39f: pop      rbx
0xd3a0: pop      rbp
0xd3a1: ret
0xd3a2: nop      word ptr [rax + rax]
0xd3a8: xor      eax, eax
0xd3aa: call     0xcc30
0xd3af: cmp      rbx, qword ptr [rbp + 0x18]
0xd3b3: jae      0xd3e4
0xd3b5: call     0x4ef0
0xd3ba: nop      word ptr [rax + rax]
0xd3c0: mov      rdx, qword ptr [rbp + 0x58]
0xd3c4: mov      rcx, qword ptr [rax]
0xd3c7: add      rdx, rbx
0xd3ca: add      rbx, 1
0xd3ce: movsx    rsi, byte ptr [rdx]
0xd3d2: mov      ecx, dword ptr [rcx + rsi*4]
0xd3d5: mov      byte ptr [rdx], cl
0xd3d7: mov      rbp, qword ptr [rip + 0x1e02a]
0xd3de: cmp      qword ptr [rbp + 0x18], rbx
0xd3e2: ja       0xd3c0
0xd3e4: mov      dword ptr [rbp + 0x10], 0x12
0xd3eb: add      rsp, 8
0xd3ef: pop      rbx
0xd3f0: pop      rbp
0xd3f1: ret
0xd3f2: nop      word ptr cs:[rax + rax]
0xd3fd: nop      dword ptr [rax]
0xd400: endbr64
0xd404: push     rbp
0xd405: push     rbx
0xd406: sub      rsp, 8
0xd40a: mov      rbp, qword ptr [rip + 0x1dff7]
0xd411: mov      rbx, qword ptr [rbp + 0x18]
0xd415: cmp      rbx, qword ptr [rbp + 0x38]
0xd419: jb       0xd428
0xd41b: add      rsp, 8
0xd41f: pop      rbx

```



```

0xd420: pop      rbp
0xd421: ret
0xd422: nop      word ptr [rax + rax]
0xd428: xor      eax, eax
0xd42a: call     0xcc30
0xd42f: cmp      rbx, qword ptr [rbp + 0x18]
0xd433: jae      0xd464
0xd435: call     0x47c0
0xd43a: nop      word ptr [rax + rax]
0xd440: mov      rdx, qword ptr [rbp + 0x58]
0xd444: mov      rcx, qword ptr [rax]
0xd447: add      rdx, rbx
0xd44a: add      rbx, 1
0xd44e: movsx    rsi, byte ptr [rdx]
0xd452: mov      ecx, dword ptr [rcx + rsi*4]
0xd455: mov      byte ptr [rdx], cl
0xd457: mov      rbp, qword ptr [rip + 0x1dfaa]
0xd45e: cmp      qword ptr [rbp + 0x18], rbx
0xd462: ja       0xd440
0xd464: mov      dword ptr [rbp + 0x10], 0x13
0xd46b: add      rsp, 8
0xd46f: pop      rbx
0xd470: pop      rbp
0xd471: ret
0xd472: nop      word ptr cs:[rax + rax]
0xd47d: nop      dword ptr [rax]
0xd480: endbr64
0xd484: push     r14
0xd486: push     r13
0xd488: push     r12
0xd48a: push     rbp
0xd48b: mov      rbp, qword ptr [rip + 0x1df76]
0xd492: push     rbx
0xd493: mov      rbx, qword ptr [rbp + 0x18]
0xd497: cmp      rbx, qword ptr [rbp + 0x38]
0xd49b: jbe      0xd4b0
0xd49d: pop      rbx
0xd49e: pop      rbp
0xd49f: pop      r12
0xd4a1: pop      r13
0xd4a3: pop      r14
0xd4a5: ret
0xd4a6: nop      word ptr cs:[rax + rax]
0xd4b0: xor      eax, eax
0xd4b2: call     0xcc30
0xd4b7: cmp      rbx, qword ptr [rbp + 0x18]
0xd4bb: jae      0xd528
0xd4bd: call     0x4f00

```

```

0xd4c2: mov     r14d, 1
0xd4c8: mov     r13, rax
0xd4cb: jmp     0xd4f3
0xd4cd: nop     dword ptr [rax]
0xd4d0: call    0x47c0
0xd4d5: xor     r14d, r14d
0xd4d8: mov     rax, qword ptr [rax]
0xd4db: mov     eax, dword ptr [rax + r12*4]
0xd4df: mov     byte ptr [rbp], al
0xd4e2: mov     rbp, qword ptr [rip + 0x1df1f]
0xd4e9: add     rbx, 1
0xd4ed: cmp     qword ptr [rbp + 0x18], rbx
0xd4f1: jbe     0xd528
0xd4f3: mov     rax, qword ptr [rbp + 0x58]
0xd4f7: add     rax, rbx
0xd4fa: movsx   r12, byte ptr [rax]
0xd4fe: mov     rbp, rax
0xd501: mov     rax, qword ptr [r13]
0xd505: test    byte ptr [rax + r12*2], 8
0xd50a: je      0xd4e2
0xd50c: test    r14d, r14d
0xd50f: jne     0xd4d0
0xd511: call    0x4ef0
0xd516: mov     rax, qword ptr [rax]
0xd519: mov     eax, dword ptr [rax + r12*4]
0xd51d: mov     byte ptr [rbp], al
0xd520: jmp     0xd4e2
0xd522: nop     word ptr [rax + rax]
0xd528: mov     dword ptr [rbp + 0x10], 0x14
0xd52f: pop     rbx
0xd530: pop     rbp
0xd531: pop     r12
0xd533: pop     r13
0xd535: pop     r14
0xd537: ret
0xd538: nop     dword ptr [rax + rax]
0xd540: endbr64
0xd544: push    r12
0xd546: mov     r12, rdi
0xd549: push    rbp
0xd54a: push    rbx
0xd54b: call    0x49b0
0xd550: mov     rbp, qword ptr [rip + 0x1deb1]
0xd557: mov     rbx, rax
0xd55a: mov     rsi, qword ptr [rbp + 0x40]
0xd55e: sub     rax, rsi
0xd561: add     qword ptr [rbp + 0x18], rax
0xd565: add     qword ptr [rbp + 0x20], rax

```

```

0xd569: mov     rax, qword ptr [rbp + 0x48]
0xd56d: lea     rdx, [rax + 1]
0xd571: lea     r8, [rbx + rdx]
0xd575: cmp     r8, qword ptr [rbp + 0x50]
0xd579: ja      0xd5d8
0xd57b: mov     rdi, qword ptr [rbp + 0x58]
0xd57f: add     rsi, rdi
0xd582: add     rdi, rbx
0xd585: call    0x4d40
0xd58a: mov     rax, qword ptr [rip + 0x1de77]
0xd591: mov     rdx, rbx
0xd594: mov     rsi, r12
0xd597: mov     rdi, qword ptr [rax + 0x58]
0xd59b: call    0x4bb0
0xd5a0: mov     rax, qword ptr [rip + 0x1de61]
0xd5a7: mov     rsi, rbx
0xd5aa: mov     rdi, qword ptr [rax + 0x58]
0xd5ae: call    0x184d0
0xd5b3: mov     rax, qword ptr [rip + 0x1de4e]
0xd5ba: mov     qword ptr [rax + 0x40], rbx
0xd5be: add     rbx, qword ptr [rax + 0x48]
0xd5c2: mov     qword ptr [rax + 0x38], rbx
0xd5c6: mov     dword ptr [rax + 0x10], 0x15
0xd5cd: pop     rbx
0xd5ce: pop     rbp
0xd5cf: pop     r12
0xd5d1: ret
0xd5d2: nop     word ptr [rax + rax]
0xd5d8: mov     qword ptr [rbp + 0x50], r8
0xd5dc: mov     rdi, qword ptr [rbp + 0x58]
0xd5e0: mov     rsi, r8
0xd5e3: call    0x1d690
0xd5e8: mov     qword ptr [rbp + 0x58], rax
0xd5ec: mov     rbp, qword ptr [rip + 0x1de15]
0xd5f3: mov     rax, qword ptr [rbp + 0x48]
0xd5f7: mov     rsi, qword ptr [rbp + 0x40]
0xd5fb: lea     rdx, [rax + 1]
0xd5ff: jmp     0xd57b
0xd604: nop     word ptr cs:[rax + rax]
0xd60f: nop
0xd610: endbr64
0xd614: test    rdi, rdi
0xd617: je      0xd6f0
0xd61d: push    r12
0xd61f: push    rbp
0xd620: mov     rbp, rdi
0xd623: push    rbx
0xd624: call    0x49b0

```

```

0xd629: mov     r12, qword ptr [rip + 0x1ddd8]
0xd630: mov     rbx, rax
0xd633: mov     rax, qword ptr [r12 + 0x38]
0xd638: lea     rdx, [rax + 1]
0xd63c: mov     rax, qword ptr [r12 + 0x50]
0xd641: lea     rcx, [rbx + rdx]
0xd645: cmp     rcx, rax
0xd648: ja      0xd6c0
0xd64a: mov     rax, qword ptr [r12 + 0x18]
0xd64f: mov     rdi, qword ptr [r12 + 0x58]
0xd654: sub     rdx, rax
0xd657: lea     rsi, [rdi + rax]
0xd65b: add     rax, rbx
0xd65e: add     rdi, rax
0xd661: call    0x4d40
0xd666: mov     rax, qword ptr [rip + 0x1dd9b]
0xd66d: mov     rdx, rbx
0xd670: mov     rsi, rbp
0xd673: mov     rdi, qword ptr [rax + 0x18]
0xd677: add     rdi, qword ptr [rax + 0x58]
0xd67b: call    0x4bb0
0xd680: mov     rax, qword ptr [rip + 0x1dd81]
0xd687: mov     rsi, rbx
0xd68a: mov     rdi, qword ptr [rax + 0x18]
0xd68e: add     rdi, qword ptr [rax + 0x58]
0xd692: call    0x184d0
0xd697: mov     rax, qword ptr [rip + 0x1dd6a]
0xd69e: add     qword ptr [rax + 0x18], rbx
0xd6a2: add     qword ptr [rax + 0x38], rbx
0xd6a6: add     qword ptr [rax + 0x48], rbx
0xd6aa: mov     dword ptr [rax + 0x10], 0x16
0xd6b1: pop     rbx
0xd6b2: pop     rbp
0xd6b3: pop     r12
0xd6b5: ret
0xd6b6: nop     word ptr cs:[rax + rax]
0xd6c0: lea     rsi, [rax + rbx + 0x21]
0xd6c5: mov     rdi, qword ptr [r12 + 0x58]
0xd6ca: mov     qword ptr [r12 + 0x50], rsi
0xd6cf: call    0x1d690
0xd6d4: mov     qword ptr [r12 + 0x58], rax
0xd6d9: mov     r12, qword ptr [rip + 0x1dd28]
0xd6e0: mov     rax, qword ptr [r12 + 0x38]
0xd6e5: lea     rdx, [rax + 1]
0xd6e9: jmp     0xd64a
0xd6ee: nop
0xd6f0: ret
0xd6f1: nop     word ptr cs:[rax + rax]

```

```

0xd6fc: nop      dword ptr [rax]
0xd700: endbr64
0xd704: mov      rax, qword ptr [rip + 0x1dcfd]
0xd70b: mov      rdi, qword ptr [rax + 0x60]
0xd70f: test     rdi, rdi
0xd712: je       0xd730
0xd714: sub      rsp, 8
0xd718: call     0xd610
0xd71d: mov      rax, qword ptr [rip + 0x1dce4]
0xd724: mov      dword ptr [rax + 0x10], 0x1a
0xd72b: add      rsp, 8
0xd72f: ret
0xd730: ret
0xd731: nop      word ptr cs:[rax + rax]
0xd73c: nop      dword ptr [rax]
0xd740: endbr64
0xd744: mov      rax, qword ptr [rip + 0x1dcbd]
0xd74b: push     r12
0xd74d: push     rbp
0xd74e: push     rbx
0xd74f: mov      rbx, qword ptr [rax + 0x40]
0xd753: mov      r12, qword ptr [rax + 0x28]
0xd757: mov      eax, dword ptr [rax + 0xc]
0xd75a: lea      rbp, [rbx + 3]
0xd75e: test     eax, eax
0xd760: je       0xd7f0
0xd766: mov      edx, dword ptr [rip + 0x1caa4]
0xd76c: mov      esi, dword ptr [rip + 0x1ca9a]
0xd772: mov      edi, dword ptr [rip + 0x1ca9c]
0xd778: call     0xa570
0xd77d: mov      rcx, qword ptr [rip + 0x1dc84]
0xd784: cmp      r12, rbp
0xd787: jnb      0xd810
0xd78d: mov      rsi, qword ptr [rcx + 0x18]
0xd791: mov      rdx, qword ptr [rcx + 0x40]
0xd795: xor      ebx, ebx
0xd797: mov      rdi, qword ptr [rcx + 0x28]
0xd79b: mov      rax, rdi
0xd79e: sub      rax, rdx
0xd7a1: mov      edx, 1
0xd7a6: mov      r8, rax
0xd7a9: shr      r8, 2
0xd7ad: cmp      rax, 3
0xd7b1: cmova    edx, r8d
0xd7b5: xor      eax, eax
0xd7b7: cmp      rdi, rsi
0xd7ba: ja       0xd7d3
0xd7bc: mov      rax, rsi

```

```

0xd7bf: sub    rax, rdi
0xd7c2: movsxd  rdi, edx
0xd7c5: lea     r8d, [rdx + rax]
0xd7c9: xor     edx, edx
0xd7cb: div     rdi
0xd7ce: mov     eax, r8d
0xd7d1: sub     eax, edx
0xd7d3: mov     edx, esi
0xd7d5: mov     rdi, qword ptr [rcx]
0xd7d8: xor     esi, esi
0xd7da: sub     edx, eax
0xd7dc: call    0xc680
0xd7e1: cmp     r12, rbp
0xd7e4: jb      0xd840
0xd7e6: pop     rbx
0xd7e7: pop     rbp
0xd7e8: pop     r12
0xd7ea: ret
0xd7eb: nop     dword ptr [rax + rax]
0xd7f0: mov     edx, dword ptr [rip + 0x1ca0e]
0xd7f6: mov     esi, dword ptr [rip + 0x1ca04]
0xd7fc: mov     edi, dword ptr [rip + 0x1ca06]
0xd802: jmp     0xd778
0xd807: nop     word ptr [rax + rax]
0xd810: mov     rbx, qword ptr [rcx + 0x40]
0xd814: mov     rsi, qword ptr [rcx + 0x18]
0xd818: mov     qword ptr [rcx + 0x40], 0
0xd820: xor     edx, edx
0xd822: add     qword ptr [rcx + 0x58], rbx
0xd826: sub     rsi, rbx
0xd829: sub     qword ptr [rcx + 0x38], rbx
0xd82d: sub     qword ptr [rcx + 0x20], rbx
0xd831: movsxd  rbx, ebx
0xd834: mov     qword ptr [rcx + 0x18], rsi
0xd838: jmp     0xd797
0xd83d: nop     dword ptr [rax]
0xd840: mov     rax, qword ptr [rip + 0x1dbc1]
0xd847: movsxd  rbx, ebx
0xd84a: mov     qword ptr [rax + 0x40], rbx
0xd84e: sub     qword ptr [rax + 0x58], rbx
0xd852: add     qword ptr [rax + 0x38], rbx
0xd856: add     qword ptr [rax + 0x18], rbx
0xd85a: add     qword ptr [rax + 0x20], rbx
0xd85e: pop     rbx
0xd85f: pop     rbp
0xd860: pop     r12
0xd862: ret
0xd863: nop     word ptr cs:[rax + rax]

```

```

0xd86e: nop
0xd870: endbr64
0xd874: push    r15
0xd876: push    r14
0xd878: push    r13
0xd87a: push    r12
0xd87c: push    rbp
0xd87d: push    rbx
0xd87e: sub     rsp, 0x28
0xd882: mov     rax, qword ptr fs:[0x28]
0xd88b: mov     qword ptr [rsp + 0x18], rax
0xd890: mov     rax, qword ptr [rip + 0x1db71]
0xd897: lea     r14, [rsp + 0x17]
0xd89c: mov     rdi, r14
0xd89f: mov     rbx, qword ptr [rax + 0x40]
0xd8a3: mov     r13, qword ptr [rax + 0x28]
0xd8a7: call    0xa6a0
0xd8ac: mov     rax, qword ptr [rip + 0x1db55]
0xd8b3: add     rbx, 3
0xd8b7: mov     ecx, dword ptr [rax + 0xc]
0xd8ba: test    ecx, ecx
0xd8bc: je      0xda20
0xd8c2: mov     edx, dword ptr [rip + 0x1c948]
0xd8c8: mov     esi, dword ptr [rip + 0x1c93e]
0xd8ce: mov     edi, dword ptr [rip + 0x1c940]
0xd8d4: call    0xa570
0xd8d9: mov     rdx, qword ptr [rip + 0x1db28]
0xd8e0: cmp     r13, rbx
0xd8e3: jnb     0xda90
0xd8e9: mov     qword ptr [rsp + 8], 0
0xd8f2: mov     rax, qword ptr [rdx + 0x18]
0xd8f6: mov     rcx, qword ptr [rdx + 0x40]
0xd8fa: mov     rdi, qword ptr [rdx + 0x28]
0xd8fe: mov     r15, rdi
0xd901: sub     r15, rcx
0xd904: cmp     r15, 3
0xd908: jbe     0xda80
0xd90e: shr     r15, 2
0xd912: movsxd  rcx, r15d
0xd915: cmp     rdi, rax
0xd918: ja      0xda70
0xd91e: mov     edx, eax
0xd920: sub     rax, rdi
0xd923: sub     edx, edi
0xd925: add     r15d, edx
0xd928: xor     edx, edx
0xd92a: div     rcx
0xd92d: sub     r15d, edx

```

```

0xd930: mov     r12d, r15d
0xd933: call    0x1d670
0xd938: mov     esi, 0x20
0xd93d: mov     rbp, rax
0xd940: mov     rax, qword ptr [rip + 0x1dac1]
0xd947: mov     rdi, rbp
0xd94a: mov     rdx, qword ptr [rax + 0x28]
0xd94e: call    0x4a50
0xd953: mov     rax, qword ptr [rip + 0x1daae]
0xd95a: mov     edx, dword ptr [rax + 8]
0xd95d: test    edx, edx
0xd95f: je      0xda40
0xd965: mov     rdx, qword ptr [rax + 0x38]
0xd969: sub     rdx, qword ptr [rax + 0x40]
0xd96d: mov     rdi, rbp
0xd970: sub     rdx, r12
0xd973: mov     rsi, qword ptr [rax + 0x28]
0xd977: sub     rsi, qword ptr [rax + 0x40]
0xd97b: cmp     rdx, rsi
0xd97e: cmova   rdx, rsi
0xd982: mov     rsi, qword ptr [rax + 0x40]
0xd986: add     rsi, r12
0xd989: add     rsi, qword ptr [rax + 0x58]
0xd98d: call    0x4bb0
0xd992: mov     rax, qword ptr [rip + 0x1da6f]
0xd999: xor     edx, edx
0xd99b: xor     esi, esi
0xd99d: mov     rdi, qword ptr [rax]
0xd9a0: call    0xc680
0xd9a5: cmp     r13, rbx
0xd9a8: jae     0xdaf0
0xd9ae: mov     rax, qword ptr [rip + 0x1da53]
0xd9b5: mov     rsi, rbp
0xd9b8: mov     rdi, qword ptr [rax]
0xd9bb: mov     rdx, qword ptr [rax + 0x28]
0xd9bf: sub     edx, dword ptr [rax + 0x40]
0xd9c2: call    0xc600
0xd9c7: mov     rax, qword ptr [rip + 0x1da3a]
0xd9ce: xor     esi, esi
0xd9d0: mov     edx, dword ptr [rax + 0x18]
0xd9d3: mov     rdi, qword ptr [rax]
0xd9d6: sub     edx, r15d
0xd9d9: call    0xc680
0xd9de: cmp     r13, rbx
0xd9e1: jb      0xdac8
0xd9e7: mov     rdi, rbp
0xd9ea: call    0x18d10
0xd9ef: mov     rdi, r14

```



```

0xd9f2: call    0xa6b0
0xd9f7: mov     rax, qword ptr [rsp + 0x18]
0xd9fc: sub     rax, qword ptr fs:[0x28]
0xda05: jne     0xdb0b
0xda0b: add     rsp, 0x28
0xda0f: pop     rbx
0xda10: pop     rbp
0xda11: pop     r12
0xda13: pop     r13
0xda15: pop     r14
0xda17: pop     r15
0xda19: ret
0xda1a: nop     word ptr [rax + rax]
0xda20: mov     edx, dword ptr [rip + 0x1c7de]
0xda26: mov     esi, dword ptr [rip + 0x1c7d4]
0xda2c: mov     edi, dword ptr [rip + 0x1c7d6]
0xda32: jmp     0xd8d4
0xda37: nop     word ptr [rax + rax]
0xda40: mov     rsi, qword ptr [rax + 0x40]
0xda44: mov     rdx, qword ptr [rax + 0x38]
0xda48: mov     rdi, rbp
0xda4b: mov     rax, qword ptr [rax + 0x28]
0xda4f: sub     rdx, rsi
0xda52: sub     rdx, r12
0xda55: sub     rax, rsi
0xda58: mov     esi, 0x2a
0xda5d: cmp     rdx, rax
0xda60: cmova   rdx, rax
0xda64: call    0x4a50
0xda69: jmp     0xd992
0xda6e: nop
0xda70: xor     r12d, r12d
0xda73: xor     r15d, r15d
0xda76: jmp     0xd933
0xda7b: nop     dword ptr [rax + rax]
0xda80: mov     ecx, 1
0xda85: mov     r15d, 1
0xda8b: jmp     0xd915
0xda90: mov     r12, qword ptr [rdx + 0x40]
0xda94: mov     rax, qword ptr [rdx + 0x18]
0xda98: mov     qword ptr [rdx + 0x40], 0
0xdaa0: add     qword ptr [rdx + 0x58], r12
0xdaa4: movsxd rcx, r12d
0xdaa7: sub     rax, r12
0xdaaa: sub     qword ptr [rdx + 0x38], r12
0xdaae: mov     qword ptr [rsp + 8], rcx
0xdab3: xor     ecx, ecx
0xdab5: sub     qword ptr [rdx + 0x20], r12

```

```

0xdab9: mov     qword ptr [rdx + 0x18], rax
0xdabd: jmp     0xd8fa
0xdac2: nop     word ptr [rax + rax]
0xdac8: mov     rax, qword ptr [rip + 0x1d939]
0xdacf: movsxd  r12, dword ptr [rsp + 8]
0xdad4: sub     qword ptr [rax + 0x58], r12
0xdad8: add     qword ptr [rax + 0x38], r12
0xdadc: add     qword ptr [rax + 0x18], r12
0xdae0: add     qword ptr [rax + 0x20], r12
0xdae4: mov     qword ptr [rax + 0x40], r12
0xdae8: jmp     0xd9e7
0xdaed: nop     dword ptr [rax]
0xdaf0: mov     rax, qword ptr [rip + 0x1d911]
0xdaf7: mov     rsi, qword ptr [rax + 0x58]
0xdafb: mov     edx, dword ptr [rax + 0x40]
0xdafe: mov     rdi, qword ptr [rax]
0xdb01: call    0xc600
0xdb06: jmp     0xd9ae
0xdb0b: call    0x49d0
0xdb10: endbr64
0xdb14: mov     rax, qword ptr [rip + 0x1d8ed]
0xdb1b: push    rbx
0xdb1c: mov     rbx, rdi
0xdb1f: mov     rdi, qword ptr [rdi]
0xdb22: mov     rsi, qword ptr [rax + 0x48]
0xdb26: add     rsi, 1
0xdb2a: call    0x1d690
0xdb2f: mov     qword ptr [rbx], rax
0xdb32: mov     rdi, rax
0xdb35: mov     rax, qword ptr [rip + 0x1d8cc]
0xdb3c: mov     rcx, qword ptr [rax + 0x48]
0xdb40: mov     rsi, qword ptr [rax + 0x40]
0xdb44: add     rsi, qword ptr [rax + 0x58]
0xdb48: lea     rdx, [rcx + 1]
0xdb4c: call    0x4bb0
0xdb51: mov     rax, qword ptr [rip + 0x1d8b0]
0xdb58: pop     rbx
0xdb59: mov     eax, dword ptr [rax + 0x48]
0xdb5c: ret
0xdb5d: nop     dword ptr [rax]
0xdb60: endbr64
0xdb64: push    rbp
0xdb65: xor     eax, eax
0xdb67: mov     rbp, rdi
0xdb6a: call    0xcff0
0xdb6f: test    rbp, rbp
0xdb72: lea     rax, [rip + 0x133bf]
0xdb79: cmov    rbp, rax

```

```

0xdb7d: mov     rdi, rbp
0xdb80: call    0xd540
0xdb85: xor     eax, eax
0xdb87: call    0xd870
0xdb8c: xor     eax, eax
0xdb8e: pop     rbp
0xdb8f: jmp     0xd740
0xdb94: nop     word ptr cs:[rax + rax]
0xdb9f: nop
0xdba0: endbr64
0xdba4: mov     rax, qword ptr [rip + 0xd85d]
0xdbab: mov     dword ptr [rax + 0xc], edi
0xdbae: ret
0xdbaf: nop
0xdbb0: endbr64
0xdbb4: push    r12
0xdbb6: push    rbp
0xdbb7: push    rbx
0xdbb8: cmp     edi, 1
0xdbbb: je      0xdbd0
0xdbbd: cmp     edi, 2
0xdbc0: je      0xdc90
0xdbc6: test    edi, edi
0xdbc8: je      0xdc40
0xdbca: pop     rbx
0xdbc b: pop     rbp
0xdbcc: pop     r12
0xdbce: ret
0xdbcf: nop
0xdbd0: mov     ecx, dword ptr [rip + 0xd83e]
0xdbd6: test    ecx, ecx
0xdbd8: je      0xdd38
0xdbde: xor     eax, eax
0xdbe0: call    0x1ab10
0xdbe5: mov     rbp, rax
0xdbe8: test    rax, rax
0xdbeb: je      0xdbca
0xdbed: mov     r12, qword ptr [rbp]
0xdbf1: mov     rbx, qword ptr [rip + 0xd810]
0xdbf8: mov     rdi, r12
0xdbfb: call    0x49b0
0xdc00: mov     rdi, qword ptr [rbx + 0x40]
0xdc04: mov     qword ptr [rbx + 0x48], rax
0xdc08: lea     rsi, [rax + rdi]
0xdc0c: mov     qword ptr [rbx + 0x38], rsi
0xdc10: mov     qword ptr [rbx + 0x18], rsi
0xdc14: add     rsi, 1
0xdc18: cmp     rsi, qword ptr [rbx + 0x50]

```

```

0xdc1c: ja      0xdd70
0xdc22: add     rdi, qword ptr [rbx + 0x58]
0xdc26: mov     rsi, r12
0xdc29: call    0x48b0
0xdc2e: pop     rbx
0xdc2f: xor     eax, eax
0xdc31: pop     rbp
0xdc32: pop     r12
0xdc34: jmp     0xd870
0xdc39: nop     dword ptr [rax]
0xdc40: mov     eax, dword ptr [rip + 0x1d7ce]
0xdc46: test    eax, eax
0xdc48: jne     0xdd18
0xdc4e: xor     eax, eax
0xdc50: call    0x1ab10
0xdc55: test    rax, rax
0xdc58: je      0xdd98
0xdc5e: mov     rdx, qword ptr [rip + 0x1d7a3]
0xdc65: mov     rsi, qword ptr [rax]
0xdc68: mov     rbp, qword ptr [rdx + 0x40]
0xdc6c: add     rbp, qword ptr [rdx + 0x58]
0xdc70: mov     rdi, rbp
0xdc73: call    0x4b30
0xdc78: test    eax, eax
0xdc7a: jne     0xdd58
0xdc80: pop     rbx
0xdc81: xor     eax, eax
0xdc83: pop     rbp
0xdc84: pop     r12
0xdc86: jmp     0x1ab40
0xdc8b: nop     dword ptr [rax + rax]
0xdc90: mov     edx, dword ptr [rip + 0x1d77e]
0xdc96: test    edx, edx
0xdc98: je      0xdcf8
0xdc9a: xor     eax, eax
0xdc9c: call    0x1ab40
0xdca1: mov     rbp, rax
0xdca4: test    rax, rax
0xdca7: jne     0xdbed
0xdcad: mov     rax, qword ptr [rip + 0x1d754]
0xdcb4: mov     edi, 2
0xdcb9: mov     rdx, qword ptr [rax + 0x40]
0xdcbd: mov     dword ptr [rax + 0x10], 6
0xdcc4: mov     qword ptr [rax + 0x20], rdx
0xdcc8: mov     rdx, qword ptr [rax + 0x38]
0xdccc: mov     qword ptr [rax + 0x18], rdx
0xdcd0: xor     eax, eax
0xdcd2: call    0xc6e0

```

```

0xdcd7: mov     rax, qword ptr [rip + 0x1d72a]
0xdcde: mov     rdx, qword ptr [rax + 0x18]
0xdce2: mov     dword ptr [rax + 0x10], 0xd
0xdce9: mov     qword ptr [rax + 0x20], rdx
0xdcde: jmp     0xdc2e
0xdcf2: nop     word ptr [rax + rax]
0xdcf8: mov     dword ptr [rip + 0x1d712], 1
0xdd02: xor     eax, eax
0xdd04: call    0x1a570
0xdd09: mov     dword ptr [rip + 0x1d701], eax
0xdd0f: jmp     0xdc9a
0xdd11: nop     dword ptr [rax]
0xdd18: mov     edi, dword ptr [rip + 0x1d6f2]
0xdd1e: xor     eax, eax
0xdd20: call    0x1ac30
0xdd25: mov     dword ptr [rip + 0x1d6e5], 0
0xdd2f: jmp     0xdc4e
0xdd34: nop     dword ptr [rax]
0xdd38: mov     dword ptr [rip + 0x1d6d2], 1
0xdd42: xor     eax, eax
0xdd44: call    0x1a570
0xdd49: mov     dword ptr [rip + 0x1d6c1], eax
0xdd4f: jmp     0xdbde
0xdd54: nop     dword ptr [rax]
0xdd58: xor     eax, eax
0xdd5a: mov     rdi, rbp
0xdd5d: call    0x1a380
0xdd62: xor     eax, eax
0xdd64: call    0x1ab40
0xdd69: jmp     0xdc80
0xdd6e: nop
0xdd70: mov     qword ptr [rbx + 0x50], rsi
0xdd74: mov     rdi, qword ptr [rbx + 0x58]
0xdd78: call    0x1d690
0xdd7d: mov     qword ptr [rbx + 0x58], rax
0xdd81: mov     rbx, qword ptr [rip + 0x1d680]
0xdd88: mov     r12, qword ptr [rbp]
0xdd8c: mov     rdi, qword ptr [rbx + 0x40]
0xdd90: jmp     0xdc22
0xdd95: nop     dword ptr [rax]
0xdd98: mov     rax, qword ptr [rip + 0x1d669]
0xdd9f: mov     rdi, qword ptr [rax + 0x40]
0xdda3: add     rdi, qword ptr [rax + 0x58]
0xdda7: xor     eax, eax
0xdda9: call    0x1a380
0xddae: jmp     0xdc80
0xddb3: nop     word ptr cs:[rax + rax]
0xddbd: nop     dword ptr [rax]

```

```

0xddc0: endbr64
0xddc4: mov     eax, dword ptr [rip + 0x1be96]
0xddca: push    rbp
0xddcb: mov     rbp, rdi
0xddce: lea     rdi, [rip + 0x1be73]
0xddd5: test    eax, eax
0xddd7: lea     rax, [rip + 0x1be52]
0xddde: cmov    rdi, rax
0dde2: call    0x17c50
0dde7: mov     edx, 9
0ddeec: lea     rsi, [rip + 0x1c4ad]
0xddf3: lea     rdi, [rip + 0x1c466]
0xddfa: call    0x17f70
0xddff: mov     rdi, rbp
0xde02: call    0x1d7a0
0xde07: mov     rdi, rax
0xde0a: mov     rbp, rax
0xde0d: mov     qword ptr [rip + 0x1d794], rax
0xde14: call    0x49b0
0xde19: mov     rdi, rbp
0xde1c: mov     rsi, rax
0xde1f: call    0x184d0
0xde24: xor     eax, eax
0xde26: call    0xc590
0xde2b: lea     rdi, [rip + 0x1d5ee]
0xde32: pop     rbp
0xde33: mov     qword ptr [rip + 0x1d78e], rax
0xde3a: jmp     0x4a30
0xde3f: nop
0xde40: endbr64
0xde44: mov     rdi, qword ptr [rip + 0x1d77d]
0xde4b: jmp     0xc5c0
0xde50: endbr64
0xde54: push    rbp
0xde55: mov     rbp, rdi
0xde58: push    rbx
0xde59: mov     rbx, rsi
0xde5c: sub     rsp, 8
0xde60: mov     rdi, qword ptr [rip + 0x1d749]
0xde67: test    rdi, rdi
0xde6a: je      0xde71
0xde6c: call    0x18d10
0xde71: mov     rdi, rbp
0xde74: call    0x1d670
0xde79: mov     rdi, qword ptr [rip + 0x1d748]
0xde80: mov     edx, ebx
0xde82: xor     esi, esi
0xde84: mov     qword ptr [rip + 0x1d725], rax

```

```

0xde8b: add     rsp, 8
0xde8f: mov     r8d, ebp
0xde92: mov     ecx, 1
0xde97: pop     rbx
0xde98: pop     rbp
0xde99: jmp     0xc5e0
0xde9e: nop
0xdea0: endbr64
0xdea4: push    rbp
0xdea5: mov     rbp, rsp
0xdea8: push    r15
0xdeaa: push    r14
0xdeac: push    r13
0xdeae: push    r12
0xdeb0: push    rbx
0xdeb1: sub     rsp, 0x68
0xdeb5: mov     rax, qword ptr fs:[0x28]
0xdebe: mov     qword ptr [rbp - 0x38], rax
0xdec2: xor     eax, eax
0xdec4: lea     rax, [rbp - 0x61]
0xdec8: mov     rdi, rax
0xdecb: mov     qword ptr [rbp - 0x80], rax
0xdecf: call    0xa6a0
0xded4: mov     rax, qword ptr [rip + 0x1d6ed]
0xdedb: mov     esi, 0x20
0xdee0: mov     rdi, qword ptr [rip + 0x1d6c9]
0xdee7: movsxd  rdx, dword ptr [rax + 0xc]
0xdeeb: call    0x4a50
0xdef0: mov     rdi, qword ptr [rip + 0x1d6c9]
0xdef7: call    0x49b0
0xdefc: lea     r14, [rax + 1]
0xdf00: mov     rdi, r14
0xdf03: call    0x1d670
0xdf08: mov     rcx, qword ptr [rip + 0x1d6b1]
0xdf0f: mov     r13, rax
0xdf12: movzx   eax, byte ptr [rcx]
0xdf15: test    al, al
0xdf17: je      0xe430
0xdf1d: xor     esi, esi
0xdf1f: xor     ebx, ebx
0xdf21: lea     r15, [rip + 0x13188]
0xdf28: xor     edx, edx
0xdf2a: mov     dword ptr [rbp - 0x74], esi
0xdf2d: jmp     0xdf48
0xdf2f: nop
0xdf30: cmp     al, 9
0xdf32: je      0xdf80
0xdf34: add     dword ptr [rbp - 0x74], 1

```

```

0xdf38: mov     byte ptr [r13 + r12], al
0xdf3d: movsxd   rdx, ebx
0xdf40: movzx    eax, byte ptr [rcx + rdx]
0xdf44: test     al, al
0xdf46: je       0xdfc0
0xdf48: movsxd   r12, dword ptr [rbp - 0x74]
0xdf4c: add      ebx, 1
0xdf4f: cmp      al, 0x5c
0xdf51: jne      0xdf30
0xdf53: lea      rdx, [rcx + rdx + 1]
0xdf58: movzx    eax, byte ptr [rdx]
0xdf5b: test     al, al
0xdf5d: je       0xe320
0xdf63: sub      eax, 0x5c
0xdf66: cmp      al, 0x17
0xdf68: ja       0xe0d8
0xdf6e: movzx    eax, al
0xdf71: movsxd   rax, dword ptr [r15 + rax*4]
0xdf75: add      rax, r15
0xdf78: jmp      rax
0xdf7b: -----

0xdf7e: -----
0xdf7f: -----
0xdf80: add      r14, 8
0xdf84: mov      rdi, r13
0xdf87: mov      rsi, r14
0xdf8a: call     0x1d690
0xdf8f: mov      rcx, qword ptr [rip + 0x1d62a]
0xdf96: movsxd   rdx, ebx
0xdf99: add      dword ptr [rbp - 0x74], 8
0xdf9d: mov      r13, rax
0xdfa0: movabs   rax, 0x2020202020202020
0xdfaa: mov      qword ptr [r13 + r12], rax
0xdfaf: movzx    eax, byte ptr [rcx + rdx]
0xdfb3: test     al, al
0xdfb5: jne      0xdf48
0xdfb7: nop      word ptr [rax + rax]
0xdfc0: movsxd   rax, dword ptr [rbp - 0x74]
0xdfc4: add      rax, r13
0xdfc7: mov      byte ptr [rax], 0
0xdfca: mov      rdi, r13
0xdfcd: call     0x49b0
0xdfd2: cmp      byte ptr [rip + 0x1d5df], 0
0xdfd9: mov      rdi, qword ptr [rip + 0x1d5d0]
0xdfe0: mov      rdx, rax

```



```

0xdfe3: mov     rax, qword ptr [rip + 0x1d5de]
0xdfea: mov     ecx, edx
0xdfec: jne     0xe350
0xdff2: movsxd  rax, dword ptr [rax + 0xc]
0xdff6: cmp     edx, eax
0xdff8: jl      0xe120
0xdffe: cmp     ecx, eax
0xe000: mov     rsi, r13
0xe003: cmovle  eax, ecx
0xe006: movsxd  rdx, eax
0xe009: call    0x4bb0
0xe00e: mov     rdi, r13
0xe011: call    0x18d10
0xe016: mov     rsi, qword ptr [rip + 0x1d5ab]
0xe01d: mov     rcx, qword ptr [rip + 0x1d58c]
0xe024: xor     eax, eax
0xe026: mov     edx, dword ptr [rsi + 0xc]
0xe029: test    edx, edx
0xe02b: jg      0xe039
0xe02d: jmp     0xe054
0xe02f: nop
0xe030: add     rax, 1
0xe034: cmp     dword ptr [rsi + 0xc], eax
0xe037: jle     0xe054
0xe039: movzx   edx, byte ptr [rcx + rax]
0xe03d: cmp     dl, 0xd
0xe040: je      0xe047
0xe042: cmp     dl, 0xa
0xe045: jne     0xe030
0xe047: mov     byte ptr [rcx + rax], 0x20
0xe04b: add     rax, 1
0xe04f: cmp     dword ptr [rsi + 0xc], eax
0xe052: jg      0xe039
0xe054: movzx   eax, byte ptr [rip + 0x1d55e]
0xe05b: cmp     al, 1
0xe05d: je      0xe330
0xe063: cmp     al, 2
0xe065: jne     0xe100
0xe06b: mov     edx, dword ptr [rip + 0x1c20b]
0xe071: mov     esi, dword ptr [rip + 0x1c201]
0xe077: mov     edi, dword ptr [rip + 0x1c203]
0xe07d: call    0xa570
0xe082: mov     rdi, qword ptr [rip + 0x1d53f]
0xe089: xor     edx, edx
0xe08b: xor     esi, esi
0xe08d: call    0xc680
0xe092: mov     rdi, qword ptr [rip + 0x1d52f]
0xe099: mov     rsi, qword ptr [rip + 0x1d510]

```

```

0xe0a0: mov     edx, dword ptr [rdi + 0xc]
0xe0a3: call    0xc600
0xe0a8: mov     rdi, qword ptr [rbp - 0x80]
0xe0ac: call    0xa6b0
0xe0b1: mov     rax, qword ptr [rbp - 0x38]
0xe0b5: sub     rax, qword ptr fs:[0x28]
0xe0be: jne     0xe479
0xe0c4: lea     rsp, [rbp - 0x28]
0xe0c8: pop     rbx
0xe0c9: pop     r12
0xe0cb: pop     r13
0xe0cd: pop     r14
0xe0cf: pop     r15
0xe0d1: pop     rbp
0xe0d2: ret
0xe0d3: nop     dword ptr [rax + rax]
0xe0d8: mov     edi, dword ptr [rbp - 0x74]
0xe0db: mov     byte ptr [r13 + r12], 0x5c
0xe0e1: add     ebx, 1
0xe0e4: movzx   edx, byte ptr [rdx]
0xe0e7: lea     eax, [rdi + 1]
0xe0ea: add     edi, 2
0xe0ed: cdqe
0xe0ef: mov     dword ptr [rbp - 0x74], edi
0xe0f2: mov     byte ptr [r13 + rax], dl
0xe0f7: jmp     0xdf3d
0xe0fc: nop     dword ptr [rax]
0xe100: mov     edx, dword ptr [rip + 0x1c15e]
0xe106: mov     esi, dword ptr [rip + 0x1c154]
0xe10c: mov     edi, dword ptr [rip + 0x1c156]
0xe112: call    0xa570
0xe117: jmp     0xe082
0xe11c: nop     dword ptr [rax]
0xe120: sub     rax, rdx
0xe123: mov     rsi, r13
0xe126: shr     rax, 1
0xe129: add     rdi, rax
0xe12c: call    0x4bb0
0xe131: jmp     0xe00e
0xe136: nop     word ptr cs:[rax + rax]
0xe140: lea     rax, [rip + 0x1d2d9]
0xe147: lea     rcx, [rip + 0x1d2d2]
0xe14e: movzx   eax, byte ptr [rax]
0xe151: test    al, al
0xe153: je      0xe1d2
0xe155: mov     rdi, rcx
0xe158: mov     qword ptr [rbp - 0x90], rcx
0xe15f: call    0x49b0

```

```

0xe164: mov     rdi, r13
0xe167: add     r14, rax
0xe16a: mov     qword ptr [rbp - 0x88], rax
0xe171: mov     rsi, r14
0xe174: call    0x1d690
0xe179: mov     rdx, qword ptr [rbp - 0x88]
0xe180: mov     rcx, qword ptr [rbp - 0x90]
0xe187: add     r12, rax
0xe18a: mov     r13, rax
0xe18d: cmp     edx, 8
0xe190: jae     0xe360
0xe196: test    dl, 4
0xe199: jne     0xe438
0xe19f: test    edx, edx
0xe1a1: jne     0xe408
0xe1a7: add     dword ptr [rbp - 0x74], edx
0xe1aa: mov     rcx, qword ptr [rip + 0x1d40f]
0xe1b1: add     ebx, 1
0xe1b4: jmp     0xdf3d
0xe1b9: nop     dword ptr [rax]
0xe1c0: movzx   eax, byte ptr [rip + 0x1d35d]
0xe1c7: lea     rcx, [rip + 0x1d356]
0xe1ce: test    al, al
0xe1d0: jne     0xe155
0xe1d2: add     r14, 6
0xe1d6: mov     rdi, r13
0xe1d9: add     ebx, 1
0xe1dc: mov     rsi, r14
0xe1df: call    0x1d690
0xe1e4: mov     edi, 0x2965
0xe1e9: mov     edx, 6
0xe1ee: add     dword ptr [rbp - 0x74], edx
0xe1f1: add     r12, rax
0xe1f4: mov     rcx, qword ptr [rip + 0x1d3c5]
0xe1fb: mov     r13, rax
0xe1fe: mov     dword ptr [r12], 0x6e6f6e28
0xe206: mov     word ptr [r12 + 4], di
0xe20c: jmp     0xdf3d
0xe211: nop     dword ptr [rax]
0xe218: xor     eax, eax
0xe21a: call    0x18330
0xe21f: lea     r10, [rbp - 0x60]
0xe223: mov     edi, dword ptr [rax + 0x14]
0xe226: movsxd  rsi, dword ptr [rax + 0x10]
0xe22a: movsxd  rcx, dword ptr [rax + 0x18]
0xe22e: lea     edx, [rdi + 0x76c]
0xe234: mov     rdi, r10
0xe237: push    rdx

```

```

0xe238: mov     eax, dword ptr [rax + 0xc]
0xe23b: mov     edx, 0x20
0xe240: push    rax
0xe241: lea     rax, [rip + 0x1c4b8]
0xe248: mov     r9, qword ptr [rax + rsi*8]
0xe24c: lea     rax, [rip + 0x1c50d]
0xe253: mov     esi, 1
0xe258: mov     r8, qword ptr [rax + rcx*8]
0xe25c: lea     rcx, [rip + 0x12d65]
0xe263: xor     eax, eax
0xe265: mov     qword ptr [rbp - 0x88], r10
0xe26c: call    0x4f10
0xe271: mov     r10, qword ptr [rbp - 0x88]
0xe278: mov     rdi, r10
0xe27b: mov     qword ptr [rbp - 0x90], r10
0xe282: call    0x49b0
0xe287: mov     rdi, r13
0xe28a: add     r14, rax
0xe28d: mov     qword ptr [rbp - 0x88], rax
0xe294: mov     rsi, r14
0xe297: call    0x1d690
0xe29c: mov     rdx, qword ptr [rbp - 0x88]
0xe2a3: mov     r10, qword ptr [rbp - 0x90]
0xe2aa: add     r12, rax
0xe2ad: mov     r13, rax
0xe2b0: cmp     edx, 8
0xe2b3: jae     0xe3a8
0xe2b9: test    dl, 4
0xe2bc: jne     0xe44e
0xe2c2: test    edx, edx
0xe2c4: je      0xe2d7
0xe2c6: movzx   eax, byte ptr [rbp - 0x60]
0xe2ca: mov     byte ptr [r12], al
0xe2ce: test    dl, 2
0xe2d1: jne     0xe466
0xe2d7: pop     rcx
0xe2d8: add     dword ptr [rbp - 0x74], edx
0xe2db: add     ebx, 1
0xe2de: mov     rcx, qword ptr [rip + 0x1d2db]
0xe2e5: pop     rsi
0xe2e6: jmp     0xdf3d
0xe2eb: nop     dword ptr [rax + rax]
0xe2f0: add     dword ptr [rbp - 0x74], 1
0xe2f4: add     ebx, 1
0xe2f7: mov     byte ptr [r13 + r12], 0x5c
0xe2fd: jmp     0xdf3d
0xe302: nop     word ptr [rax + rax]
0xe308: movzx   eax, byte ptr [rip + 0x1d152]

```

```

0xe30f: lea      rcx, [rip + 0x1d14b]
0xe316: jmp      0xe151
0xe31b: nop
0xe320: add      dword ptr [rbp - 0x74], 1
0xe324: mov      byte ptr [r13 + r12], 0x5c
0xe32a: jmp      0xdf3d
0xe32f: nop
0xe330: mov      edx, dword ptr [rip + 0x1bf3a]
0xe336: mov      esi, dword ptr [rip + 0x1bf30]
0xe33c: mov      edi, dword ptr [rip + 0x1bf32]
0xe342: call     0xa570
0xe347: jmp      0xe082
0xe34c: nop
0xe350: mov      eax, dword ptr [rax + 0xc]
0xe353: jmp      0xdffe
0xe358: nop
0xe360: mov      rax, qword ptr [rcx]
0xe363: lea      rdi, [r12 + 8]
0xe368: add      ebx, 1
0xe36b: and      rdi, 0xfffffffffffffff8
0xe36f: mov      qword ptr [r12], rax
0xe373: mov      eax, edx
0xe375: mov      rsi, qword ptr [rcx + rax - 8]
0xe37a: mov      qword ptr [r12 + rax - 8], rsi
0xe37f: sub      r12, rdi
0xe382: mov      rsi, rcx
0xe385: sub      rsi, r12
0xe388: add      r12d, edx
0xe38b: shr      r12d, 3
0xe38f: mov      ecx, r12d
0xe392: rep movsq      qword ptr [rdi], qword ptr [rsi]
0xe395: add      dword ptr [rbp - 0x74], edx
0xe398: mov      rcx, qword ptr [rip + 0x1d221]
0xe39f: jmp      0xdf3d
0xe3a4: nop
0xe3a8: mov      rax, qword ptr [rbp - 0x60]
0xe3ac: lea      rsi, [r12 + 8]
0xe3b1: and      rsi, 0xfffffffffffffff8
0xe3b5: mov      qword ptr [r12], rax
0xe3b9: mov      eax, edx
0xe3bb: mov      rcx, qword ptr [r10 + rax - 8]
0xe3c0: mov      qword ptr [r12 + rax - 8], rcx
0xe3c5: sub      r12, rsi
0xe3c8: sub      r10, r12
0xe3cb: add      r12d, edx
0xe3ce: and      r12d, 0xffffffff
0xe3d2: cmp      r12d, 8
0xe3d6: jb      0xe2d7

```

```

0xe3dc: mov     r11d, dword ptr [rbp - 0x74]
0xe3e0: and     r12d, 0xffffffff8
0xe3e4: xor     eax, eax
0xe3e6: mov     ecx, eax
0xe3e8: add     eax, 8
0xe3eb: mov     rdi, qword ptr [r10 + rcx]
0xe3ef: mov     qword ptr [rsi + rcx], rdi
0xe3f3: cmp     eax, r12d
0xe3f6: jb      0xe3e6
0xe3f8: mov     dword ptr [rbp - 0x74], r11d
0xe3fc: jmp     0xe2d7
0xe401: nop     dword ptr [rax]
0xe408: movzx   eax, byte ptr [rcx]
0xe40b: mov     byte ptr [r12], al
0xe40f: test    dl, 2
0xe412: je      0xe1a7
0xe418: mov     eax, edx
0xe41a: movzx   ecx, word ptr [rcx + rax - 2]
0xe41f: mov     word ptr [r12 + rax - 2], cx
0xe425: jmp     0xe1a7
0xe42a: nop     word ptr [rax + rax]
0xe430: mov     rax, r13
0xe433: jmp     0xdfc7
0xe438: mov     eax, dword ptr [rcx]
0xe43a: mov     dword ptr [r12], eax
0xe43e: mov     eax, edx
0xe440: mov     ecx, dword ptr [rcx + rax - 4]
0xe444: mov     dword ptr [r12 + rax - 4], ecx
0xe449: jmp     0xe1a7
0xe44e: mov     eax, dword ptr [rbp - 0x60]
0xe451: mov     dword ptr [r12], eax
0xe455: mov     eax, edx
0xe457: mov     ecx, dword ptr [r10 + rax - 4]
0xe45c: mov     dword ptr [r12 + rax - 4], ecx
0xe461: jmp     0xe2d7
0xe466: mov     eax, edx
0xe468: movzx   ecx, word ptr [r10 + rax - 2]
0xe46e: mov     word ptr [r12 + rax - 2], cx
0xe474: jmp     0xe2d7
0xe479: call    0x49d0
0xe47e: nop
0xe480: endbr64
0xe484: push    r12
0xe486: mov     r12d, esi
0xe489: push    rbp
0xe48a: mov     rbp, rdi
0xe48d: mov     rdi, qword ptr [rip + 0x1d12c]
0xe494: push    rbx

```

```

0xe495: mov     ebx, edx
0xe497: test    rdi, rdi
0xe49a: je      0xe4a1
0xe49c: call    0x18d10
0xe4a1: mov     rdi, rbp
0xe4a4: call    0x1d7a0
0xe4a9: mov     rdi, rax
0xe4ac: mov     rbp, rax
0xe4af: mov     qword ptr [rip + 0x1d10a], rax
0xe4b6: call    0x49b0
0xe4bb: mov     rdi, rbp
0xe4be: mov     rsi, rax
0xe4c1: call    0x184d0
0xe4c6: mov     byte ptr [rip + 0x1d0ec], r12b
0xe4cd: xor     eax, eax
0xe4cf: mov     byte ptr [rip + 0x1d0e3], bl
0xe4d5: pop     rbx
0xe4d6: pop     rbp
0xe4d7: pop     r12
0xe4d9: jmp     0xdea0
0xe4de: nop
0xe4e0: endbr64
0xe4e4: push    rbp
0xe4e5: mov     rdi, qword ptr [rip + 0x1d0bc]
0xe4ec: call    0x1d7a0
0xe4f1: mov     rdi, qword ptr [rip + 0x1d0c8]
0xe4f8: mov     rbp, rax
0xe4fb: test    rdi, rdi
0xe4fe: je      0xe505
0xe500: call    0x18d10
0xe505: mov     rdi, rbp
0xe508: call    0x1d7a0
0xe50d: mov     rdi, rax
0xe510: mov     rbp, rax
0xe513: mov     qword ptr [rip + 0x1d0a6], rax
0xe51a: call    0x49b0
0xe51f: mov     rdi, rbp
0xe522: mov     rsi, rax
0xe525: call    0x184d0
0xe52a: xor     eax, eax
0xe52c: pop     rbp
0xe52d: mov     byte ptr [rip + 0x1d085], 0
0xe534: mov     byte ptr [rip + 0x1d07d], 0
0xe53b: jmp     0xdea0
0xe540: endbr64
0xe544: mov     rax, qword ptr [rsi]
0xe547: mov     rdx, qword ptr [rdi]
0xe54a: mov     rax, qword ptr [rax + 8]

```

```

0xe54e: sub     eax, dword ptr [rdx + 8]
0xe551: ret
0xe552: nop     word ptr cs:[rax + rax]
0xe55d: nop     dword ptr [rax]
0xe560: push    rbp
0xe561: mov     rbp, rdi
0xe564: mov     rdi, rdx
0xe567: push    rbx
0xe568: mov     ebx, esi
0xe56a: lea     rsi, [rip + 0x1bdf]
0xe571: sub     rsp, 8
0xe575: call    0x48b0
0xe57a: movsxd  rsi, ebx
0xe57d: mov     rcx, rax
0xe580: lea     rsi, [rsi + rsi*4]
0xe584: mov     rax, qword ptr [rbp + 0x38]
0xe588: shl     rsi, 4
0xe58c: mov     edx, dword ptr [rax + rsi + 0x10]
0xe590: mov     eax, 0x2d
0xe595: mov     edi, edx
0xe597: and     edi, 0xf000
0xe59d: cmp     edi, 0x8000
0xe5a3: je      0xe600
0xe5a5: mov     eax, 0x64
0xe5aa: cmp     edi, 0x4000
0xe5b0: je      0xe600
0xe5b2: mov     eax, 0x63
0xe5b7: cmp     edi, 0x2000
0xe5bd: je      0xe600
0xe5bf: mov     eax, 0x62
0xe5c4: cmp     edi, 0x6000
0xe5ca: je      0xe600
0xe5cc: mov     eax, 0x70
0xe5d1: cmp     edi, 0x1000
0xe5d7: je      0xe600
0xe5d9: mov     eax, 0x73
0xe5de: cmp     edi, 0xc000
0xe5e4: je      0xe600
0xe5e6: cmp     edi, 0xa000
0xe5ec: mov     eax, 0x6c
0xe5f1: mov     edi, 0x3f
0xe5f6: cmovne  eax, edi
0xe5f9: nop     dword ptr [rax]
0xe600: mov     byte ptr [rcx], al
0xe602: lea     rax, [rcx + 9]
0xe606: nop     word ptr cs:[rax + rax]
0xe610: test    dl, 1
0xe613: jne     0xe618

```



```

0xe615: mov     byte ptr [rax], 0x2d
0xe618: sub     rax, 1
0xe61c: shr     edx, 1
0xe61e: cmp     rax, rcx
0xe621: jne     0xe610
0xe623: mov     rax, qword ptr [rbp + 0x38]
0xe627: mov     eax, dword ptr [rax + rsi + 0x10]
0xe62b: test    ah, 8
0xe62e: je      0xe644
0xe630: cmp     byte ptr [rcx + 3], 0x78
0xe634: mov     edx, 0x73
0xe639: mov     esi, 0x53
0xe63e: cmovne  edx, esi
0xe641: mov     byte ptr [rcx + 3], dl
0xe644: test    ah, 4
0xe647: je      0xe65d
0xe649: cmp     byte ptr [rcx + 6], 0x78
0xe64d: mov     edx, 0x73
0xe652: mov     esi, 0x53
0xe657: cmovne  edx, esi
0xe65a: mov     byte ptr [rcx + 6], dl
0xe65d: test    ah, 2
0xe660: je      0xe676
0xe662: cmp     byte ptr [rcx + 9], 0x78
0xe666: mov     eax, 0x74
0xe66b: mov     edx, 0x54
0xe670: cmovne  eax, edx
0xe673: mov     byte ptr [rcx + 9], al
0xe676: add     rsp, 8
0xe67a: pop     rbx
0xe67b: pop     rbp
0xe67c: ret
0xe67d: nop     dword ptr [rax]
0xe680: endbr64
0xe684: push    r13
0xe686: push    r12
0xe688: push    rbp
0xe689: push    rbx
0xe68a: sub     rsp, 8
0xe68e: movzx   eax, byte ptr [rdi + 0x48]
0xe692: test    al, al
0xe694: sete    cl
0xe697: cmp     byte ptr [rsi + 0x48], 0
0xe69b: sete    dl
0xe69e: cmp     cl, dl
0xe6a0: je      0xe6c0
0xe6a2: cmp     al, 1
0xe6a4: sbb     r12d, r12d

```

```

0xe6a7: or      r12d, 1
0xe6ab: add      rsp, 8
0xe6af: mov      eax, r12d
0xe6b2: pop      rbx
0xe6b3: pop      rbp
0xe6b4: pop      r12
0xe6b6: pop      r13
0xe6b8: ret
0xe6b9: nop      dword ptr [rax]
0xe6c0: mov      r12d, dword ptr [rip + 0x1cf61]
0xe6c7: mov      rbx, rdi
0xe6ca: mov      rbp, rsi
0xe6cd: cmp      r12d, 8
0xe6d1: ja       0xe6e7
0xe6d3: lea      rdx, [rip + 0x131c6]
0xe6da: mov      eax, r12d
0xe6dd: movsxd   rax, dword ptr [rdx + rax*4]
0xe6e1: add      rax, rdx
0xe6e4: jmp      rax
0xe6e7: lea      rdi, [rip + 0x12a22]
0xe6ee: call     0x8b00
0xe6f3: mov      rbp, qword ptr [rbp]
0xe6f7: mov      r13, qword ptr [rbx]
0xe6fa: mov      rsi, rbp
0xe6fd: mov      rdi, r13
0xe700: add      rsp, 8
0xe704: pop      rbx
0xe705: pop      rbp
0xe706: pop      r12
0xe708: pop      r13
0xe70a: jmp      0x4b30
0xe70f: nop
0xe710: mov      eax, dword ptr [rsi + 0x14]
0xe713: cmp      dword ptr [rdi + 0x14], eax
0xe716: je       0xe6f3
0xe718: mov      rsi, qword ptr [rsi + 0x28]
0xe71c: mov      rdi, qword ptr [rdi + 0x28]
0xe720: jmp      0xe700
0xe722: nop      word ptr [rax + rax]
0xe728: mov      r12d, dword ptr [rdi + 0x18]
0xe72c: mov      eax, dword ptr [rsi + 0x18]
0xe72f: cmp      r12d, eax
0xe732: je       0xe6f3
0xe734: add      rsp, 8
0xe738: sub      r12d, eax
0xe73b: pop      rbx
0xe73c: mov      eax, r12d
0xe73f: pop      rbp

```

```

0xe740: pop      r12
0xe742: pop      r13
0xe744: ret
0xe745: nop      dword ptr [rax]
0xe748: mov      r12d, dword ptr [rdi + 0x14]
0xe74c: mov      eax, dword ptr [rsi + 0x14]
0xe74f: cmp      r12d, eax
0xe752: jne      0xe734
0xe754: jmp      0xe6f3
0xe756: nop      word ptr cs:[rax + rax]
0xe760: mov      r12d, dword ptr [rdi + 0x10]
0xe764: mov      eax, dword ptr [rsi + 0x10]
0xe767: cmp      r12d, eax
0xe76a: jne      0xe734
0xe76c: jmp      0xe6f3
0xe76e: nop
0xe770: mov      r12, qword ptr [rdi + 0x20]
0xe774: mov      rax, qword ptr [rsi + 0x20]
0xe778: cmp      r12, rax
0xe77b: jne      0xe734
0xe77d: jmp      0xe6f3
0xe782: nop      word ptr [rax + rax]
0xe788: mov      rdx, qword ptr [rdi + 8]
0xe78c: mov      rax, qword ptr [rsi + 8]
0xe790: mov      r12d, edx
0xe793: sub      r12d, eax
0xe796: cmp      rdx, rax
0xe799: jne      0xe6ab
0xe79f: jmp      0xe6f3
0xe7a4: nop      dword ptr [rax]
0xe7a8: mov      r13, qword ptr [rdi]
0xe7ab: mov      esi, 0x2e
0xe7b0: mov      rdi, r13
0xe7b3: call     0x4a20
0xe7b8: mov      rbp, qword ptr [rbp]
0xe7bc: mov      esi, 0x2e
0xe7c1: mov      rbx, rax
0xe7c4: mov      rdi, rbp
0xe7c7: call     0x4a20
0xe7cc: test     rbx, rbx
0xe7cf: je       0xe818
0xe7d1: test     rax, rax
0xe7d4: je       0xe818
0xe7d6: lea      rsi, [rax + 1]
0xe7da: lea      rdi, [rbx + 1]
0xe7de: call     0x4b30
0xe7e3: mov      r12d, eax
0xe7e6: test     eax, eax

```

```

0xe7e8: je      0xe6fa
0xe7ee: jmp      0xe6ab
0xe7f3: nop      dword ptr [rax + rax]
0xe7f8: mov      eax, dword ptr [rsi + 0x18]
0xe7fb: cmp      dword ptr [rdi + 0x18], eax
0xe7fe: je      0xe6f3
0xe804: mov      rsi, qword ptr [rsi + 0x30]
0xe808: mov      rdi, qword ptr [rdi + 0x30]
0xe80c: jmp      0xe700
0xe811: nop      dword ptr [rax]
0xe818: or       rax, rbx
0xe81b: je      0xe6fa
0xe821: test     rbx, rbx
0xe824: mov      eax, 0xffffffff
0xe829: cmovne   r12d, eax
0xe82d: jmp      0xe6ab
0xe832: nop      word ptr cs:[rax + rax]
0xe83d: nop      dword ptr [rax]
0xe840: push     r13
0xe842: mov      ecx, 0x12
0xe847: push     r12
0xe849: push     rbp
0xe84a: mov      rbp, rdi
0xe84d: push     rbx
0xe84e: sub      rsp, 0x138
0xe855: mov      rax, qword ptr fs:[0x28]
0xe85e: mov      qword ptr [rsp + 0x128], rax
0xe866: xor      eax, eax
0xe868: mov      r8, rsp
0xe86b: mov      rdi, r8
0xe86e: rep stosq      qword ptr [rdi], rax
0xe871: movsxd   rax, esi
0xe874: mov      rsi, r8
0xe877: lea      rbx, [rax + rax*4]
0xe87b: mov      rax, qword ptr [rbp + 0x38]
0xe87f: shl      rbx, 4
0xe883: mov      rdi, qword ptr [rax + rbx]
0xe887: call     0x18f90
0xe88c: mov      rax, qword ptr [rbp + 0x38]
0xe890: mov      edx, dword ptr [rsp + 0x18]
0xe894: mov      edi, dword ptr [rsp + 0x1c]
0xe898: mov      ecx, dword ptr [rsp + 0x20]
0xe89c: add      rax, rbx
0xe89f: cmp      qword ptr [rsp + 8], 0
0xe8a5: mov      dword ptr [rax + 0x10], edx
0xe8a8: mov      dword ptr [rax + 0x14], edi
0xe8ab: mov      dword ptr [rax + 0x18], ecx
0xe8ae: je      0xeab0

```

```

0xe8b4: mov     ecx, edx
0xe8b6: and     ecx, 0xf000
0xe8bc: cmp     ecx, 0x4000
0xe8c2: je      0xe908
0xe8c4: cmp     ecx, 0x8000
0xe8ca: je      0xeb88
0xe8d0: mov     byte ptr [rax + 0x4a], 0
0xe8d4: cmp     ecx, 0x1000
0xe8da: je      0xeb30
0xe8e0: cmp     ecx, 0xc000
0xe8e6: je      0xebb0
0xe8ec: cmp     ecx, 0xa000
0xe8f2: je      0xebc0
0xe8f8: mov     byte ptr [rax + 0x48], 1
0xe8fc: mov     rdx, qword ptr [rsp + 0x30]
0xe901: jmp     0xe911
0xe903: nop     dword ptr [rax + rax]
0xe908: mov     byte ptr [rax + 0x48], 0
0xe90c: mov     rdx, qword ptr [rsp + 0x30]
0xe911: mov     qword ptr [rax + 8], rdx
0xe915: call    0x1d310
0xe91a: mov     r12, rax
0xe91d: test    rax, rax
0xe920: je      0xea4e
0xe926: mov     rdi, rax
0xe929: call    0x49b0
0xe92e: mov     r13, qword ptr [rbp + 0x38]
0xe932: cmp     rax, 7
0xe936: lea     rdi, [rax + 1]
0xe93a: mov     eax, 9
0xe93f: cmovbe  rdi, rax
0xe943: add     r13, rbx
0xe946: call    0x1d670
0xe94b: mov     r8, r12
0xe94e: mov     esi, 1
0xe953: lea     rcx, [rip + 0x127c6]
0xe95a: mov     qword ptr [r13 + 0x28], rax
0xe95e: mov     rax, qword ptr [rbp + 0x38]
0xe962: mov     rdx, -1
0xe969: mov     rdi, qword ptr [rax + rbx + 0x28]
0xe96e: xor     eax, eax
0xe970: call    0x4f10
0xe975: mov     edi, dword ptr [rsp + 0x20]
0xe979: call    0x1d4c0
0xe97e: mov     r12, rax
0xe981: test    rax, rax
0xe984: je      0xeb40
0xe98a: mov     rdi, rax

```

```

0xe98d: call    0x49b0
0xe992: mov     r13, qword ptr [rbp + 0x38]
0xe996: cmp     rax, 7
0xe99a: lea     rdi, [rax + 1]
0xe99e: mov     eax, 9
0xe9a3: cmovbe  rdi, rax
0xe9a7: add     r13, rbx
0xe9aa: call    0x1d670
0xe9af: mov     r8, r12
0xe9b2: mov     esi, 1
0xe9b7: lea     rcx, [rip + 0x12762]
0xe9be: mov     qword ptr [r13 + 0x30], rax
0xe9c2: mov     rax, qword ptr [rbp + 0x38]
0xe9c6: mov     rdx, -1
0xe9cd: mov     rdi, qword ptr [rax + rbx + 0x30]
0xe9d2: xor     eax, eax
0xe9d4: call    0x4f10
0xe9d9: mov     rdx, qword ptr [rsp + 0x58]
0xe9de: mov     rax, qword ptr [rbp + 0x38]
0xe9e2: lea     rdi, [rsp + 0x58]
0xe9e7: mov     qword ptr [rax + rbx + 0x20], rdx
0xe9ec: call    0x4840
0xe9f1: mov     rcx, rax
0xe9f4: movsxd  rax, dword ptr [rax + 8]
0xe9f8: mov     r9d, dword ptr [rcx + 0xc]
0xe9fc: mov     rdx, rax
0xe9ff: imul    rax, rax, 0x2aaaaaab
0xea06: mov     esi, edx
0xea08: sar     esi, 0x1f
0xea0b: sar     rax, 0x21
0xea0f: sub     eax, esi
0xea11: lea     esi, [rax + rax*2]
0xea14: mov     eax, edx
0xea16: shl     esi, 2
0xea19: sub     eax, esi
0xea1b: mov     esi, 0xc
0xea20: cmovbe  eax, esi
0xea23: cmp     edx, 0xc
0xea26: mov     esi, 0x70
0xea2b: mov     edx, 0x61
0xea30: cmovge  edx, esi
0xea33: mov     esi, dword ptr [rcx + 0x10]
0xea36: add     rbx, qword ptr [rbp + 0x38]
0xea3a: lea     rdi, [rbx + 0x38]
0xea3e: push    rdx
0xea3f: mov     edx, dword ptr [rcx + 4]
0xea42: lea     r8d, [rsi + 1]
0xea46: mov     esi, 1

```

```

0xea4b: push    rdx
0xea4c: push    rax
0xea4d: movsxd  rax, dword ptr [rcx + 0x14]
0xea51: mov     rdx, rax
0xea54: imul    rax, rax, 0x51eb851f
0xea5b: mov     ecx, edx
0xea5d: sar     ecx, 0x1f
0xea60: sar     rax, 0x25
0xea64: sub     eax, ecx
0xea66: lea     rcx, [rip + 0x126bd]
0xea6d: imul    eax, eax, 0x64
0xea70: sub     edx, eax
0xea72: xor     eax, eax
0xea74: push    rdx
0xea75: mov     edx, 0x10
0xea7a: call    0x4f10
0xea7f: add     rsp, 0x20
0xea83: mov     rax, qword ptr [rsp + 0x128]
0xea8b: sub     rax, qword ptr fs:[0x28]
0xea94: jne     0xec3e
0xea9a: add     rsp, 0x138
0xeaa1: pop     rbx
0xeaa2: pop     rbp
0xeaa3: pop     r12
0xeaa5: pop     r13
0xeaa7: ret
0xeaa8: nop     dword ptr [rax + rax]
0xeab0: mov     byte ptr [rax + 0x48], 2
0xeab4: mov     rdi, qword ptr [rax]
0xeab7: call    0x18e80
0xeabc: xor     edx, edx
0xeabe: mov     edi, dword ptr [rsp + 0x1c]
0xeac2: cmp     eax, -1
0xeac5: cmov    eax, edx
0xeac8: mov     rdx, qword ptr [rbp + 0x38]
0xeacc: cdq     rdx
0xeace: mov     qword ptr [rdx + rbx + 8], rax
0xead3: call    0x1d310
0xead8: mov     r12, rax
0xeadb: test    rax, rax
0xeade: jne     0xe926
0xeae4: mov     r12, qword ptr [rbp + 0x38]
0xeae8: mov     edi, 0x20
0xeaed: call    0x1d670
0xeaf2: mov     r8d, dword ptr [rsp + 0x1c]
0xeaf7: mov     esi, 1
0xeafc: lea     rcx, [rip + 0x12622]
0xeb03: add     r12, rbx

```

```

0xeb06: mov     rdx, -1
0xeb0d: mov     qword ptr [r12 + 0x28], rax
0xeb12: mov     rax, qword ptr [rbp + 0x38]
0xeb16: mov     rdi, qword ptr [rax + rbx + 0x28]
0xeb1b: xor     eax, eax
0xeb1d: call    0x4f10
0xeb22: jmp     0xe975
0xeb27: nop     word ptr [rax + rax]
0xeb30: mov     byte ptr [rax + 0x48], 3
0xeb34: mov     rdx, qword ptr [rsp + 0x30]
0xeb39: jmp     0xe911
0xeb3e: nop
0xeb40: mov     r12, qword ptr [rbp + 0x38]
0xeb44: mov     edi, 0x20
0xeb49: call    0x1d670
0xeb4e: mov     r8d, dword ptr [rsp + 0x20]
0xeb53: mov     esi, 1
0xeb58: lea     rcx, [rip + 0x125c6]
0xeb5f: add     r12, rbx
0xeb62: mov     rdx, -1
0xeb69: mov     qword ptr [r12 + 0x30], rax
0xeb6e: mov     rax, qword ptr [rbp + 0x38]
0xeb72: mov     rdi, qword ptr [rax + rbx + 0x30]
0xeb77: xor     eax, eax
0xeb79: call    0x4f10
0xeb7e: jmp     0xe9d9
0xeb83: nop     dword ptr [rax + rax]
0xeb88: xor     ecx, ecx
0xeb8a: and     edx, 0x49
0xeb8d: mov     byte ptr [rax + 0x48], 1
0xeb91: je      0xeb9b
0xeb93: mov     edx, dword ptr [rbp + 0x10]
0xeb96: test    edx, edx
0xeb98: sete    cl
0xeb9b: mov     byte ptr [rax + 0x4a], cl
0xeb9e: mov     rdx, qword ptr [rsp + 0x30]
0xeba3: jmp     0xe911
0xeba8: nop     dword ptr [rax + rax]
0xebb0: mov     byte ptr [rax + 0x48], 4
0xebb4: mov     rdx, qword ptr [rsp + 0x30]
0xebb9: jmp     0xe911
0xebbe: nop
0xebc0: mov     rdi, qword ptr [rax]
0xebc3: lea     rsi, [rsp + 0x90]
0xebcb: call    0x18f40
0xebd0: cmp     eax, -1
0xebd3: je      0xec12
0xebd5: mov     ecx, dword ptr [rsp + 0xa8]

```



```

0xebdc: mov     rax, qword ptr [rbp + 0x38]
0xebe0: mov     edx, ecx
0xebe2: add     rax, rbx
0xebe5: and     edx, 0xf000
0xebeb: cmp     edx, 0x4000
0xebf1: mov     rdx, qword ptr [rsp + 0xc0]
0xebf9: setne   byte ptr [rax + 0x48]
0xebfd: and     ecx, 0x49
0xec00: mov     qword ptr [rsp + 0x30], rdx
0xec05: setne   byte ptr [rax + 0x4a]
0xec09: mov     edi, dword ptr [rsp + 0x1c]
0xec0d: jmp     0xe911
0xec12: mov     rax, qword ptr [rbp + 0x38]
0xec16: add     rax, rbx
0xec19: mov     byte ptr [rax + 0x48], 2
0xec1d: mov     rdi, qword ptr [rax]
0xec20: call    0x18e80
0xec25: xor     edx, edx
0xec27: cmp     eax, -1
0xec2a: cmov    eax, edx
0xec2d: movsxd  rdx, eax
0xec30: mov     rax, qword ptr [rbp + 0x38]
0xec34: mov     qword ptr [rsp + 0x30], rdx
0xec39: add     rax, rbx
0xec3c: jmp     0xec09
0xec3e: call    0x49d0
0xec43: nop     word ptr cs:[rax + rax]
0xec4e: nop
0xec50: push    r15
0xec52: push    r14
0xec54: push    r13
0xec56: push    r12
0xec58: push    rbp
0xec59: push    rbx
0xec5a: mov     rbx, rdi
0xec5d: sub     rsp, 0x6c8
0xec64: mov     rax, qword ptr fs:[0x28]
0xec6d: mov     qword ptr [rsp + 0x6b8], rax
0xec75: xor     eax, eax
0xec77: lea     rbp, [rsp + 0x1f]
0xec7c: mov     rdi, rbp
0xec7f: call    0xa6a0
0xec84: cmp     dword ptr [rbx + 0x60], 2
0xec88: jg      0xef20
0xec8e: mov     eax, dword ptr [rbx + 0x64]
0xec91: cmp     eax, 0x27
0xec94: jle     0xeeb8
0xec9a: mov     esi, dword ptr [rbx + 0x70]

```

```

0xec9d: test    esi, esi
0xec9f: je        0xee38
0xeca5: mov      edx, dword ptr [rbx + 0x68]
0xeca8: test     edx, edx
0xecaa: jle      0xf0e8
0xecb0: sub      edx, 1
0xecb3: mov      rcx, qword ptr [rbx + 0x38]
0xecb7: xor      edi, edi
0xecb9: lea      rdx, [rdx + rdx*4]
0xecbd: shl      rdx, 4
0xecc1: lea      rax, [rcx + 8]
0xecc5: lea      rdx, [rcx + rdx + 0x58]
0xecca: nop      word ptr [rax + rax]
0xecd0: cmp      byte ptr [rax + 0x41], 0
0xecd4: je       0xece0
0xecd6: cmp      byte ptr [rax + 0x40], 1
0xecda: jne      0xece0
0xecd8: add      rdi, qword ptr [rax]
0xecdf: nop
0xece0: add      rax, 0x50
0xece4: cmp      rax, rdx
0xece7: jne      0xecd0
0xece9: mov      r8d, 1
0xecef: lea      rsi, [rsp + 0x20]
0xecf4: mov      ecx, 1
0xecf9: mov      edx, 0x19c
0xecfe: call     0x1c520
0xed03: cmp      byte ptr [rax], 0x20
0xed06: mov      r8, rax
0xed09: jne      0xed1d
0xed0b: lea      rdx, [rax + 1]
0xed0f: nop
0xed10: mov      r8, rdx
0xed13: add      rdx, 1
0xed17: cmp      byte ptr [rdx - 1], 0x20
0xed1b: je       0xed10
0xed1d: mov      r9d, dword ptr [rbx + 0x70]
0xed21: lea      rdx, [rip + 0x136bd]
0xed28: lea      rax, [rip + 0x126ec]
0xed2f: mov      esi, 1
0xed34: lea      r13, [rsp + 0x2b0]
0xed3c: lea      rcx, [rip + 0x12410]
0xed43: cmp      r9d, 1
0xed47: mov      rdi, r13
0xed4a: cmovle   rax, rdx
0xed4e: sub      rsp, 8
0xed52: mov      edx, 0x400
0xed57: push     rax

```

```

0xed58: xor      eax, eax
0xed5a: call     0x4f10
0xed5f: mov      edi, dword ptr [rip + 0x1b607]
0xed65: call     0xa530
0xed6a: mov      edi, dword ptr [rip + 0x1b5f8]
0xed70: call     0xa4f0
0xed75: pop      rdx
0xed76: pop      rcx
0xed77: mov      rdi, r13
0xed7a: call     0x49b0
0xed7f: mov      r8, qword ptr [rbx + 0x20]
0xed83: mov      r12, rax
0xed86: cmp      eax, 8
0xed89: jae      0xee78
0xed8f: test     al, 4
0xed91: jne      0xf0b0
0xed97: test     eax, eax
0xed99: je       0xedb4
0xed9b: movzx    eax, byte ptr [rsp + 0x2b0]
0xeda3: mov      byte ptr [r8], al
0xeda6: test     r12b, 2
0xeda9: jne      0xf0ef
0xedb0: mov      r8, qword ptr [rbx + 0x20]
0xedb4: movsxd   rdx, dword ptr [rbx + 0x64]
0xedb8: lea      rcx, [r12 + 2]
0xedbd: cmp      rcx, rdx
0xedc0: jb       0xef00
0xedc6: mov      rdi, r8
0xedc9: mov      rsi, r12
0xedcc: call     0x184d0
0xedd1: mov      edi, dword ptr [rip + 0x1b549]
0xedd7: call     0xa510
0xeddc: mov      eax, dword ptr [rbx + 0x60]
0xeddf: mov      rdi, qword ptr [rbx + 8]
0xede3: mov      edx, 2
0xede8: lea      esi, [rax - 1]
0xedeb: call     0xc680
0xedf0: mov      eax, dword ptr [rbx + 0x64]
0xedf3: mov      rdi, qword ptr [rbx + 8]
0xedf7: mov      rsi, qword ptr [rbx + 0x20]
0xedfb: lea      edx, [rax - 4]
0xedfe: call     0xc600
0xee03: mov      rdi, rbp
0xee06: call     0xa6b0
0xee0b: mov      rax, qword ptr [rsp + 0x6b8]
0xee13: sub      rax, qword ptr fs:[0x28]
0xee1c: jne      0xf107
0xee22: add      rsp, 0x6c8

```

```

0xee29: pop    rbx
0xee2a: pop    rbp
0xee2b: pop    r12
0xee2d: pop    r13
0xee2f: pop    r14
0xee31: pop    r15
0xee33: ret
0xee34: nop    dword ptr [rax]
0xee38: mov    eax, dword ptr [rip + 0x1c7e2]
0xee3e: test   eax, eax
0xee40: jne    0xefc0
0xee46: mov    byte ptr [rsp + 0x2b0], 0
0xee4e: lea    r13, [rsp + 0x2b0]
0xee56: mov    edi, dword ptr [rip + 0x1b508]
0xee5c: call   0xa530
0xee61: mov    edi, dword ptr [rip + 0x1b4f9]
0xee67: call   0xa4f0
0xee6c: jmp    0xed77
0xee71: nop    dword ptr [rax]
0xee78: mov    rax, qword ptr [rsp + 0x2b0]
0xee80: lea    rdi, [r8 + 8]
0xee84: mov    rsi, r13
0xee87: and    rdi, 0xfffffffffffffff8
0xee8b: mov    qword ptr [r8], rax
0xee8e: mov    eax, r12d
0xee91: mov    rdx, qword ptr [r13 + rax - 8]
0xee96: mov    qword ptr [r8 + rax - 8], rdx
0xee9b: sub    r8, rdi
0xee9e: sub    rsi, r8
0xeea1: add    r8d, r12d
0xeea4: shr    r8d, 3
0xeea8: mov    ecx, r8d
0xeeab: rep movsq    qword ptr [rdi], qword ptr [rsi]
0xeeae: jmp    0xedb0
0xeeb3: nop    dword ptr [rax + rax]
0xeeb8: cmp    eax, 0xd
0xeebb: jle    0xee46
0xeebd: mov    esi, dword ptr [rbx + 0x28]
0xeec0: lea    r12, [rsp + 0x20]
0xeec5: mov    rdi, rbx
0xeec8: xor    eax, eax
0xeeca: mov    rdx, r12
0xeecd: lea    r13, [rsp + 0x2b0]
0xeed5: call   0xe560
0xeeda: mov    r8, r12
0xeedd: mov    edx, 0x400
0xeeee2: mov    rdi, r13
0xeeee5: lea    rcx, [rip + 0x12289]

```

```

0xeeec: mov     esi, 1
0xeef1: xor     eax, eax
0xeef3: call    0x4f10
0xeef8: jmp     0xee56
0xeefd: nop     dword ptr [rax]
0xef00: lea     edx, [rdx - 2]
0xef03: lea     rdi, [r8 + r12]
0xef07: mov     esi, 0x20
0xef0c: movsxd  rdx, edx
0xef0f: sub     rdx, r12
0xef12: call    0x4a50
0xef17: mov     r8, qword ptr [rbx + 0x20]
0xef1b: jmp     0xedc6
0xef20: mov     esi, dword ptr [rip + 0x1b3fa]
0xef26: xor     edi, edi
0xef28: mov     edx, esi
0xef2a: call    0xa570
0xef2f: mov     eax, dword ptr [rbx + 0x64]
0xef32: mov     rdi, qword ptr [rbx + 8]
0xef36: lea     edx, [rax - 1]
0xef39: mov     eax, dword ptr [rbx + 0x8c]
0xef3f: lea     esi, [rax + 1]
0xef42: call    0xc680
0xef47: mov     rdi, qword ptr [rbx + 8]
0xef4b: mov     esi, 0x20
0xef50: call    0xc650
0xef55: mov     eax, dword ptr [rbx + 0x60]
0xef58: mov     ecx, dword ptr [rbx + 0x68]
0xef5b: lea     edx, [rax - 1]
0xef5e: cmp     edx, ecx
0xef60: jg      0xec8e
0xef66: sub     eax, 2
0xef69: imul    eax, dword ptr [rbx + 0x28]
0xef6d: mov     esi, dword ptr [rip + 0x1b3b5]
0xef73: mov     edi, dword ptr [rip + 0x1b3b7]
0xef79: cdq
0xef7a: idiv    ecx
0xef7c: mov     edx, dword ptr [rip + 0x1b3aa]
0xef82: mov     dword ptr [rbx + 0x8c], eax
0xef88: call    0xa570
0xef8d: mov     eax, dword ptr [rbx + 0x64]
0xef90: mov     rdi, qword ptr [rbx + 8]
0xef94: lea     edx, [rax - 1]
0xef97: mov     eax, dword ptr [rbx + 0x8c]
0xef9d: lea     esi, [rax + 1]
0xefa0: call    0xc680
0xefa5: mov     rdi, qword ptr [rbx + 8]
0xefa9: mov     esi, 0x23

```

```

0xefae: call    0xc650
0xefb3: jmp     0xec8e
0xefb8: nop     dword ptr [rax + rax]
0xefc0: lea     r9, [rsp + 0x20]
0xefc5: mov     esi, dword ptr [rbx + 0x28]
0xefc8: xor     eax, eax
0xefca: mov     rdi, rbx
0xefcd: mov     rdx, r9
0xefd0: mov     qword ptr [rsp + 8], r9
0xefd5: lea     r13, [rsp + 0x2b0]
0xefdd: call    0xe560
0xefe2: mov     eax, dword ptr [rbx + 0x64]
0xefe5: lea     r12d, [rax - 0x1a]
0xefe9: movsxd  rax, dword ptr [rbx + 0x28]
0xefed: movsxd  r12, r12d
0xefff0: lea     r15, [rax + rax*4]
0xefff4: shl     r15, 4
0xefff8: add     r15, qword ptr [rbx + 0x38]
0xeffc: mov     rsi, qword ptr [r15]
0xefff: mov     rdi, rsi
0xf002: mov     qword ptr [rsp], rsi
0xf006: call    0x49b0
0xf00b: mov     rsi, qword ptr [rsp]
0xf00f: mov     ecx, 0x400
0xf014: mov     rdi, r13
0xf017: cmp     r12, rax
0xf01a: cmovbe  rax, r12
0xf01e: mov     rdx, rax
0xf021: mov     r14, rax
0xf024: call    0x4b70
0xf029: mov     rdx, r12
0xf02c: lea     rdi, [r13 + r14]
0xf031: mov     esi, 0x20
0xf036: sub     rdx, r14
0xf039: call    0x4a50
0xf03e: cmp     byte ptr [r15 + 0x48], 0
0xf043: mov     r9, qword ptr [rsp + 8]
0xf048: jne     0xf088
0xf04a: mov     rax, qword ptr [r15]
0xf04d: cmp     byte ptr [rax], 0x2e
0xf050: jne     0xf058
0xf052: cmp     byte ptr [rax + 1], 0x2e
0xf056: je      0xf0d0
0xf058: lea     r8, [rip + 0x120ec]
0xf05f: lea     rdi, [r13 + r12]
0xf064: lea     rcx, [rip + 0x120f8]
0xf06b: mov     rdx, -1
0xf072: xor     eax, eax

```

```

0xf074: mov     esi, 1
0xf079: call    0x4f10
0xf07e: jmp     0xee56
0xf083: nop     dword ptr [rax + rax]
0xf088: mov     r8, qword ptr [r15 + 8]
0xf08c: lea     rdi, [r13 + r12]
0xf091: mov     esi, 1
0xf096: xor     eax, eax
0xf098: lea     rcx, [rip + 0x120cf]
0xf09f: mov     rdx, -1
0xf0a6: call    0x4f10
0xf0ab: jmp     0xee56
0xf0b0: mov     eax, dword ptr [rsp + 0x2b0]
0xf0b7: mov     dword ptr [r8], eax
0xf0ba: mov     eax, r12d
0xf0bd: mov     edx, dword ptr [r13 + rax - 4]
0xf0c2: mov     dword ptr [r8 + rax - 4], edx
0xf0c7: mov     r8, qword ptr [rbx + 0x20]
0xf0cb: jmp     0xedb4
0xf0d0: cmp     byte ptr [rax + 2], 0
0xf0d4: lea     r8, [rip + 0x12068]
0xf0db: je      0xf05f
0xf0dd: jmp     0xf058
0xf0e2: nop     word ptr [rax + rax]
0xf0e8: xor     edi, edi
0xf0ea: jmp     0xece9
0xf0ef: mov     eax, r12d
0xf0f2: movzx   edx, word ptr [r13 + rax - 2]
0xf0f8: mov     word ptr [r8 + rax - 2], dx
0xf0fe: mov     r8, qword ptr [rbx + 0x20]
0xf102: jmp     0xedb4
0xf107: call    0x49d0
0xf10c: nop     dword ptr [rax]
0xf110: mov     eax, dword ptr [rdi + 0x2c]
0xf113: mov     edx, dword ptr [rdi + 0x28]
0xf116: mov     ecx, dword ptr [rdi + 0x60]
0xf119: mov     esi, dword ptr [rdi + 0x68]
0xf11c: cmp     eax, edx
0xf11e: jg      0xf130
0xf120: lea     edi, [rcx + rax - 2]
0xf124: cmp     edx, edi
0xf126: jge     0xf130
0xf128: cmp     edi, esi
0xf12a: jle     0xf159
0xf12c: nop     dword ptr [rax]
0xf130: sub     ecx, 2
0xf133: mov     eax, ecx
0xf135: shr     eax, 0x1f

```

```

0xf138: add    eax, ecx
0xf13a: sar    eax, 1
0xf13c: sub    edx, eax
0xf13e: mov    eax, 0
0xf143: lea    edi, [rcx + rdx]
0xf146: cmovns eax, edx
0xf149: cmp    esi, edi
0xf14b: jg     0xf159
0xf14d: mov    eax, esi
0xf14f: mov    edx, 0
0xf154: sub    eax, ecx
0xf156: cmovs  eax, edx
0xf159: ret
0xf15a: nop    word ptr [rax + rax]
0xf160: endbr64
0xf164: push   r12
0xf166: push   rbp
0xf167: mov    rbp, rdi
0xf16a: mov    edi, 0x98
0xf16f: sub    rsp, 8
0xf173: call   0x1d670
0xf178: mov    ecx, 0x100
0xf17d: mov    rdi, rbp
0xf180: mov    dword ptr [rax + 0x50], 0xffffffff
0xf187: mov    r12, rax
0xf18a: mov    qword ptr [rax + 0x20], 0
0xf192: mov    qword ptr [rax + 0x28], 0
0xf19a: mov    qword ptr [rax], 0
0xf1a1: mov    qword ptr [rax + 0x48], 0
0xf1a9: mov    qword ptr [rax + 0x58], 0
0xf1b1: movabs rax, 0x1000000001
0xf1bb: mov    qword ptr [r12 + 0x60], rax
0xf1c0: movabs rax, 0x3fffffff00000000
0xf1ca: mov    dword ptr [r12 + 0x68], 0
0xf1d3: mov    word ptr [r12 + 0x6c], cx
0xf1d9: mov    qword ptr [r12 + 0x80], 0
0xf1e5: mov    qword ptr [r12 + 0x88], 1
0xf1f1: mov    qword ptr [r12 + 0x70], 0
0xf1fa: mov    qword ptr [r12 + 0x10], rax
0xf1ff: mov    qword ptr [r12 + 0x38], 0
0xf208: call   0x49c0
0xf20d: cmp    eax, -1
0xf210: je     0xf3b0
0xf216: xor    eax, eax
0xf218: call   0x18fe0
0xf21d: mov    qword ptr [r12 + 0x18], rax
0xf222: mov    rdi, rax
0xf225: test   rax, rax

```



```

0xf228: je      0xf3c8
0xf22e: call   0x18040
0xf233: mov    rdi, qword ptr [r12 + 0x18]
0xf238: call   0x49b0
0xf23d: mov    qword ptr [r12 + 0x78], rax
0xf242: xor    eax, eax
0xf244: call   0xc590
0xf249: mov    edx, dword ptr [rip + 0x1c3b1]
0xf24f: mov    qword ptr [r12 + 8], rax
0xf254: test   edx, edx
0xf256: je      0xf280
0xf258: mov    eax, dword ptr [rip + 0x1c3d6]
0xf25e: mov    dword ptr [r12 + 0x80], eax
0xf266: mov    eax, dword ptr [rip + 0x1c3c4]
0xf26c: mov    dword ptr [r12 + 0x84], eax
0xf274: add    rsp, 8
0xf278: mov    rax, r12
0xf27b: pop    rbp
0xf27c: pop    r12
0xf27e: ret
0xf27f: nop
0xf280: lea    rdi, [rip + 0x11406]
0xf287: call   0x17c50
0xf28c: xor    esi, esi
0xf28e: lea    rdi, [rip + 0x11ee5]
0xf295: call   0x17d10
0xf29a: lea    rdi, [rip + 0x11411]
0xf2a1: mov    dword ptr [rip + 0x1c385], eax
0xf2a7: call   0x17c50
0xf2ac: xor    ecx, ecx
0xf2ae: mov    edx, 6
0xf2b3: lea    rsi, [rip + 0x1b326]
0xf2ba: lea    rdi, [rip + 0x11ecb]
0xf2c1: call   0x17d70
0xf2c6: xor    ecx, ecx
0xf2c8: mov    edx, 9
0xf2cd: lea    rsi, [rip + 0x1b2ac]
0xf2d4: mov    dword ptr [r12 + 0x80], eax
0xf2dc: lea    rdi, [rip + 0x11ec0]
0xf2e3: mov    dword ptr [rip + 0x1c34b], eax
0xf2e9: call   0x17d70
0xf2ee: mov    esi, 1
0xf2f3: lea    rdi, [rip + 0x11ebf]
0xf2fa: mov    dword ptr [r12 + 0x84], eax
0xf302: mov    dword ptr [rip + 0x1c328], eax
0xf308: call   0x17e50
0xf30d: mov    esi, 1
0xf312: lea    rdi, [rip + 0x11eac]

```

```

0xf319: mov     dword ptr [rip + 0x1c301], eax
0xf31f: call    0x17e50
0xf324: lea     rdi, [rip + 0x1a91d]
0xf32b: mov     dword ptr [rip + 0x1c2f3], eax
0xf331: mov     eax, dword ptr [rip + 0x1a929]
0xf337: test    eax, eax
0xf339: lea     rax, [rip + 0x1a8f0]
0xf340: cmov     rdi, rax
0xf344: call    0x17c50
0xf349: mov     edx, 0x14
0xf34e: lea     rsi, [rip + 0x1b18b]
0xf355: lea     rdi, [rip + 0x1afc4]
0xf35c: call    0x17f70
0xf361: mov     edi, 8
0xf366: call    0x19100
0xf36b: mov     edi, 8
0xf370: mov     qword ptr [rip + 0x1c2a1], rax
0xf377: call    0x19100
0xf37c: mov     edi, 8
0xf381: mov     qword ptr [rip + 0x1c288], rax
0xf388: call    0x19100
0xf38d: mov     dword ptr [rip + 0x1c269], 1
0xf397: mov     qword ptr [rip + 0x1c26a], rax
0xf39e: add     rsp, 8
0xf3a2: mov     rax, r12
0xf3a5: pop     rbp
0xf3a6: pop     r12
0xf3a8: ret
0xf3a9: nop     dword ptr [rax]
0xf3b0: lea     rdi, [rip + 0x12641]
0xf3b7: call    0x8b00
0xf3bc: jmp     0xf216
0xf3c1: nop     dword ptr [rax]
0xf3c8: lea     rdi, [rip + 0x11581]
0xf3cf: call    0x8b00
0xf3d4: mov     rdi, qword ptr [r12 + 0x18]
0xf3d9: jmp     0xf22e
0xf3de: nop
0xf3e0: endbr64
0xf3e4: push    r12
0xf3e6: mov     r12, rdi
0xf3e9: push    rbp
0xf3ea: push    rbx
0xf3eb: mov     rdi, qword ptr [rdi]
0xf3ee: test    rdi, rdi
0xf3f1: je      0xf3f8
0xf3f3: call    0x4ac0
0xf3f8: mov     eax, dword ptr [r12 + 0x68]

```

```

0xf3fd: mov     r8, qword ptr [r12 + 0x38]
0xf402: test    eax, eax
0xf404: jle     0xf43e
0xf406: xor     ebx, ebx
0xf408: nop     dword ptr [rax + rax]
0xf410: lea     rbp, [rbx + rbx*4]
0xf414: shl     rbp, 4
0xf418: mov     rdi, qword ptr [r8 + rbp]
0xf41c: test    rdi, rdi
0xf41f: je      0xf433
0xf421: call    0x18d10
0xf426: mov     r8, qword ptr [r12 + 0x38]
0xf42b: mov     qword ptr [r8 + rbp], 0
0xf433: add     rbx, 1
0xf437: cmp     dword ptr [r12 + 0x68], ebx
0xf43c: jg      0xf410
0xf43e: mov     rdi, r8
0xf441: call    0x18d10
0xf446: mov     rdi, qword ptr [r12 + 0x20]
0xf44b: call    0x18d10
0xf450: mov     rdi, qword ptr [r12 + 8]
0xf455: call    0xc5c0
0xf45a: pop     rbx
0xf45b: mov     rdi, r12
0xf45e: pop     rbp
0xf45f: pop     r12
0xf461: jmp     0x18d10
0xf466: nop     word ptr cs:[rax + rax]
0xf470: endbr64
0xf474: mov     rax, qword ptr [rdi + 8]
0xf478: ret
0xf479: nop     dword ptr [rax]
0xf480: endbr64
0xf484: mov     dword ptr [rdi + 0x14], 0x3fffffff
0xf48b: ret
0xf48c: nop     dword ptr [rax]
0xf490: endbr64
0xf494: movsxd  rax, dword ptr [rdi + 0x28]
0xf498: lea     rax, [rax + rax*4]
0xf49c: shl     rax, 4
0xf4a0: add     rax, qword ptr [rdi + 0x38]
0xf4a4: mov     rax, qword ptr [rax]
0xf4a7: ret
0xf4a8: nop     dword ptr [rax + rax]
0xf4b0: endbr64
0xf4b4: movsxd  rax, dword ptr [rdi + 0x28]
0xf4b8: lea     rax, [rax + rax*4]
0xf4bc: shl     rax, 4

```

```

0xf4c0: add    rax, qword ptr [rdi + 0x38]
0xf4c4: mov     eax, dword ptr [rax + 0x14]
0xf4c7: ret
0xf4c8: nop     dword ptr [rax + rax]
0xf4d0: endbr64
0xf4d4: movsxd  rax, dword ptr [rdi + 0x28]
0xf4d8: lea     rax, [rax + rax*4]
0xf4dc: shl     rax, 4
0xf4e0: add     rax, qword ptr [rdi + 0x38]
0xf4e4: mov     eax, dword ptr [rax + 0x18]
0xf4e7: ret
0xf4e8: nop     dword ptr [rax + rax]
0xf4f0: endbr64
0xf4f4: movsxd  rax, dword ptr [rdi + 0x28]
0xf4f8: lea     rax, [rax + rax*4]
0xf4fc: shl     rax, 4
0xf500: add     rax, qword ptr [rdi + 0x38]
0xf504: mov     eax, dword ptr [rax + 0x10]
0xf507: ret
0xf508: nop     dword ptr [rax + rax]
0xf510: endbr64
0xf514: movsxd  rax, dword ptr [rdi + 0x28]
0xf518: lea     rax, [rax + rax*4]
0xf51c: shl     rax, 4
0xf520: add     rax, qword ptr [rdi + 0x38]
0xf524: movsx   eax, byte ptr [rax + 0x48]
0xf528: ret
0xf529: nop     dword ptr [rax]
0xf530: endbr64
0xf534: mov     eax, dword ptr [rdi + 0x60]
0xf537: mov     esi, dword ptr [rdi + 0x68]
0xf53a: mov     r8d, 0
0xf540: lea     edx, [rax - 2]
0xf543: mov     eax, dword ptr [rdi + 0x28]
0xf546: mov     ecx, edx
0xf548: shr     ecx, 0x1f
0xf54b: add     ecx, edx
0xf54d: sar     ecx, 1
0xf54f: sub     eax, ecx
0xf551: lea     ecx, [rdx + rax]
0xf554: cmovs   eax, r8d
0xf558: cmp     esi, ecx
0xf55a: jg      0xf568
0xf55c: mov     eax, esi
0xf55e: sub     eax, edx
0xf560: mov     edx, 0
0xf565: cmovs   eax, edx
0xf568: mov     dword ptr [rdi + 0x2c], eax

```

```

0xf56b: ret
0xf56c: nop      dword ptr [rax]
0xf570: endbr64
0xf574: mov      byte ptr [rdi + 0x6d], 1
0xf578: ret
0xf579: nop      dword ptr [rax]
0xf580: endbr64
0xf584: mov      byte ptr [rdi + 0x6d], 0
0xf588: ret
0xf589: nop      dword ptr [rax]
0xf590: endbr64
0xf594: mov      eax, dword ptr [rip + 0x1c092]
0xf59a: push     r14
0xf59c: mov      r14d, ecx
0xf59f: push     r13
0xf5a1: mov      r13d, edx
0xf5a4: push     r12
0xf5a6: mov      r12d, esi
0xf5a9: push     rbp
0xf5aa: mov      rbp, r8
0xf5ad: push     rbx
0xf5ae: mov      rbx, rdi
0xf5b1: mov      dword ptr [rdi + 0x58], esi
0xf5b4: mov      dword ptr [rdi + 0x5c], edx
0xf5b7: mov      dword ptr [rdi + 0x60], ecx
0xf5ba: mov      dword ptr [rdi + 0x64], r8d
0xf5be: mov      dword ptr [rdi + 0x8c], 0
0xf5c8: test     eax, eax
0xf5ca: jle      0xf648
0xf5cc: lea      edx, [rcx - 1]
0xf5cf: cmp      eax, edx
0xf5d1: jge      0xf648
0xf5d3: mov      dword ptr [rbx + 0x88], eax
0xf5d9: mov      rdi, qword ptr [rbx + 0x20]
0xf5dd: movsxd   rsi, ebp
0xf5e0: call     0x1d690
0xf5e5: mov      rdi, qword ptr [rbx + 8]
0xf5e9: mov      ecx, r14d
0xf5ec: mov      edx, r13d
0xf5ef: mov      qword ptr [rbx + 0x20], rax
0xf5f3: mov      esi, r12d
0xf5f6: mov      r8d, ebp
0xf5f9: call     0xc5e0
0xf5fe: mov      eax, dword ptr [rbx + 0x60]
0xf601: mov      esi, dword ptr [rbx + 0x68]
0xf604: mov      edi, 0
0xf609: lea      edx, [rax - 2]
0xf60c: mov      eax, dword ptr [rbx + 0x28]

```

```

0xf60f: mov     ecx, edx
0xf611: shr     ecx, 0x1f
0xf614: add     ecx, edx
0xf616: sar     ecx, 1
0xf618: sub     eax, ecx
0xf61a: lea     ecx, [rdx + rax]
0xf61d: cmovs   eax, edi
0xf620: cmp     esi, ecx
0xf622: jg      0xf630
0xf624: mov     eax, esi
0xf626: sub     eax, edx
0xf628: mov     edx, 0
0xf62d: cmovs   eax, edx
0xf630: mov     dword ptr [rbx + 0x2c], eax
0xf633: mov     dword ptr [rbx + 0x74], 0
0xf63a: pop     rbx
0xf63b: pop     rbp
0xf63c: pop     r12
0xf63e: pop     r13
0xf640: pop     r14
0xf642: ret
0xf643: nop     dword ptr [rax + rax]
0xf648: mov     eax, ecx
0xf64a: shr     eax, 0x1f
0xf64d: add     eax, ecx
0xf64f: sar     eax, 1
0xf651: jmp     0xf5d3
0xf653: nop     word ptr cs:[rax + rax]
0xf65e: nop
0xf660: endbr64
0xf664: mov     byte ptr [rdi + 0x6e], sil
0xf668: ret
0xf669: nop     dword ptr [rax]
0xf670: endbr64
0xf674: push    r13
0xf676: push    r12
0xf678: push    rbp
0xf679: push    rbx
0xf67a: sub     rsp, 8
0xf67e: test    ecx, ecx
0xf680: js      0xf6ce
0xf682: movsxd  rax, ecx
0xf685: mov     rbp, rsi
0xf688: mov     r13, rdx
0xf68b: mov     r12d, ecx
0xf68e: lea     rbx, [rax + rax*4]
0xf692: shl     rbx, 4
0xf696: add     rbx, qword ptr [rdi + 0x38]

```

```

0xf69a: jmp      0xf6ae
0xf69c: nop      dword ptr [rax]
0xf6a0: sub      r12d, 1
0xf6a4: sub      rbx, 0x50
0xf6a8: cmp      r12d, -1
0xf6ac: je       0xf6c0
0xf6ae: mov      rsi, qword ptr [rbx]
0xf6b1: mov      rdx, r13
0xf6b4: mov      rdi, rbp
0xf6b7: call     0x4c30
0xf6bc: test     eax, eax
0xf6be: jne      0xf6a0
0xf6c0: add      rsp, 8
0xf6c4: mov      eax, r12d
0xf6c7: pop      rbx
0xf6c8: pop      rbp
0xf6c9: pop      r12
0xf6cb: pop      r13
0xf6cd: ret
0xf6ce: mov      r12d, 0xffffffff
0xf6d4: jmp      0xf6c0
0xf6d6: nop      word ptr cs:[rax + rax]
0xf6e0: endbr64
0xf6e4: push     r14
0xf6e6: push     r13
0xf6e8: push     r12
0xf6ea: push     rbp
0xf6eb: push     rbx
0xf6ec: mov      r14d, dword ptr [rdi + 0x68]
0xf6f0: cmp      ecx, r14d
0xf6f3: jge      0xf740
0xf6f5: movsxd   rax, ecx
0xf6f8: mov      rbp, rsi
0xf6fb: mov      r13, rdx
0xf6fe: mov      r12d, ecx
0xf701: lea      rbx, [rax + rax*4]
0xf705: shl      rbx, 4
0xf709: add      rbx, qword ptr [rdi + 0x38]
0xf70d: jmp      0xf71d
0xf70f: nop
0xf710: add      r12d, 1
0xf714: add      rbx, 0x50
0xf718: cmp      r12d, r14d
0xf71b: je       0xf740
0xf71d: mov      rsi, qword ptr [rbx]
0xf720: mov      rdx, r13
0xf723: mov      rdi, rbp
0xf726: call     0x4c30

```

```

0xf72b: test    eax, eax
0xf72d: jne      0xf710
0xf72f: pop      rbx
0xf730: mov      eax, r12d
0xf733: pop      rbp
0xf734: pop      r12
0xf736: pop      r13
0xf738: pop      r14
0xf73a: ret
0xf73b: nop      dword ptr [rax + rax]
0xf740: mov      r12d, 0xffffffff
0xf746: pop      rbx
0xf747: pop      rbp
0xf748: mov      eax, r12d
0xf74b: pop      r12
0xf74d: pop      r13
0xf74f: pop      r14
0xf751: ret
0xf752: nop      word ptr cs:[rax + rax]
0xf75d: nop      dword ptr [rax]
0xf760: endbr64
0xf764: push     r14
0xf766: mov      r14d, ecx
0xf769: push     r13
0xf76b: mov      r13, rdx
0xf76e: push     r12
0xf770: mov      r12, rdi
0xf773: push     rbp
0xf774: push     rbx
0xf775: mov      rbx, rsi
0xf778: call     0x49b0
0xf77d: mov      rdi, rbx
0xf780: mov      rbp, rax
0xf783: call     0x49b0
0xf788: lea      rdi, [rbp + rax + 1]
0xf78d: call     0x1d670
0xf792: mov      r8, rbx
0xf795: mov      rcx, r12
0xf798: mov      esi, 1
0xf79d: mov      rbp, rax
0xf7a0: mov      rdi, rax
0xf7a3: mov      rdx, -1
0xf7aa: xor      eax, eax
0xf7ac: call     0x4f10
0xf7b1: mov      edx, r14d
0xf7b4: mov      rsi, r13
0xf7b7: mov      rdi, rbp
0xf7ba: call     0x8070

```



```

0xf7bf: mov     rdi, rbp
0xf7c2: mov     r12d, eax
0xf7c5: call    0x18d10
0xf7ca: pop     rbx
0xf7cb: mov     eax, r12d
0xf7ce: pop     rbp
0xf7cf: pop     r12
0xf7d1: pop     r13
0xf7d3: pop     r14
0xf7d5: ret
0xf7d6: nop     word ptr cs:[rax + rax]
0xf7e0: endbr64
0xf7e4: push    r15
0xf7e6: mov     r15d, r8d
0xf7e9: push    r14
0xf7eb: mov     r14, rcx
0xf7ee: push    r13
0xf7f0: mov     r13, rdx
0xf7f3: push    r12
0xf7f5: mov     r12, rdi
0xf7f8: push    rbp
0xf7f9: push    rbx
0xf7fa: mov     rbx, rsi
0xf7fd: sub     rsp, 0x18
0xf801: call    0x49b0
0xf806: mov     rdi, rbx
0xf809: mov     qword ptr [rsp + 8], rax
0xf80e: call    0x49b0
0xf813: mov     rdi, r13
0xf816: mov     rbp, rax
0xf819: call    0x49b0
0xf81e: mov     rdx, qword ptr [rsp + 8]
0xf823: lea     rdi, [rdx + rbp + 1]
0xf828: add     rdi, rax
0xf82b: call    0x1d670
0xf830: mov     r9, r13
0xf833: mov     r8, rbx
0xf836: mov     rcx, r12
0xf839: mov     rbp, rax
0xf83c: mov     rdi, rax
0xf83f: mov     esi, 1
0xf844: xor     eax, eax
0xf846: mov     rdx, -1
0xf84d: call    0x4f10
0xf852: mov     edx, r15d
0xf855: mov     rsi, r14
0xf858: mov     rdi, rbp
0xf85b: call    0x8070

```

```

0xf860: mov     rdi, rbp
0xf863: mov     r12d, eax
0xf866: call    0x18d10
0xf86b: add     rsp, 0x18
0xf86f: mov     eax, r12d
0xf872: pop     rbx
0xf873: pop     rbp
0xf874: pop     r12
0xf876: pop     r13
0xf878: pop     r14
0xf87a: pop     r15
0xf87c: ret
0xf87d: nop     dword ptr [rax]
0xf880: endbr64
0xf884: push    r15
0xf886: mov     r15, rdi
0xf889: mov     rdi, rsi
0xf88c: push    r14
0xf88e: push    r13
0xf890: push    r12
0xf892: push    rbp
0xf893: mov     rbp, rsi
0xf896: push    rbx
0xf897: mov     ebx, edx
0xf899: sub     rsp, 0x198
0xf8a0: mov     rax, qword ptr fs:[0x28]
0xf8a9: mov     qword ptr [rsp + 0x188], rax
0xf8b1: xor     eax, eax
0xf8b3: call    0x4950
0xf8b8: mov     r12, rax
0xf8bb: test    rax, rax
0xf8be: je      0xfda8
0xf8c4: mov     rdi, rbp
0xf8c7: call    0x49c0
0xf8cc: cmp     eax, -1
0xf8cf: je      0x10050
0xf8d5: mov     rdi, qword ptr [r15]
0xf8d8: test    rdi, rdi
0xf8db: je      0xf8e2
0xf8dd: call    0x4ac0
0xf8e2: mov     rax, qword ptr [r15 + 0x78]
0xf8e6: mov     qword ptr [r15], r12
0xf8e9: lea     rdi, [rax + 1]
0xf8ed: call    0x1d670
0xf8f2: mov     rsi, qword ptr [r15 + 0x18]
0xf8f6: mov     rdi, rax
0xf8f9: mov     qword ptr [rsp + 0x18], rax
0xf8fe: call    0x48b0

```

```

0xf903: cmp    byte ptr [rbp], 0x2f
0xf907: je      0xfb20
0xf90d: xor     eax, eax
0xf90f: call    0x18fe0
0xf914: mov     r12, rax
0xf917: test    rax, rax
0xf91a: je      0xff20
0xf920: mov     rdi, qword ptr [r15 + 0x18]
0xf924: call    0x18d10
0xf929: mov     qword ptr [r15 + 0x18], r12
0xf92d: mov     rdi, r12
0xf930: call    0x18040
0xf935: mov     rbp, qword ptr [r15 + 0x18]
0xf939: mov     rdi, rbp
0xf93c: call    0x49b0
0xf941: lea     rsi, [rsp + 0xf0]
0xf949: mov     rdi, rbp
0xf94c: mov     qword ptr [r15 + 0x78], rax
0xf950: call    0x18f40
0xf955: cmp     qword ptr [rsp + 0x120], 0x7ff
0xf961: jg      0xff08
0xf967: lea     rsi, [rsp + 0x70]
0xf96c: lea     rdi, [rip + 0x11d6b]
0xf973: call    0x4c00
0xf978: xor     eax, eax
0xf97a: cmp     qword ptr [rsp + 0x70], 0x4d44
0xf983: sete    al
0xf986: mov     dword ptr [r15 + 0x10], eax
0xf98a: test    ebx, ebx
0xf98c: je      0xf99b
0xf98e: mov     r13d, dword ptr [r15 + 0x70]
0xf992: test    r13d, r13d
0xf995: jne     0xff78
0xf99b: mov     rbp, qword ptr [r15 + 0x38]
0xf99f: test    rbp, rbp
0xf9a2: je      0x10018
0xf9a8: mov     r12d, dword ptr [r15 + 0x68]
0xf9ac: test    r12d, r12d
0xf9af: jle     0xf9e6
0xf9b1: xor     ebx, ebx
0xf9b3: nop     dword ptr [rax + rax]
0xf9b8: lea     r12, [rbx + rbx*4]
0xf9bc: shl     r12, 4
0xf9c0: mov     rdi, qword ptr [rbp + r12]
0xf9c5: test    rdi, rdi
0xf9c8: je      0xf9dc
0xf9ca: call    0x18d10
0xf9cf: mov     rbp, qword ptr [r15 + 0x38]

```

```

0xf9d3: mov     qword ptr [rbp + r12], 0
0xf9dc: add     rbx, 1
0xf9e0: cmp     dword ptr [r15 + 0x68], ebx
0xf9e4: jg      0xf9b8
0xf9e6: mov     rdi, rbp
0xf9e9: xor     ebp, ebp
0xf9eb: call    0x18d10
0xf9f0: mov     qword ptr [r15 + 0x38], 0
0xf9f8: mov     dword ptr [rsp + 0xc], 0
0xfa00: mov     dword ptr [rsp], 0
0xfa07: mov     edi, 0x50
0xfa0c: call    0x1d670
0xfa11: mov     dword ptr [r15 + 0x70], 0
0xfa19: mov     dword ptr [r15 + 0x30], 0
0xfa21: mov     dword ptr [rsp + 0x14], 0xffffffff
0xfa29: mov     dword ptr [rsp + 0x10], 0
0xfa31: mov     qword ptr [r15 + 0x38], rax
0xfa35: xor     eax, eax
0xfa37: nop     word ptr [rax + rax]
0xfa40: mov     dword ptr [r15 + 0x68], eax
0xfa44: mov     rdi, qword ptr [r15]
0xfa47: call    0x4c10
0xfa4c: test    rax, rax
0xfa4f: je      0xfbf0
0xfa55: movabs  rdx, 0xffff000000
0xfa5f: and     rdx, qword ptr [rax + 0x10]
0xfa63: cmp     rdx, 0x2e000000
0xfa6a: je      0xfbf0
0xfa70: cmp     byte ptr [rax + 0x13], 0x2e
0xfa74: lea     r13, [rax + 0x13]
0xfa78: je      0xfbf38
0xfa7e: mov     eax, dword ptr [r15 + 0x68]
0xfa82: add     eax, 1
0xfa85: mov     rdi, qword ptr [r15 + 0x38]
0xfa89: cdqe
0xfa8b: lea     rsi, [rax + rax*4]
0xfa8f: shl     rsi, 4
0xfa93: call    0x1d690
0xfa98: mov     ebx, dword ptr [rsp + 0xc]
0xfa9c: mov     qword ptr [r15 + 0x38], rax
0xfaa0: mov     r14, rax
0xfaa3: test    ebx, ebx
0xfaa5: jne     0xfbf90
0xfaab: movsxd  rax, dword ptr [r15 + 0x68]
0xfaaf: lea     rdx, [rax + rax*4]
0xfab3: shl     rdx, 4
0xfab7: add     r14, rdx
0xfaba: mov     byte ptr [r14 + 0x49], 0

```

```

0xfabf: mov     byte ptr [r14 + 0x4b], 0
0xfac4: mov     rdi, r13
0xfac7: call    0x49b0
0xfacc: lea     edi, [rax + 1]
0xfacf: mov     rbx, rax
0xfad2: movsxd  rdi, edi
0xfad5: call    0x1d670
0xfada: mov     rsi, r13
0xfadd: mov     qword ptr [r14], rax
0xfae0: movsxd  rax, dword ptr [r15 + 0x68]
0xfae4: lea     rax, [rax + rax*4]
0xfae8: shl     rax, 4
0xfaec: add     rax, qword ptr [r15 + 0x38]
0xfaf0: mov     rdi, qword ptr [rax]
0xfaf3: call    0x48b0
0xfaf8: mov     eax, dword ptr [r15 + 0x30]
0xfafc: mov     esi, dword ptr [r15 + 0x68]
0xfb00: mov     rdi, r15
0xfb03: cmp     eax, ebx
0xfb05: cmovl   eax, ebx
0xfb08: mov     dword ptr [r15 + 0x30], eax
0xfb0c: xor     eax, eax
0xfb0e: call    0xe840
0xfb13: mov     eax, dword ptr [r15 + 0x68]
0xfb17: add     eax, 1
0xfb1a: jmp     0xfa40
0xfb1f: nop
0xfb20: mov     rdi, rbp
0xfb23: call    0x1d7a0
0xfb28: mov     qword ptr [r15 + 0x18], rax
0xfb2c: mov     r12, rax
0xfb2f: jmp     0xf92d
0xfb34: nop     dword ptr [rax]
0xfb38: cmp     byte ptr [r13 + 1], 0x2e
0xfb3d: jne     0xfa7e
0xfb43: cmp     byte ptr [r13 + 2], 0
0xfb48: jne     0xfa7e
0xfb4e: mov     rdx, qword ptr [r15 + 0x18]
0xfb52: mov     eax, dword ptr [r15 + 0x68]
0xfb56: cmp     byte ptr [rdx + 1], 0
0xfb5a: je      0xfb80
0xfb5c: mov     dword ptr [rsp + 0x14], eax
0xfb60: mov     dword ptr [rsp + 0x10], 1
0xfb68: jmp     0xfa82
0xfb6d: nop     dword ptr [rax]
0xfb70: mov     eax, dword ptr [r15 + 0x68]
0xfb74: jmp     0xfa40
0xfb79: nop     dword ptr [rax]

```

```

0xfb80: mov     dword ptr [rsp + 0x10], 1
0xfb88: jmp     0xfa40
0xfb8d: nop     dword ptr [rax]
0xfb90: mov     r11d, dword ptr [rsp]
0xfb94: test    r11d, r11d
0xfb97: jle     0x1002c
0xfb9d: mov     rbx, rbp
0xfba0: xor     r12d, r12d
0xfba3: jmp     0xfbba
0xfba5: nop     dword ptr [rax]
0xfba8: add     r12d, 1
0xfbac: add     rbx, 0x50
0xfbb0: cmp     dword ptr [rsp], r12d
0xfbb4: je      0xfaab
0xfbb8: mov     rsi, qword ptr [rbx]
0xfbbd: mov     rdi, r13
0xfbc0: call    0x4b30
0xfbc5: test    eax, eax
0xfbc7: jne     0xfba8
0xfbc9: movsxd  rax, dword ptr [r15 + 0x68]
0xfbcd: lea     rdx, [rax + rax*4]
0xfbd1: movsx   eax, byte ptr [rbx + 0x49]
0xfbd5: shl     rdx, 4
0xfbd9: add     r14, rdx
0xfbd9: mov     byte ptr [r14 + 0x49], al
0xfbe0: add     dword ptr [r15 + 0x70], eax
0xfbe4: jmp     0xfabf
0xfbe9: nop     dword ptr [rax]
0xfbf0: mov     r9d, dword ptr [rsp + 0x10]
0xfbf5: test    r9d, r9d
0xfbf8: je      0xfe48
0xfbfe: movsxd  rax, dword ptr [rsp + 0x14]
0xfc03: cmp     eax, -1
0xfc06: je      0xfe00
0xfc0c: mov     rdi, qword ptr [r15 + 0x38]
0xfc10: lea     rax, [rax + rax*4]
0xfc14: shl     rax, 4
0xfc18: add     rax, rdi
0xfc1b: mov     edx, dword ptr [rax + 0x10]
0xfc1e: and     edx, 0xf000
0xfc24: cmp     edx, 0x4000
0xfc2a: je      0xfc33
0xfc2c: mov     dword ptr [rax + 0x10], 0x4000
0xfc33: mov     edx, dword ptr [rsp + 0xc]
0xfc37: test    edx, edx
0xfc39: je      0xffb0
0xfc3f: mov     eax, dword ptr [rsp]
0xfc42: test    eax, eax

```

```

0xfc44: jle      0xfc7d
0xfc46: sub      eax, 1
0xfc49: mov      rbx, rbp
0xfc4c: lea      rax, [rax + rax*4]
0xfc50: shl      rax, 4
0xfc54: lea      r12, [rbp + rax + 0x50]
0xfc59: nop      dword ptr [rax]
0xfc60: mov      rdi, qword ptr [rbx]
0xfc63: test     rdi, rdi
0xfc66: je       0xfc74
0xfc68: call     0x18d10
0xfc6d: mov      qword ptr [rbx], 0
0xfc74: add      rbx, 0x50
0xfc78: cmp      r12, rbx
0xfc7b: jne      0xfc60
0xfc7d: mov      rdi, rbp
0xfc80: call     0x18d10
0xfc85: mov      esi, dword ptr [r15 + 0x68]
0xfc89: mov      rdi, qword ptr [r15 + 0x38]
0xfc8d: mov      eax, dword ptr [r15 + 0x84]
0xfc94: mov      dword ptr [r15 + 0x74], 0
0xfc9c: mov      dword ptr [rip + 0x1b986], eax
0xfca2: movsxd   rax, dword ptr [rsp + 0x14]
0xfca7: cmp      eax, -1
0xfcaa: je       0xfe29
0xfcb0: lea      rax, [rax + rax*4]
0xfcb4: shl      rax, 4
0xfcb8: add      rax, rdi
0xfcbb: movdqu   xmm5, xmmword ptr [rax]
0xfcbf: movdqu   xmm4, xmmword ptr [rdi]
0xfcc3: sub      esi, 1
0xfcc6: add      rdi, 0x50
0xfcca: movdqu   xmm3, xmmword ptr [rdi - 0x40]
0xfccf: movdqu   xmm2, xmmword ptr [rdi - 0x30]
0xfcd4: movsxd   rsi, esi
0xfcd7: mov      edx, 0x50
0xfcdc: movups   xmmword ptr [rdi - 0x50], xmm5
0xfce0: movdqu   xmm6, xmmword ptr [rax + 0x10]
0xfce5: movdqu   xmm1, xmmword ptr [rdi - 0x20]
0xfcea: lea      rcx, [rip - 0x1671]
0xfcf1: movdqu   xmm0, xmmword ptr [rdi - 0x10]
0xfcf6: movaps   xmmword ptr [rsp + 0x20], xmm4
0xfcfb: movups   xmmword ptr [rdi - 0x40], xmm6
0xfcff: movdqu   xmm7, xmmword ptr [rax + 0x20]
0xfd04: movaps   xmmword ptr [rsp + 0x30], xmm3
0xfd09: movups   xmmword ptr [rdi - 0x30], xmm7
0xfd0d: movdqu   xmm5, xmmword ptr [rax + 0x30]
0xfd12: movaps   xmmword ptr [rsp + 0x40], xmm2

```

```

0xfd17: movups  xmmword ptr [rdi - 0x20], xmm5
0xfd1b: movdqu  xmm6, xmmword ptr [rax + 0x40]
0xfd20: movaps  xmmword ptr [rsp + 0x50], xmm1
0xfd25: movups  xmmword ptr [rdi - 0x10], xmm6
0xfd29: movups  xmmword ptr [rax], xmm4
0xfd2c: movups  xmmword ptr [rax + 0x10], xmm3
0xfd30: movups  xmmword ptr [rax + 0x20], xmm2
0xfd34: movups  xmmword ptr [rax + 0x30], xmm1
0xfd38: movups  xmmword ptr [rax + 0x40], xmm0
0xfd3c: movaps  xmmword ptr [rsp + 0x60], xmm0
0xfd41: call    0x48e0
0xfd46: mov     edx, dword ptr [r15 + 0x68]
0xfd4a: mov     eax, dword ptr [r15 + 0x28]
0xfd4e: mov     rdi, r15
0xfd51: mov     r13d, 1
0xfd57: cmp     edx, eax
0xfd59: lea     ecx, [rdx - 1]
0xfd5c: cmovle  eax, ecx
0xfd5f: mov     dword ptr [r15 + 0x28], eax
0xfd63: xor     eax, eax
0xfd65: call    0xf110
0xfd6a: mov     rdi, qword ptr [rsp + 0x18]
0xfd6f: mov     dword ptr [r15 + 0x2c], eax
0xfd73: call    0x18d10
0xfd78: mov     rax, qword ptr [rsp + 0x188]
0xfd80: sub     rax, qword ptr fs:[0x28]
0xfd89: jne     0x100b2
0xfd8f: add     rsp, 0x198
0xfd96: mov     eax, r13d
0xfd99: pop     rbx
0xfd9a: pop     rbp
0xfd9b: pop     r12
0xfd9d: pop     r13
0xfd9f: pop     r14
0xfda1: pop     r15
0xfda3: ret
0xfda4: nop     dword ptr [rax]
0xfda8: cmp     byte ptr [rbp], 0x2e
0xfdac: jne     0xfdf0
0xfdae: cmp     byte ptr [rbp + 1], 0x2e
0xfdb2: jne     0xfdf0
0xfdb4: movzx   r13d, byte ptr [rbp + 2]
0xfdb9: test    r13d, r13d
0xfdbc: jne     0xfdf0
0xfdbe: xor     eax, eax
0xfdc0: call    0x18fe0
0xfdc5: mov     rbp, rax
0xfdc8: test    rax, rax

```



```

0xfdc b: je      0x10060
0xfdd1: mov     rdi, rbp
0xfdd4: call    0x4950
0xfdd9: mov     r12, rax
0xfddc: test    rax, rax
0xfddf: jne     0xf8c4
0xfde5: mov     rdi, rbp
0xfde8: call    0x18d10
0xfded: jmp     0xfd78
0xfdef: nop
0xfdf0: xor     r13d, r13d
0xfdf3: jmp     0xfd78
0xfdf8: nop     dword ptr [rax + rax]
0xfe00: mov     eax, dword ptr [rsp + 0xc]
0xfe04: test    eax, eax
0xfe06: jne     0xfc3f
0xfe0c: mov     esi, dword ptr [r15 + 0x68]
0xfe10: mov     rdi, qword ptr [r15 + 0x38]
0xfe14: mov     dword ptr [r15 + 0x74], 0
0xfe1c: mov     eax, dword ptr [r15 + 0x84]
0xfe23: mov     dword ptr [rip + 0x1b7ff], eax
0xfe29: movsxd  rsi, esi
0xfe2c: lea     rcx, [rip - 0x17b3]
0xfe33: mov     edx, 0x50
0xfe38: call    0x48e0
0xfe3d: jmp     0xfd46
0xfe42: nop     word ptr [rax + rax]
0xfe48: mov     eax, dword ptr [r15 + 0x68]
0xfe4c: mov     rdi, qword ptr [r15 + 0x38]
0xfe50: add     eax, 1
0xfe53: cdqe
0xfe55: lea     rsi, [rax + rax*4]
0xfe59: shl     rsi, 4
0xfe5d: call    0x1d690
0xfe62: movsxd  rdx, dword ptr [r15 + 0x68]
0xfe66: mov     edi, 3
0xfe6b: mov     qword ptr [r15 + 0x38], rax
0xfe6f: lea     rdx, [rdx + rdx*4]
0xfe73: shl     rdx, 4
0xfe77: lea     rbx, [rax + rdx]
0xfe7b: mov     byte ptr [rbx + 0x49], 0
0xfe7f: mov     byte ptr [rbx + 0x4b], 1
0xfe83: call    0x1d670
0xfe88: mov     ecx, 0x2e2e
0xfe8d: mov     edx, 2
0xfe92: mov     rdi, r15
0xfe95: mov     qword ptr [rbx], rax
0xfe98: movsxd  rax, dword ptr [r15 + 0x68]

```

```

0xfe9c: lea    rax, [rax + rax*4]
0xfea0: shl    rax, 4
0xfea4: add    rax, qword ptr [r15 + 0x38]
0xfea8: mov    rax, qword ptr [rax]
0xfeab: mov    word ptr [rax], cx
0xfeae: mov    byte ptr [rax + 2], 0
0xfeb2: mov    eax, dword ptr [r15 + 0x30]
0xfeb6: mov    esi, dword ptr [r15 + 0x68]
0xfeba: cmp    eax, edx
0xfebc: cmovl  eax, edx
0xfebf: mov    dword ptr [r15 + 0x30], eax
0xfec3: xor    eax, eax
0xfec5: call   0xe840
0xfeca: movsxd rax, dword ptr [r15 + 0x68]
0xfece: mov    rdi, qword ptr [r15 + 0x38]
0xfed2: mov    r8d, dword ptr [rsp + 0xc]
0xfed7: mov    rsi, rax
0xfeda: lea    rax, [rax + rax*4]
0xfede: shl    rax, 4
0xfee2: add    esi, 1
0xfee5: add    rax, rdi
0xfee8: mov    dword ptr [rax + 0x10], 0x4000
0xfeef: mov    byte ptr [rax + 0x48], 0
0xfef3: mov    dword ptr [r15 + 0x68], esi
0xfef7: test   r8d, r8d
0xfefa: je     0xfc8d
0xff00: jmp    0xfc3f
0xff05: nop    dword ptr [rax]
0xff08: mov    rdi, qword ptr [rip + 0x1a571]
0xff0f: call   0xdb60
0xff14: xor    eax, eax
0xff16: call   0xa820
0xff1b: jmp    0xf967
0xff20: mov    r13, qword ptr [r15 + 0x18]
0xff24: mov    rdi, r13
0xff27: call   0x49b0
0xff2c: mov    qword ptr [r15 + 0x78], rax
0xff30: cmp    byte ptr [rbp], 0x2e
0xff34: mov    r12, rax
0xff37: jne    0xffd0
0xff3d: cmp    byte ptr [rbp + 1], 0x2e
0xff41: jne    0xffd0
0xff47: cmp    byte ptr [rbp + 2], 0
0xff4b: jne    0xffd0
0xff51: mov    esi, 0x2f
0xff56: mov    rdi, r13
0xff59: call   0x4a20
0xff5e: test   rax, rax

```

```

0xff61: je      0x1008b
0xff67: mov     byte ptr [rax], 0
0xff6a: mov     r12, qword ptr [r15 + 0x18]
0xff6e: jmp     0xf92d
0xff73: nop     dword ptr [rax + rax]
0xff78: mov     rsi, qword ptr [r15 + 0x18]
0xff7c: mov     rdi, qword ptr [rsp + 0x18]
0xff81: call    0x4b30
0xff86: test    eax, eax
0xff88: jne     0xf99b
0xff8e: mov     eax, dword ptr [r15 + 0x68]
0xff92: mov     rbp, qword ptr [r15 + 0x38]
0xff96: mov     dword ptr [rsp + 0xc], 1
0xff9e: mov     qword ptr [r15 + 0x38], 0
0xffa6: mov     dword ptr [rsp], eax
0xffa9: jmp     0xfa07
0xffae: nop
0xffb0: mov     edx, dword ptr [r15 + 0x84]
0xffb7: mov     esi, dword ptr [r15 + 0x68]
0xffbb: mov     dword ptr [r15 + 0x74], 0
0xffc3: mov     dword ptr [rip + 0x1b65f], edx
0xffc9: jmp     0xfcbb
0xffce: nop
0xffd0: mov     rdi, rbp
0xffd3: mov     r14d, 0x2f
0xffd9: call    0x49b0
0xffde: mov     rdi, r13
0xffe1: lea     rsi, [r12 + rax + 2]
0xffe6: call    0x1d690
0xffeb: mov     qword ptr [r15 + 0x18], rax
0xffef: mov     rdi, rax
0xfff2: mov     r12, rax
0xffff5: call    0x49b0
0xffffa: mov     rsi, rbp
0xffffd: mov     word ptr [r12 + rax], r14w
0x10002: mov     rdi, qword ptr [r15 + 0x18]
0x10006: call    0x4df0
0x1000b: mov     r12, qword ptr [r15 + 0x18]
0x1000f: jmp     0xf92d
0x10014: nop     dword ptr [rax]
0x10018: mov     dword ptr [rsp + 0xc], 0
0x10020: mov     dword ptr [rsp], 0
0x10027: jmp     0xfa07
0x1002c: movsxd  rax, dword ptr [r15 + 0x68]
0x10030: mov     r10d, dword ptr [rsp]
0x10034: lea     rdx, [rax + rax*4]
0x10038: shl     rdx, 4
0x1003c: add     r14, rdx

```

```

0x1003f:      test    r10d, r10d
0x10042:      jne     0xfabf
0x10048:      jmp     0xfaba
0x1004d:      nop     dword ptr [rax]
0x10050:      mov     rdi, r12
0x10053:      xor     r13d, r13d
0x10056:      call    0x4ac0
0x1005b:      jmp     0xfd78
0x10060:      mov     rdi, qword ptr [r15 + 0x18]
0x10064:      call    0x1d7a0
0x10069:      mov     esi, 0x2f
0x1006e:      mov     rdi, rax
0x10071:      mov     rbp, rax
0x10074:      call    0x4a20
0x10079:      mov     r12, rax
0x1007c:      test    rax, rax
0x1007f:      je      0x100a4
0x10081:      mov     byte ptr [r12], 0
0x10086:      jmp     0xfdd1
0x1008b:      lea     rdi, [rip + 0x11151]
0x10092:      mov     qword ptr [rsp], rax
0x10096:      call    0x8b00
0x1009b:      mov     rax, qword ptr [rsp]
0x1009f:      jmp     0xff67
0x100a4:      lea     rdi, [rip + 0x1112a]
0x100ab:      call    0x8b00
0x100b0:      jmp     0x10081
0x100b2:      call    0x49d0
0x100b7:      nop     word ptr [rax + rax]
0x100c0:      endbr64
0x100c4:      mov     eax, dword ptr [rdi + 0x70]
0x100c7:      mov     dword ptr [rdi + 0x50], 0xffffffff
0x100ce:      mov     dword ptr [rdi + 0x54], eax
0x100d1:      ret
0x100d2:      nop     word ptr cs:[rax + rax]
0x100dd:      nop     dword ptr [rax]
0x100e0:      endbr64
0x100e4:      mov     eax, dword ptr [rdi + 0x54]
0x100e7:      movsxd  rdx, dword ptr [rdi + 0x50]
0x100eb:      test    eax, eax
0x100ed:      je      0x10128
0x100ef:      mov     ecx, dword ptr [rdi + 0x68]
0x100f2:      lea     eax, [rdx + 1]
0x100f5:      cmp     ecx, eax
0x100f7:      jle     0x10154
0x100f9:      mov     rsi, qword ptr [rdi + 0x38]
0x100fd:      lea     rdx, [rdx + rdx*4]
0x10101:      shl     rdx, 4

```

```

0x10105:    lea     rdx, [rsi + rdx + 0x99]
0x1010d:    jmp     0x1011b
0x1010f:    nop
0x10110:    add     eax, 1
0x10113:    add     rdx, 0x50
0x10117:    cmp     ecx, eax
0x10119:    je      0x10154
0x1011b:    cmp     byte ptr [rdx], 0
0x1011e:    je      0x10110
0x10120:    mov     dword ptr [rdi + 0x50], eax
0x10123:    ret
0x10124:    nop     dword ptr [rax]
0x10128:    test    edx, edx
0x1012a:    je      0x10154
0x1012c:    mov     rdx, qword ptr [rdi + 0x18]
0x10130:    mov     eax, dword ptr [rdi + 0x28]
0x10133:    mov     dword ptr [rdi + 0x50], 0
0x1013a:    cmp     byte ptr [rdx], 0x2f
0x1013d:    jne     0x10150
0x1013f:    cmp     byte ptr [rdx + 1], 0
0x10143:    jne     0x10150
0x10145:    ret
0x10146:    nop     word ptr cs:[rax + rax]
0x10150:    test    eax, eax
0x10152:    jne     0x10145
0x10154:    mov     eax, 0xffffffff
0x10159:    ret
0x1015a:    nop     word ptr [rax + rax]
0x10160:    endbr64
0x10164:    push    r13
0x10166:    push    r12
0x10168:    push    rbp
0x10169:    push    rbx
0x1016a:    sub     rsp, 0x18
0x1016e:    mov     rax, qword ptr fs:[0x28]
0x10177:    mov     qword ptr [rsp + 8], rax
0x1017c:    xor     eax, eax
0x1017e:    cmp     dword ptr [rdi + 0x64], 0x13
0x10182:    jle     0x1018d
0x10184:    cmp     byte ptr [rdi + 0x6d], 0
0x10188:    mov     rbx, rdi
0x1018b:    jne     0x101b0
0x1018d:    mov     rax, qword ptr [rsp + 8]
0x10192:    sub     rax, qword ptr fs:[0x28]
0x1019b:    jne     0x10236
0x101a1:    add     rsp, 0x18
0x101a5:    pop     rbx
0x101a6:    pop     rbp

```

```

0x101a7:      pop     r12
0x101a9:      pop     r13
0x101ab:      ret
0x101ac:      nop     dword ptr [rax]
0x101b0:      lea     r13, [rsp + 7]
0x101b5:      mov     rdi, r13
0x101b8:      call    0xa6a0
0x101bd:      movsxd  rdx, dword ptr [rbx + 0x64]
0x101c1:      mov     rdi, qword ptr [rbx + 0x20]
0x101c5:      mov     esi, 0x20
0x101ca:      mov     rbp, rdx
0x101cd:      lea     r12d, [rdx - 0x13]
0x101d1:      call    0x4a50
0x101d6:      mov     rsi, qword ptr [rbx + 0x20]
0x101da:      mov     rdi, qword ptr [rbx + 0x18]
0x101de:      mov     edx, r12d
0x101e1:      call    0x18200
0x101e6:      mov     rdi, qword ptr [rbx + 0x20]
0x101ea:      movsxd  rsi, r12d
0x101ed:      call    0x184d0
0x101f2:      mov     edx, dword ptr [rip + 0x1a128]
0x101f8:      mov     esi, dword ptr [rip + 0x1a152]
0x101fe:      mov     edi, dword ptr [rip + 0x1a150]
0x10204:      call    0xa570
0x10209:      mov     rdi, qword ptr [rbx + 8]
0x1020d:      xor     esi, esi
0x1020f:      mov     edx, 2
0x10214:      call    0xc680
0x10219:      mov     rdi, qword ptr [rbx + 8]
0x1021d:      mov     rsi, qword ptr [rbx + 0x20]
0x10221:      lea     edx, [rbp - 0x12]
0x10224:      call    0xc600
0x10229:      mov     rdi, r13
0x1022c:      call    0xa6b0
0x10231:      jmp     0x1018d
0x10236:      call    0x49d0
0x1023b:      nop     dword ptr [rax + rax]
0x10240:      endbr64
0x10244:      push    r15
0x10246:      push    r14
0x10248:      push    r13
0x1024a:      push    r12
0x1024c:      push    rbp
0x1024d:      push    rbx
0x1024e:      sub     rsp, 0x2f8
0x10255:      mov     rax, qword ptr fs:[0x28]
0x1025e:      mov     qword ptr [rsp + 0x2e8], rax
0x10266:      xor     eax, eax

```

```

0x10268:    cmp     byte ptr [rdi + 0x6d], 0
0x1026c:    jne     0x102a0
0x1026e:    mov     rax, qword ptr [rsp + 0x2e8]
0x10276:    sub     rax, qword ptr fs:[0x28]
0x1027f:    jne     0x103f2
0x10285:    add     rsp, 0x2f8
0x1028c:    pop     rbx
0x1028d:    pop     rbp
0x1028e:    pop     r12
0x10290:    pop     r13
0x10292:    pop     r14
0x10294:    pop     r15
0x10296:    ret
0x10297:    nop     word ptr [rax + rax]
0x102a0:    mov     eax, dword ptr [rdi + 0x64]
0x102a3:    lea     r14, [rsp + 0xf]
0x102a8:    mov     rbx, rdi
0x102ab:    mov     rdi, r14
0x102ae:    lea     r12d, [rax - 4]
0x102b2:    call    0xa6a0
0x102b7:    mov     qword ptr [rsp + 0x18], -1
0x102c0:    cmp     r12d, 5
0x102c4:    jle     0x10350
0x102ca:    mov     rdi, qword ptr [rbx + 0x18]
0x102ce:    xor     esi, esi
0x102d0:    lea     rdx, [rsp + 0x10]
0x102d5:    call    0x1c490
0x102da:    test    eax, eax
0x102dc:    js      0x10350
0x102de:    cmp     qword ptr [rsp + 0x18], -1
0x102e4:    je      0x10350
0x102e6:    call    0x4a60
0x102eb:    test    eax, eax
0x102ed:    je      0x103e8
0x102f3:    mov     rdi, qword ptr [rsp + 0x28]
0x102f8:    xor     eax, eax
0x102fa:    test    rdi, rdi
0x102fd:    mov     edx, 0xc
0x10302:    mov     r8d, 1
0x10308:    cmovs   rdi, rax
0x1030c:    imul    rdi, qword ptr [rsp + 0x10]
0x10312:    mov     eax, 0x9c
0x10317:    mov     ecx, 1
0x1031c:    lea     rsi, [rsp + 0x50]
0x10321:    cmp     rdi, 0x2710
0x10328:    cmovge  edx, eax
0x1032b:    call    0x1c520
0x10330:    mov     edi, dword ptr [rip + 0x1a026]

```

```

0x10336:      mov     rbp, rax
0x10339:      call    0xa530
0x1033e:      mov     edi, dword ptr [rip + 0x1a014]
0x10344:      call    0xa4f0
0x10349:      jmp     0x10388
0x1034b:      nop     dword ptr [rax + rax]
0x10350:      lea     rbp, [rsp + 0x50]
0x10355:      mov     ecx, 0x51
0x1035a:      movabs  rax, 0x2020202020202020
0x10364:      mov     rdi, rbp
0x10367:      rep stosq      qword ptr [rdi], rax
0x1036a:      mov     eax, 0x2020
0x1036f:      mov     word ptr [rdi], ax
0x10372:      mov     byte ptr [rdi + 2], 0x20
0x10376:      xor     edi, edi
0x10378:      call    0xa530
0x1037d:      mov     edi, dword ptr [rip + 0x19f9d]
0x10383:      call    0xa4f0
0x10388:      mov     edi, dword ptr [rip + 0x19f92]
0x1038e:      movsxd  r15, r12d
0x10391:      call    0xa510
0x10396:      mov     rdi, rbp
0x10399:      mov     r13d, dword ptr [rbx + 0x64]
0x1039d:      call    0x49b0
0x103a2:      mov     rdi, qword ptr [rbx + 8]
0x103a6:      cmp     rax, r15
0x103a9:      lea     edx, [r13 - 2]
0x103ad:      cmova   eax, r12d
0x103b1:      xor     esi, esi
0x103b3:      sub     edx, eax
0x103b5:      call    0xc680
0x103ba:      mov     rdi, rbp
0x103bd:      call    0x49b0
0x103c2:      mov     rdi, qword ptr [rbx + 8]
0x103c6:      mov     rsi, rbp
0x103c9:      cmp     r15, rax
0x103cc:      cmovae  r12d, eax
0x103d0:      mov     edx, r12d
0x103d3:      call    0xc600
0x103d8:      mov     rdi, r14
0x103db:      call    0xa6b0
0x103e0:      jmp     0x1026e
0x103e5:      nop     dword ptr [rax]
0x103e8:      mov     rdi, qword ptr [rsp + 0x20]
0x103ed:      jmp     0x102f8
0x103f2:      call    0x49d0
0x103f7:      nop     word ptr [rax + rax]
0x10400:      endbr64

```



```

0x10404:      cmp     byte ptr [rdi + 0x6d], 0
0x10408:      jne     0x10410
0x1040a:      ret
0x1040b:      nop     dword ptr [rax + rax]
0x10410:      xor     eax, eax
0x10412:      jmp     0xec50
0x10417:      nop     word ptr [rax + rax]
0x10420:      endbr64
0x10424:      push    r12
0x10426:      push    rbp
0x10427:      push    rbx
0x10428:      sub     rsp, 0x2a0
0x1042f:      mov     rax, qword ptr fs:[0x28]
0x10438:      mov     qword ptr [rsp + 0x298], rax
0x10440:      xor     eax, eax
0x10442:      cmp     edx, 5
0x10445:      ja      0x10660
0x1044b:      mov     ebp, ecx
0x1044d:      mov     edx, edx
0x1044f:      lea     rcx, [rip + 0x1146e]
0x10456:      mov     rbx, rdi
0x10459:      movsxd  rax, dword ptr [rcx + rdx*4]
0x1045d:      add     rax, rcx
0x10460:      jmp     rax
0x10463:      -----

```

```

0x104ce:      -----
0x104cf:      -----
0x104d0:      mov     rax, qword ptr [rsp + 0x298]
0x104d8:      sub     rax, qword ptr fs:[0x28]
0x104e1:      jne     0x1075a
0x104e7:      add     rsp, 0x2a0
0x104ee:      pop     rbx
0x104ef:      pop     rbp
0x104f0:      pop     r12
0x104f2:      ret
0x104f3:      nop     dword ptr [rax + rax]
0x104f8:      mov     rdx, rsp
0x104fb:      xor     eax, eax
0x104fd:      call    0xe560
0x10502:      movsxd  rax, dword ptr [rbx + 0x64]
0x10506:      mov     rdx, qword ptr [rbx + 0x20]
0x1050a:      movsxd  rcx, ebp
0x1050d:      sub     rax, rcx
0x10510:      lea     rax, [rdx + rax - 2]
0x10515:      mov     rdx, qword ptr [rsp]

```

```

0x10519:      mov     qword ptr [rax], rdx
0x1051c:      movzx   edx, word ptr [rsp + 8]
0x10521:      mov     word ptr [rax + 8], dx
0x10525:      jmp     0x104d0
0x10527:      nop     word ptr [rax + rax]
0x10530:      movsxd  rax, dword ptr [rdi + 0x64]
0x10534:      mov     rdx, qword ptr [rdi + 0x20]
0x10538:      movsxd  rcx, ebp
0x1053b:      movsxd  rsi, esi
0x1053e:      sub     rax, rcx
0x10541:      lea     rdx, [rdx + rax - 2]
0x10546:      lea     rax, [rsi + rsi*4]
0x1054a:      shl     rax, 4
0x1054e:      add     rax, qword ptr [rdi + 0x38]
0x10552:      lea     rcx, [rax + 0x38]
0x10556:      mov     rax, qword ptr [rax + 0x38]
0x1055a:      mov     qword ptr [rdx], rax
0x1055d:      mov     eax, dword ptr [rcx + 8]
0x10560:      mov     dword ptr [rdx + 8], eax
0x10563:      movzx   eax, word ptr [rcx + 0xc]
0x10567:      mov     word ptr [rdx + 0xc], ax
0x1056b:      movzx   eax, byte ptr [rcx + 0xe]
0x1056f:      mov     byte ptr [rdx + 0xe], al
0x10572:      jmp     0x104d0
0x10577:      nop     word ptr [rax + rax]
0x10580:      movsxd  rsi, esi
0x10583:      mov     edx, 0xc
0x10588:      mov     ecx, 1
0x1058d:      mov     r8d, 1
0x10593:      lea     rax, [rsi + rsi*4]
0x10597:      mov     rsi, rsp
0x1059a:      shl     rax, 4
0x1059e:      add     rax, qword ptr [rdi + 0x38]
0x105a2:      mov     rdi, qword ptr [rax + 8]
0x105a6:      mov     eax, 0x9c
0x105ab:      cmp     rdi, 0x2710
0x105b2:      cmovge  edx, eax
0x105b5:      call    0x1c520
0x105ba:      mov     rdi, rax
0x105bd:      mov     r12, rax
0x105c0:      call    0x49b0
0x105c5:      mov     edx, 0xa
0x105ca:      movsxd  rsi, dword ptr [rbx + 0x64]
0x105ce:      movsxd  rcx, ebp
0x105d1:      cmp     rax, rdx
0x105d4:      cmovbe  rdx, rax
0x105d8:      sub     rsi, rcx
0x105db:      mov     rcx, qword ptr [rbx + 0x20]

```

```

0x105df:    lea     rcx, [rcx + rsi - 2]
0x105e4:    cmp     rax, 9
0x105e8:    ja      0x10621
0x105ea:    movabs  rdi, 0x2020202020202020
0x105f4:    mov     eax, 0xa
0x105f9:    sub     eax, edx
0x105fb:    movsxd  r10, eax
0x105fe:    cmp     eax, 8
0x10601:    jae     0x106e0
0x10607:    test    al, 4
0x10609:    jne     0x10734
0x1060f:    test    eax, eax
0x10611:    je      0x1061e
0x10613:    mov     byte ptr [rcx], 0x20
0x10616:    test    al, 2
0x10618:    jne     0x10749
0x1061e:    add     rcx, r10
0x10621:    cmp     edx, 8
0x10624:    jae     0x10690
0x10626:    test    dl, 4
0x10629:    jne     0x1071e
0x1062f:    test    edx, edx
0x10631:    je      0x104d0
0x10637:    movzx   eax, byte ptr [r12]
0x1063c:    mov     byte ptr [rcx], al
0x1063e:    test    dl, 2
0x10641:    je      0x104d0
0x10647:    mov     edx, edx
0x10649:    movzx   eax, word ptr [r12 + rdx - 2]
0x1064f:    mov     word ptr [rcx + rdx - 2], ax
0x10654:    jmp     0x104d0
0x10659:    nop     dword ptr [rax]
0x10660:    mov     rax, qword ptr [rsp + 0x298]
0x10668:    sub     rax, qword ptr fs:[0x28]
0x10671:    jne     0x1075a
0x10677:    add     rsp, 0x2a0
0x1067e:    lea     rdi, [rip + 0x10b67]
0x10685:    pop     rbx
0x10686:    pop     rbp
0x10687:    pop     r12
0x10689:    jmp     0x8b00
0x1068e:    nop
0x10690:    mov     rax, qword ptr [r12]
0x10694:    mov     qword ptr [rcx], rax
0x10697:    mov     eax, edx
0x10699:    mov     rsi, qword ptr [r12 + rax - 8]
0x1069e:    mov     qword ptr [rcx + rax - 8], rsi
0x106a3:    lea     rsi, [rcx + 8]

```

```

0x106a7:    and     rsi, 0xfffffffffffffff8
0x106ab:    sub     rcx, rsi
0x106ae:    add     edx, ecx
0x106b0:    sub     r12, rcx
0x106b3:    and     edx, 0xffffffff8
0x106b6:    cmp     edx, 8
0x106b9:    jb      0x104d0
0x106bf:    and     edx, 0xffffffff8
0x106c2:    xor     eax, eax
0x106c4:    mov     ecx, eax
0x106c6:    add     eax, 8
0x106c9:    mov     rdi, qword ptr [r12 + rcx]
0x106cd:    mov     qword ptr [rsi + rcx], rdi
0x106d1:    cmp     eax, edx
0x106d3:    jb      0x106c4
0x106d5:    jmp     0x104d0
0x106da:    nop     word ptr [rax + rax]
0x106e0:    mov     esi, eax
0x106e2:    lea     r8, [rcx + 8]
0x106e6:    mov     qword ptr [rcx], rdi
0x106e9:    and     r8, 0xfffffffffffffff8
0x106ed:    mov     qword ptr [rcx + rsi - 8], rdi
0x106f2:    mov     rsi, rcx
0x106f5:    sub     rsi, r8
0x106f8:    add     eax, esi
0x106fa:    and     eax, 0xffffffff8
0x106fd:    cmp     eax, 8
0x10700:    jb      0x1061e
0x10706:    and     eax, 0xffffffff8
0x10709:    xor     esi, esi
0x1070b:    mov     r9d, esi
0x1070e:    add     esi, 8
0x10711:    mov     qword ptr [r8 + r9], rdi
0x10715:    cmp     esi, eax
0x10717:    jb      0x1070b
0x10719:    jmp     0x1061e
0x1071e:    mov     eax, dword ptr [r12]
0x10722:    mov     edx, edx
0x10724:    mov     dword ptr [rcx], eax
0x10726:    mov     eax, dword ptr [r12 + rdx - 4]
0x1072b:    mov     dword ptr [rcx + rdx - 4], eax
0x1072f:    jmp     0x104d0
0x10734:    mov     eax, eax
0x10736:    mov     dword ptr [rcx], 0x20202020
0x1073c:    mov     dword ptr [rcx + rax - 4], 0x20202020
0x10744:    jmp     0x1061e
0x10749:    mov     eax, eax
0x1074b:    mov     esi, 0x2020

```

```

0x10750:      mov     word ptr [rcx + rax - 2], si
0x10755:      jmp     0x1061e
0x1075a:      call    0x49d0
0x1075f:      nop
0x10760:      push    r15
0x10762:      push    r14
0x10764:      mov     r14d, 4
0x1076a:      push    r13
0x1076c:      mov     r13, rdi
0x1076f:      push    r12
0x10771:      push    rbp
0x10772:      push    rbx
0x10773:      mov     ebx, esi
0x10775:      mov     esi, 0x20
0x1077a:      sub     rsp, 0x28
0x1077e:      movsxd  rdx, dword ptr [rdi + 0x64]
0x10782:      mov     rdi, qword ptr [rdi + 0x20]
0x10786:      call    0x4a50
0x1078b:      mov     ebp, dword ptr [r13 + 0x64]
0x1078f:      cmp     ebp, 0x27
0x10792:      jle     0x107b0
0x10794:      movsxd  rax, dword ptr [r13 + 0x80]
0x1079b:      cmp     ebp, 0x49
0x1079e:      jg      0x107a5
0x107a0:      cmp     eax, 5
0x107a3:      je      0x107b0
0x107a5:      lea     rdx, [rip + 0x11234]
0x107ac:      movsxd  r14, dword ptr [rdx + rax*4]
0x107b0:      movsxd  rax, ebx
0x107b3:      lea     r12, [rax + rax*4]
0x107b7:      mov     rax, qword ptr [r13 + 0x38]
0x107bb:      shl     r12, 4
0x107bf:      add     rax, r12
0x107c2:      test    ebx, ebx
0x107c4:      jle     0x107e8
0x107c6:      cmp     byte ptr [rax + 0x48], 4
0x107ca:      ja      0x4f20
0x107d0:      movzx   edx, byte ptr [rax + 0x48]
0x107d4:      lea     rcx, [rip + 0x11101]
0x107db:      movsxd  rdx, dword ptr [rcx + rdx*4]
0x107df:      add     rdx, rcx
0x107e2:      jmp     rdx
0x107e5:      -----

0x107e6:      -----
0x107e7:      -----

```

```

0x107e8:      mov     rdx, qword ptr [r13 + 0x18]
0x107ec:      mov     byte ptr [rsp + 8], 0
0x107f1:      cmp     byte ptr [rdx], 0x2f
0x107f4:      je      0x10c18
0x107fa:      mov     r15, qword ptr [rax]
0x107fd:      mov     rdi, r15
0x10800:      call    0x49b0
0x10805:      movsxd   rdx, ebp
0x10808:      mov     rbp, rdx
0x1080b:      sub     rbp, r14
0x1080e:      cmp     rax, rbp
0x10811:      jbe     0x1082b
0x10813:      movsxd   rcx, dword ptr [r13 + 0x74]
0x10817:      lea     rsi, [rbp + rcx]
0x1081c:      cmp     rax, rsi
0x1081f:      jb      0x10a90
0x10825:      sub     rax, rcx
0x10828:      add     r15, rcx
0x1082b:      cmp     rbp, rax
0x1082e:      mov     rsi, r15
0x10831:      cmova    rbp, rax
0x10835:      mov     rax, qword ptr [r13 + 0x20]
0x10839:      mov     rdx, rbp
0x1083c:      lea     rdi, [rax + 1]
0x10840:      call    0x4bb0
0x10845:      mov     rax, qword ptr [r13 + 0x20]
0x10849:      mov     rsi, rbp
0x1084c:      lea     rdi, [rax + 1]
0x10850:      call    0x184d0
0x10855:      movzx    ecx, byte ptr [rsp + 8]
0x1085a:      test    cl, cl
0x1085c:      je      0x10866
0x1085e:      mov     rax, qword ptr [r13 + 0x20]
0x10862:      mov     byte ptr [rax + rbp + 1], cl
0x10866:      mov     eax, dword ptr [r13 + 0x64]
0x1086a:      cmp     eax, 0x27
0x1086d:      jle     0x108a0
0x1086f:      mov     edx, dword ptr [r13 + 0x80]
0x10876:      cmp     edx, 4
0x10879:      je      0x108a0
0x1087b:      jg      0x109e0
0x10881:      cmp     edx, 1
0x10884:      jg      0x109c0
0x1088a:      test    edx, edx
0x1088c:      jns     0x10c30
0x10892:      lea     rdi, [rip + 0x10953]
0x10899:      call    0x8b00
0x1089e:      nop

```

```

0x108a0:    mov     rax, qword ptr [r13 + 0x38]
0x108a4:    cmp     byte ptr [rax + r12 + 0x49], 0
0x108aa:    je      0x108b9
0x108ac:    movsxd  rax, dword ptr [r13 + 0x64]
0x108b0:    mov     rdx, qword ptr [r13 + 0x20]
0x108b4:    mov     byte ptr [rdx + rax - 3], 0x2a
0x108b9:    cmp     ebx, dword ptr [r13 + 0x28]
0x108bd:    je      0x10948
0x108c3:    mov     rax, qword ptr [r13 + 0x38]
0x108c7:    mov     ecx, dword ptr [rip + 0x1938f]
0x108cd:    movzx   edx, byte ptr [rax + r12 + 0x49]
0x108d3:    test    ecx, ecx
0x108d5:    je      0x108e8
0x108d7:    add     rax, r12
0x108da:    cmp     byte ptr [rax + 0x48], 0
0x108de:    movzx   edx, byte ptr [rax + 0x49]
0x108e2:    jne     0x1097e
0x108e8:    test    dl, dl
0x108ea:    jne     0x109a0
0x108f0:    mov     esi, dword ptr [rip + 0x19a46]
0x108f6:    mov     r9d, dword ptr [rip + 0x19a27]
0x108fd:    mov     edi, dword ptr [rip + 0x19a3d]
0x10903:    mov     edx, r9d
0x10906:    call    0xa570
0x1090b:    sub     ebx, dword ptr [r13 + 0x2c]
0x1090f:    mov     rdi, qword ptr [r13 + 8]
0x10913:    mov     edx, 1
0x10918:    lea     esi, [rbx + 1]
0x1091b:    call    0xc680
0x10920:    mov     eax, dword ptr [r13 + 0x64]
0x10924:    mov     rsi, qword ptr [r13 + 0x20]
0x10928:    mov     rdi, qword ptr [r13 + 8]
0x1092c:    add     rsp, 0x28
0x10930:    pop     rbx
0x10931:    lea     edx, [rax - 2]
0x10934:    pop     rbp
0x10935:    pop     r12
0x10937:    pop     r13
0x10939:    pop     r14
0x1093b:    pop     r15
0x1093d:    jmp     0xc600
0x10942:    nop     word ptr [rax + rax]
0x10948:    cmp     byte ptr [r13 + 0x6c], 1
0x1094d:    mov     rdx, qword ptr [r13 + 0x20]
0x10951:    sbb     eax, eax
0x10953:    and     eax, 0xffffffff
0x10956:    add     eax, 0x3e
0x10959:    mov     byte ptr [rdx], al

```

```

0x1095b:    mov     rax, qword ptr [r13 + 0x38]
0x1095f:    mov     esi, dword ptr [rip + 0x192f7]
0x10965:    add     rax, r12
0x10968:    test    esi, esi
0x1096a:    je      0x10a50
0x10970:    cmp     byte ptr [rax + 0x48], 0
0x10974:    movzx   edx, byte ptr [rax + 0x49]
0x10978:    je      0x10a50
0x1097e:    test    dl, dl
0x10980:    je      0x10b10
0x10986:    cmp     ebx, dword ptr [r13 + 0x28]
0x1098a:    jne     0x109a0
0x1098c:    cmp     byte ptr [r13 + 0x6c], 1
0x10991:    je      0x10c08
0x10997:    nop     word ptr [rax + rax]
0x109a0:    mov     esi, dword ptr [rip + 0x1998e]
0x109a6:    mov     r9d, dword ptr [rip + 0x19977]
0x109ad:    mov     edi, dword ptr [rip + 0x19985]
0x109b3:    jmp     0x10903
0x109b8:    nop     dword ptr [rax + rax]
0x109c0:    mov     ecx, 0xb
0x109c5:    mov     esi, ebx
0x109c7:    mov     rdi, r13
0x109ca:    xor     eax, eax
0x109cc:    call    0x10420
0x109d1:    jmp     0x108a0
0x109d6:    nop     word ptr cs:[rax + rax]
0x109e0:    cmp     edx, 5
0x109e3:    jne     0x10892
0x109e9:    cmp     eax, 0x49
0x109ec:    jle     0x108a0
0x109f2:    xor     edx, edx
0x109f4:    mov     esi, ebx
0x109f6:    mov     rdi, r13
0x109f9:    xor     eax, eax
0x109fb:    mov     ecx, 0x36
0x10a00:    call    0x10420
0x10a05:    mov     esi, ebx
0x10a07:    mov     rdi, r13
0x10a0a:    xor     eax, eax
0x10a0c:    mov     ecx, 0x26
0x10a11:    mov     edx, 1
0x10a16:    call    0x10420
0x10a1b:    mov     esi, ebx
0x10a1d:    mov     rdi, r13
0x10a20:    xor     eax, eax
0x10a22:    mov     ecx, 0x16
0x10a27:    mov     edx, 2

```



```

0x10a2c:    call    0x10420
0x10a31:    mov     ecx, 0xb
0x10a36:    mov     esi, ebx
0x10a38:    mov     rdi, r13
0x10a3b:    mov     edx, 3
0x10a40:    xor     eax, eax
0x10a42:    call    0x10420
0x10a47:    jmp     0x108a0
0x10a4c:    nop     dword ptr [rax]
0x10a50:    movzx   edx, byte ptr [rax + 0x49]
0x10a54:    cmp     ebx, dword ptr [r13 + 0x28]
0x10a58:    jne     0x108e8
0x10a5e:    cmp     byte ptr [r13 + 0x6c], 1
0x10a63:    jne     0x108e8
0x10a69:    mov     esi, dword ptr [rip + 0x198d9]
0x10a6f:    test    dl, dl
0x10a71:    jne     0x10c08
0x10a77:    mov     r9d, dword ptr [rip + 0x198ce]
0x10a7e:    test    dl, dl
0x10a80:    je      0x108fd
0x10a86:    jmp     0x109ad
0x10a8b:    nop     dword ptr [rax + rax]
0x10a90:    add     r14, rax
0x10a93:    mov     rcx, rdx
0x10a96:    sub     rcx, r14
0x10a99:    sub     r14, rdx
0x10a9c:    add     rax, rcx
0x10a9f:    add     r15, r14
0x10aa2:    jmp     0x1082b
0x10aa7:    nop     word ptr [rax + rax]
0x10ab0:    mov     byte ptr [rsp + 8], 0x3d
0x10ab5:    add     r14, 1
0x10ab9:    jmp     0x107fa
0x10abe:    nop
0x10ac0:    mov     byte ptr [rsp + 8], 0x7c
0x10ac5:    jmp     0x10ab5
0x10ac7:    nop     word ptr [rax + rax]
0x10ad0:    movzx   ecx, byte ptr [rax + 0x4a]
0x10ad4:    mov     byte ptr [rsp + 8], cl
0x10ad8:    test    cl, cl
0x10ada:    je      0x107fa
0x10ae0:    mov     byte ptr [rsp + 8], 0x2a
0x10ae5:    jmp     0x10ab5
0x10ae7:    nop     word ptr [rax + rax]
0x10af0:    mov     byte ptr [rsp + 8], 0x40
0x10af5:    jmp     0x10ab5
0x10af7:    nop     word ptr [rax + rax]
0x10b00:    mov     byte ptr [rsp + 8], 0x2f

```

```

0x10b05:      jmp      0x10ab5
0x10b07:      nop      word ptr [rax + rax]
0x10b10:      mov      rdi, qword ptr [r13 + 8]
0x10b14:      mov      esi, ebx
0x10b16:      sub      esi, dword ptr [r13 + 0x2c]
0x10b1a:      mov      edx, 1
0x10b1f:      add      esi, 1
0x10b22:      call     0xc680
0x10b27:      mov      rax, qword ptr [r13 + 0x38]
0x10b2b:      cmp      byte ptr [rax + r12 + 0x49], 0
0x10b31:      jne      0x10c50
0x10b37:      mov      eax, dword ptr [rip + 0x197ff]
0x10b3d:      mov      r15d, dword ptr [rip + 0x197fc]
0x10b44:      mov      dword ptr [rsp + 0xc], eax
0x10b48:      cmp      ebx, dword ptr [r13 + 0x28]
0x10b4c:      je       0x10c70
0x10b52:      mov      eax, dword ptr [rip + 0x197cc]
0x10b58:      mov      dword ptr [rsp + 8], eax
0x10b5c:      mov      edx, dword ptr [rsp + 8]
0x10b60:      mov      esi, dword ptr [rsp + 0xc]
0x10b64:      mov      edi, r15d
0x10b67:      call     0xa570
0x10b6c:      mov      rax, qword ptr [r13 + 0x20]
0x10b70:      mov      rdi, qword ptr [r13 + 8]
0x10b74:      movsx     esi, byte ptr [rax]
0x10b77:      call     0xc650
0x10b7c:      mov      rax, qword ptr [r13 + 0x38]
0x10b80:      lea      rdx, [rax + r12]
0x10b84:      cmp      byte ptr [rdx + 0x4b], 0
0x10b88:      je       0x10c90
0x10b8e:      movsx     edi, byte ptr [rdx + 0x4e]
0x10b92:      cmp      dil, 2
0x10b96:      je       0x10bb6
0x10b98:      movsx     esi, byte ptr [rdx + 0x4c]
0x10b9c:      cmp      ebx, dword ptr [r13 + 0x28]
0x10ba0:      jne      0x10bad
0x10ba2:      cmp      byte ptr [r13 + 0x6c], 1
0x10ba7:      je       0x10d52
0x10bad:      movsx     edx, byte ptr [rdx + 0x4d]
0x10bb1:      call     0xa570
0x10bb6:      mov      rax, qword ptr [r13 + 0x20]
0x10bba:      mov      rdi, qword ptr [r13 + 8]
0x10bbe:      lea      edx, [rbp + 1]
0x10bc1:      lea      rsi, [rax + 1]
0x10bc5:      call     0xc600
0x10bca:      mov      eax, dword ptr [r13 + 0x64]
0x10bce:      mov      rdi, qword ptr [r13 + 8]
0x10bd2:      lea      edx, [rax - 4]

```

```

0x10bd5:    mov     rax, qword ptr [r13 + 0x20]
0x10bd9:    sub     edx, ebp
0x10bdb:    lea     rsi, [rax + rbp + 2]
0x10be0:    call    0xc600
0x10be5:    mov     edx, dword ptr [rsp + 8]
0x10be9:    mov     esi, dword ptr [rsp + 0xc]
0x10bed:    mov     edi, r15d
0x10bf0:    add     rsp, 0x28
0x10bf4:    pop     rbx
0x10bf5:    pop     rbp
0x10bf6:    pop     r12
0x10bf8:    pop     r13
0x10bfa:    pop     r14
0x10bfc:    pop     r15
0x10bfe:    jmp     0xa570
0x10c03:    nop     dword ptr [rax + rax]
0x10c08:    mov     esi, dword ptr [rip + 0x19736]
0x10c0e:    jmp     0x10a77
0x10c13:    nop     dword ptr [rax + rax]
0x10c18:    cmp     byte ptr [rdx + 1], 0
0x10c1c:    jne     0x107fa
0x10c22:    jmp     0x107c6
0x10c27:    nop     word ptr [rax + rax]
0x10c30:    mov     ecx, 0x10
0x10c35:    mov     esi, ebx
0x10c37:    mov     rdi, r13
0x10c3a:    xor     eax, eax
0x10c3c:    call    0x10420
0x10c41:    jmp     0x108a0
0x10c46:    nop     word ptr cs:[rax + rax]
0x10c50:    mov     eax, dword ptr [rip + 0x196de]
0x10c56:    mov     r15d, dword ptr [rip + 0x196db]
0x10c5d:    mov     dword ptr [rsp + 0xc], eax
0x10c61:    jmp     0x10b48
0x10c66:    nop     word ptr cs:[rax + rax]
0x10c70:    cmp     byte ptr [r13 + 0x6c], 1
0x10c75:    jne     0x10b52
0x10c7b:    mov     eax, dword ptr [rip + 0x196cb]
0x10c81:    mov     dword ptr [rsp + 8], eax
0x10c85:    jmp     0x10b5c
0x10c8a:    nop     word ptr [rax + rax]
0x10c90:    mov     r14, qword ptr [rip + 0x1adf9]
0x10c97:    mov     byte ptr [rdx + 0x4b], 1
0x10c9b:    test    r14, r14
0x10c9e:    je      0x10d49
0x10ca4:    mov     qword ptr [rsp + 0x18], 0
0x10cad:    mov     qword ptr [rsp + 0x10], 0
0x10cb6:    nop     word ptr cs:[rax + rax]

```

```

0x10cc0:    mov     rsi, qword ptr [rax + r12]
0x10cc4:    mov     rdi, qword ptr [r14]
0x10cc7:    mov     edx, 0x15
0x10ccc:    call    0x4a40
0x10cd1:    test    eax, eax
0x10cd3:    jne     0x10cee
0x10cd5:    mov     rdi, qword ptr [r14]
0x10cd8:    call    0x49b0
0x10cdd:    cmp     qword ptr [rsp + 0x10], rax
0x10ce2:    jae     0x10cee
0x10ce4:    mov     qword ptr [rsp + 0x18], r14
0x10ce9:    mov     qword ptr [rsp + 0x10], rax
0x10cee:    mov     r14, qword ptr [r14 + 0x18]
0x10cf2:    mov     rax, qword ptr [r13 + 0x38]
0x10cf6:    test    r14, r14
0x10cf9:    jne     0x10cc0
0x10cfb:    mov     rsi, qword ptr [rsp + 0x18]
0x10d00:    lea     rdx, [rax + r12]
0x10d04:    test    rsi, rsi
0x10d07:    je      0x10d49
0x10d09:    mov     eax, dword ptr [rsi + 0x10]
0x10d0c:    xor     ecx, ecx
0x10d0e:    cmp     eax, -1
0x10d11:    cmovne  eax, ecx
0x10d14:    mov     byte ptr [rdx + 0x4e], al
0x10d17:    movsx   edi, al
0x10d1a:    mov     eax, dword ptr [rsi + 8]
0x10d1d:    mov     ecx, eax
0x10d1f:    cmp     eax, -1
0x10d22:    movzx   eax, byte ptr [rip + 0x19613]
0x10d29:    cmovne  eax, ecx
0x10d2c:    mov     byte ptr [rdx + 0x4c], al
0x10d2f:    mov     eax, dword ptr [rsi + 0xc]
0x10d32:    mov     ecx, eax
0x10d34:    cmp     eax, -1
0x10d37:    movzx   eax, byte ptr [rip + 0x195e6]
0x10d3e:    cmovne  eax, ecx
0x10d41:    mov     byte ptr [rdx + 0x4d], al
0x10d44:    jmp     0x10b92
0x10d49:    mov     byte ptr [rdx + 0x4e], 2
0x10d4d:    jmp     0x10bb6
0x10d52:    mov     edx, dword ptr [rsp + 8]
0x10d56:    mov     edi, r15d
0x10d59:    call    0xa570
0x10d5e:    jmp     0x10bb6
0x10d63:    nop     word ptr cs:[rax + rax]
0x10d6e:    nop
0x10d70:    push    r13

```

```

0x10d72:    push    r12
0x10d74:    push    rbp
0x10d75:    push    rbx
0x10d76:    mov     rbx, rdi
0x10d79:    sub     rsp, 0x18
0x10d7d:    mov     rax, qword ptr fs:[0x28]
0x10d86:    mov     qword ptr [rsp + 8], rax
0x10d8b:    xor     eax, eax
0x10d8d:    lea     r12, [rsp + 7]
0x10d92:    mov     rdi, r12
0x10d95:    call    0xa6a0
0x10d9a:    mov     ebp, dword ptr [rbx + 0x2c]
0x10d9d:    mov     eax, ebp
0x10d9f:    cmp     ebp, dword ptr [rbx + 0x68]
0x10da2:    jl      0x10dcB
0x10da4:    jmp     0x10dd9
0x10da6:    nop     word ptr cs:[rax + rax]
0x10db0:    cmp     byte ptr [rbx + 0x6d], 0
0x10db4:    je      0x10dc0
0x10db6:    mov     esi, ebp
0x10db8:    mov     rdi, rbx
0x10dbb:    call    0x10760
0x10dc0:    add     ebp, 1
0x10dc3:    cmp     ebp, dword ptr [rbx + 0x68]
0x10dc6:    jge     0x10dd9
0x10dc8:    mov     eax, dword ptr [rbx + 0x2c]
0x10dcb:    mov     edx, ebp
0x10dcd:    sub     edx, eax
0x10dcf:    mov     eax, dword ptr [rbx + 0x60]
0x10dd2:    sub     eax, 2
0x10dd5:    cmp     edx, eax
0x10dd7:    jl      0x10db0
0x10dd9:    mov     edx, dword ptr [rip + 0x19545]
0x10ddf:    xor     edi, edi
0x10de1:    mov     esi, 7
0x10de6:    call    0xa570
0x10deb:    movsxd  rdx, dword ptr [rbx + 0x64]
0x10def:    mov     rdi, qword ptr [rbx + 0x20]
0x10df3:    mov     esi, 0x20
0x10df8:    call    0x4a50
0x10dfd:    mov     eax, dword ptr [rbx + 0x60]
0x10e00:    lea     r13d, [rax - 2]
0x10e04:    mov     eax, dword ptr [rbx + 0x14]
0x10e07:    cmp     r13d, eax
0x10e0a:    cmovg   r13d, eax
0x10e0e:    cmp     r13d, ebp
0x10e11:    jle     0x10e49
0x10e13:    nop     dword ptr [rax + rax]

```

```

0x10e18:    mov     rdi, qword ptr [rbx + 8]
0x10e1c:    mov     esi, ebp
0x10e1e:    sub     esi, dword ptr [rbx + 0x2c]
0x10e21:    mov     edx, 1
0x10e26:    add     esi, 1
0x10e29:    add     ebp, 1
0x10e2c:    call    0xc680
0x10e31:    mov     eax, dword ptr [rbx + 0x64]
0x10e34:    mov     rsi, qword ptr [rbx + 0x20]
0x10e38:    mov     rdi, qword ptr [rbx + 8]
0x10e3c:    lea     edx, [rax - 2]
0x10e3f:    call    0xc600
0x10e44:    cmp     r13d, ebp
0x10e47:    jne     0x10e18
0x10e49:    mov     eax, dword ptr [rbx + 0x68]
0x10e4c:    mov     rdi, r12
0x10e4f:    mov     dword ptr [rbx + 0x14], eax
0x10e52:    call    0xa6b0
0x10e57:    mov     rax, qword ptr [rsp + 8]
0x10e5c:    sub     rax, qword ptr fs:[0x28]
0x10e65:    jne     0x10e72
0x10e67:    add     rsp, 0x18
0x10e6b:    pop     rbx
0x10e6c:    pop     rbp
0x10e6d:    pop     r12
0x10e6f:    pop     r13
0x10e71:    ret
0x10e72:    call    0x49d0
0x10e77:    nop     word ptr [rax + rax]
0x10e80:    endbr64
0x10e84:    cmp     byte ptr [rdi + 0x6d], 0
0x10e88:    jne     0x10e90
0x10e8a:    ret
0x10e8b:    nop     dword ptr [rax + rax]
0x10e90:    xor     eax, eax
0x10e92:    jmp     0x10d70
0x10e97:    nop     word ptr [rax + rax]
0x10ea0:    endbr64
0x10ea4:    cmp     byte ptr [rdi + 0x6d], 0
0x10ea8:    jne     0x10eb0
0x10eaa:    ret
0x10eab:    nop     dword ptr [rax + rax]
0x10eb0:    jmp     0x10760
0x10eb5:    nop     word ptr cs:[rax + rax]
0x10ec0:    endbr64
0x10ec4:    push    r13
0x10ec6:    push    r12
0x10ec8:    push    rbp

```

```

0x10ec9:    push    rbx
0x10eca:    mov     rbx, rdi
0x10ecd:    sub     rsp, 0x18
0x10ed1:    movsxd  rdi, dword ptr [rdi + 0x64]
0x10ed5:    mov     rax, qword ptr fs:[0x28]
0x10ede:    mov     qword ptr [rsp + 8], rax
0x10ee3:    xor     eax, eax
0x10ee5:    call    0x1d670
0x10eea:    cmp     byte ptr [rbx + 0x6d], 0
0x10eee:    jne     0x10f10
0x10ef0:    mov     rax, qword ptr [rsp + 8]
0x10ef5:    sub     rax, qword ptr fs:[0x28]
0x10efe:    jne     0x11006
0x10f04:    add     rsp, 0x18
0x10f08:    pop     rbx
0x10f09:    pop     rbp
0x10f0a:    pop     r12
0x10f0c:    pop     r13
0x10f0e:    ret
0x10f0f:    nop
0x10f10:    lea     r13, [rsp + 7]
0x10f15:    mov     r12, rax
0x10f18:    mov     ebp, 1
0x10f1d:    mov     rdi, r13
0x10f20:    call    0xa6a0
0x10f25:    mov     esi, dword ptr [rip + 0x193f5]
0x10f2b:    xor     edi, edi
0x10f2d:    mov     edx, esi
0x10f2f:    call    0xa570
0x10f34:    cmp     dword ptr [rbx + 0x60], 2
0x10f38:    jle     0x10fa4
0x10f3a:    nop     word ptr [rax + rax]
0x10f40:    mov     rdi, qword ptr [rbx + 8]
0x10f44:    xor     edx, edx
0x10f46:    mov     esi, ebp
0x10f48:    add     ebp, 1
0x10f4b:    call    0xc680
0x10f50:    mov     rdi, qword ptr [rbx + 8]
0x10f54:    mov     esi, 0x20
0x10f59:    call    0xc650
0x10f5e:    mov     eax, dword ptr [rbx + 0x60]
0x10f61:    lea     edx, [rax - 1]
0x10f64:    cmp     edx, ebp
0x10f66:    jg      0x10f40
0x10f68:    cmp     eax, 2
0x10f6b:    jle     0x10fa4
0x10f6d:    mov     ebp, 1
0x10f72:    nop     word ptr [rax + rax]

```

```

0x10f78:    mov     eax, dword ptr [rbx + 0x64]
0x10f7b:    mov     rdi, qword ptr [rbx + 8]
0x10f7f:    mov     esi, ebp
0x10f81:    add     ebp, 1
0x10f84:    lea     edx, [rax - 1]
0x10f87:    call    0xc680
0x10f8c:    mov     rdi, qword ptr [rbx + 8]
0x10f90:    mov     esi, 0x20
0x10f95:    call    0xc650
0x10f9a:    mov     eax, dword ptr [rbx + 0x60]
0x10f9d:    sub     eax, 1
0x10fa0:    cmp     eax, ebp
0x10fa2:    jg      0x10f78
0x10fa4:    movsxd  rdx, dword ptr [rbx + 0x64]
0x10fa8:    mov     rdi, r12
0x10fab:    mov     esi, 0x20
0x10fb0:    call    0x4a50
0x10fb5:    mov     rdi, qword ptr [rbx + 8]
0x10fb9:    xor     edx, edx
0x10fbb:    xor     esi, esi
0x10fbd:    call    0xc680
0x10fc2:    mov     edx, dword ptr [rbx + 0x64]
0x10fc5:    mov     rdi, qword ptr [rbx + 8]
0x10fc9:    mov     rsi, r12
0x10fcc:    call    0xc600
0x10fd1:    mov     eax, dword ptr [rbx + 0x60]
0x10fd4:    mov     rdi, qword ptr [rbx + 8]
0x10fd8:    xor     edx, edx
0x10fda:    lea     esi, [rax - 1]
0x10fdd:    call    0xc680
0x10fe2:    mov     edx, dword ptr [rbx + 0x64]
0x10fe5:    mov     rdi, qword ptr [rbx + 8]
0x10fe9:    mov     rsi, r12
0x10fec:    call    0xc600
0x10ff1:    mov     rdi, r12
0x10ff4:    call    0x18d10
0x10ff9:    mov     rdi, r13
0x10ffc:    call    0xa6b0
0x11001:    jmp     0x10ef0
0x11006:    call    0x49d0
0x1100b:    nop     dword ptr [rax + rax]
0x11010:    endbr64
0x11014:    push    rbp
0x11015:    xor     eax, eax
0x11017:    mov     rbp, rdi
0x1101a:    call    0x10ec0
0x1101f:    xor     eax, eax
0x11021:    mov     rdi, rbp

```



```

0x11024:    call    0x10160
0x11029:    cmp     byte ptr [rbp + 0x6d], 0
0x1102d:    je      0x11039
0x1102f:    mov     rdi, rbp
0x11032:    xor     eax, eax
0x11034:    call    0xec50
0x11039:    xor     eax, eax
0x1103b:    mov     rdi, rbp
0x1103e:    call    0x10240
0x11043:    cmp     byte ptr [rbp + 0x6d], 0
0x11047:    jne     0x11050
0x11049:    pop     rbp
0x1104a:    ret
0x1104b:    nop     dword ptr [rax + rax]
0x11050:    mov     rdi, rbp
0x11053:    xor     eax, eax
0x11055:    pop     rbp
0x11056:    jmp     0x10d70
0x1105b:    nop     dword ptr [rax + rax]
0x11060:    endbr64
0x11064:    mov     edx, dword ptr [rdi + 0x68]
0x11067:    test    edx, edx
0x11069:    jle     0x110ab
0x1106b:    sub     edx, 1
0x1106e:    mov     rcx, qword ptr [rdi + 0x38]
0x11072:    lea     rdx, [rdx + rdx*4]
0x11076:    shl     rdx, 4
0x1107a:    lea     rax, [rcx + 0x49]
0x1107e:    lea     rdx, [rcx + rdx + 0x99]
0x11086:    nop     word ptr cs:[rax + rax]
0x11090:    cmp     byte ptr [rax], 0
0x11093:    jne     0x110a2
0x11095:    cmp     byte ptr [rax - 1], 0
0x11099:    je      0x110a2
0x1109b:    mov     byte ptr [rax], 1
0x1109e:    add     dword ptr [rdi + 0x70], 1
0x110a2:    add     rax, 0x50
0x110a6:    cmp     rax, rdx
0x110a9:    jne     0x11090
0x110ab:    ret
0x110ac:    nop     dword ptr [rax]
0x110b0:    endbr64
0x110b4:    mov     edx, dword ptr [rdi + 0x68]
0x110b7:    test    edx, edx
0x110b9:    jle     0x110ec
0x110bb:    sub     edx, 1
0x110be:    mov     rcx, qword ptr [rdi + 0x38]
0x110c2:    lea     rdx, [rdx + rdx*4]

```

```

0x110c6:    shl     rdx, 4
0x110ca:    lea     rax, [rcx + 0x49]
0x110ce:    lea     rdx, [rcx + rdx + 0x99]
0x110d6:    nop     word ptr cs:[rax + rax]
0x110e0:    mov     byte ptr [rax], 0
0x110e3:    add     rax, 0x50
0x110e7:    cmp     rax, rdx
0x110ea:    jne     0x110e0
0x110ec:    mov     dword ptr [rdi + 0x70], 0
0x110f3:    ret
0x110f4:    nop     word ptr cs:[rax + rax]
0x110ff:    nop
0x11100:    endbr64
0x11104:    mov     rax, qword ptr [rdi + 0x18]
0x11108:    ret
0x11109:    nop     dword ptr [rax]
0x11110:    endbr64
0x11114:    mov     eax, dword ptr [rip + 0x1a996]
0x1111a:    test    eax, eax
0x1111c:    jne     0x11120
0x1111e:    ret
0x1111f:    nop
0x11120:    push    rbp
0x11121:    xor     eax, eax
0x11123:    push    rbx
0x11124:    sub     rsp, 8
0x11128:    mov     dword ptr [rip + 0x1a97e], 0
0x11132:    call    0xc8c0
0x11137:    mov     edx, 4
0x1113c:    lea     rsi, [rip + 0xf3ef]
0x11143:    lea     rdi, [rip + 0x100af]
0x1114a:    mov     rbp, rax
0x1114d:    call    0x8070
0x11152:    mov     rdi, rbp
0x11155:    mov     ebx, eax
0x11157:    call    0xc970
0x1115c:    xor     eax, eax
0x1115e:    and     ebx, 0xffffffff
0x11161:    call    0xd870
0x11166:    xor     eax, eax
0x11168:    call    0xd740
0x1116d:    xor     eax, eax
0x1116f:    call    0xa820
0x11174:    xor     eax, eax
0x11176:    cmp     bl, 0x4e
0x11179:    setne   al
0x1117c:    add     rsp, 8
0x11180:    pop     rbx

```

```

0x11181:    pop     rbp
0x11182:    ret
0x11183:    nop     word ptr cs:[rax + rax]
0x1118e:    nop
0x11190:    endbr64
0x11194:    push    rbp
0x11195:    mov     rbp, rsi
0x11198:    sub     rsp, 0x130
0x1119f:    mov     rax, qword ptr fs:[0x28]
0x111a8:    mov     qword ptr [rsp + 0x128], rax
0x111b0:    xor     eax, eax
0x111b2:    mov     rsi, rsp
0x111b5:    call    0x18f40
0x111ba:    test    eax, eax
0x111bc:    je      0x111e0
0x111be:    xor     eax, eax
0x111c0:    mov     rdx, qword ptr [rsp + 0x128]
0x111c8:    sub     rdx, qword ptr fs:[0x28]
0x111d1:    jne     0x11217
0x111d3:    add     rsp, 0x130
0x111da:    pop     rbp
0x111db:    ret
0x111dc:    nop     dword ptr [rax]
0x111e0:    lea     rsi, [rsp + 0x90]
0x111e8:    mov     rdi, rbp
0x111eb:    call    0x18f40
0x111f0:    test    eax, eax
0x111f2:    jne     0x111be
0x111f4:    mov     rax, qword ptr [rsp + 0x90]
0x111fc:    cmp     qword ptr [rsp], rax
0x11200:    jne     0x111be
0x11202:    mov     rax, qword ptr [rsp + 0x98]
0x1120a:    cmp     qword ptr [rsp + 8], rax
0x1120f:    sete    al
0x11212:    movzx   eax, al
0x11215:    jmp     0x111c0
0x11217:    call    0x49d0
0x1121c:    nop     dword ptr [rax]
0x11220:    endbr64
0x11224:    push    rbx
0x11225:    mov     edx, dword ptr [rdi + 0x70]
0x11228:    mov     rbx, rdi
0x1122b:    mov     ecx, 0x1e
0x11230:    test    edx, edx
0x11232:    je      0x11268
0x11234:    lea     rdx, [rip + 0xffd8]
0x1123b:    lea     rdi, [rip + 0x107de]
0x11242:    xor     eax, eax

```

```

0x11244:      call    0xf760
0x11249:      xor     r8d, r8d
0x1124c:      cmp     al, 0x6f
0x1124e:      je      0x1125e
0x11250:      cmp     al, 0x73
0x11252:      je      0x11280
0x11254:      mov     r8d, 2
0x1125a:      cmp     al, 0x61
0x1125c:      je      0x11298
0x1125e:      mov     eax, r8d
0x11261:      pop     rbx
0x11262:      ret
0x11263:      nop     dword ptr [rax + rax]
0x11268:      lea     rdx, [rip + 0xffa9]
0x1126f:      lea     rdi, [rip + 0x107da]
0x11276:      xor     eax, eax
0x11278:      call    0xf760
0x1127d:      jmp     0x11249
0x1127f:      nop
0x11280:      cmp     dword ptr [rbx + 0x70], 1
0x11284:      mov     r8d, 2
0x1128a:      pop     rbx
0x1128b:      sbb     r8d, -1
0x1128f:      mov     eax, r8d
0x11292:      ret
0x11293:      nop     dword ptr [rax + rax]
0x11298:      mov     eax, dword ptr [rbx + 0x70]
0x1129b:      test    eax, eax
0x1129d:      je      0x1125e
0x1129f:      mov     dword ptr [rbx + 0x90], 0
0x112a9:      xor     r8d, r8d
0x112ac:      pop     rbx
0x112ad:      mov     eax, r8d
0x112b0:      ret
0x112b1:      nop     word ptr cs:[rax + rax]
0x112bc:      nop     dword ptr [rax]
0x112c0:      endbr64
0x112c4:      push    rbp
0x112c5:      mov     rbp, rdi
0x112c8:      sub     rsp, 0xa0
0x112cf:      mov     rax, qword ptr fs:[0x28]
0x112d8:      mov     qword ptr [rsp + 0x98], rax
0x112e0:      xor     eax, eax
0x112e2:      mov     rsi, rsp
0x112e5:      call    0x18f40
0x112ea:      test    eax, eax
0x112ec:      jne     0x112fe
0x112ee:      mov     eax, dword ptr [rsp + 0x18]

```

```

0x112f2:    and     eax, 0xf000
0x112f7:    cmp     eax, 0x8000
0x112fc:    je      0x11320
0x112fe:    mov     rax, qword ptr [rsp + 0x98]
0x11306:    sub     rax, qword ptr fs:[0x28]
0x1130f:    jne     0x1132a
0x11311:    add     rsp, 0xa0
0x11318:    pop     rbp
0x11319:    ret
0x1131a:    nop     word ptr [rax + rax]
0x11320:    mov     rdi, rbp
0x11323:    call    0x4870
0x11328:    jmp     0x112fe
0x1132a:    call    0x49d0
0x1132f:    nop
0x11330:    endbr64
0x11334:    push    r15
0x11336:    push    r14
0x11338:    mov     r14, rdx
0x1133b:    push    r13
0x1133d:    push    r12
0x1133f:    mov     r12d, ecx
0x11342:    push    rbp
0x11343:    and     r12d, 0xf000
0x1134a:    mov     rbp, rdi
0x1134d:    push    rbx
0x1134e:    mov     ebx, ecx
0x11350:    sub     rsp, 0x178
0x11357:    mov     qword ptr [rsp + 8], rsi
0x1135c:    mov     dword ptr [rsp + 0x30], r8d
0x11361:    mov     dword ptr [rsp + 0x34], r9d
0x11366:    mov     rax, qword ptr fs:[0x28]
0x1136f:    mov     qword ptr [rsp + 0x168], rax
0x11377:    xor     eax, eax
0x11379:    cmp     r12d, 0xa000
0x11380:    je      0x11a10
0x11386:    cmp     r12d, 0x4000
0x1138d:    je      0x11828
0x11393:    mov     rdi, r14
0x11396:    call    0x49b0
0x1139b:    mov     rdi, r14
0x1139e:    mov     r13, rax
0x113a1:    call    0x1d7a0
0x113a6:    lea     rsi, [rsp + 0x40]
0x113ab:    mov     rdi, rax
0x113ae:    mov     qword ptr [rsp + 0x18], rax
0x113b3:    call    0x18f40
0x113b8:    test    eax, eax

```

```

0x113ba:    je      0x11630
0x113c0:    mov     rdi, qword ptr [rsp + 0x18]
0x113c5:    call    0x190c0
0x113ca:    cmp     byte ptr [rax], 0
0x113cd:    mov     r13, rax
0x113d0:    je      0x11ae5
0x113d6:    mov     esi, dword ptr [rbp + 0x90]
0x113dc:    test    esi, esi
0x113de:    je      0x113f4
0x113e0:    mov     rdi, qword ptr [rsp + 0x18]
0x113e5:    xor     esi, esi
0x113e7:    call    0x4d90
0x113ec:    test    eax, eax
0x113ee:    je      0x11a60
0x113f4:    mov     rdi, qword ptr [rsp + 8]
0x113f9:    call    0x49b0
0x113fe:    lea     rdi, [rax + 0x20]
0x11402:    call    0x1d670
0x11407:    mov     r8, qword ptr [rsp + 8]
0x1140c:    lea     rcx, [rip + 0xfe34]
0x11413:    mov     r13, rax
0x11416:    cmp     r12d, 0x8000
0x1141d:    je      0x11a48
0x11423:    mov     rdx, -1
0x1142a:    mov     esi, 1
0x1142f:    mov     rdi, r13
0x11432:    xor     eax, eax
0x11434:    call    0x4f10
0x11439:    mov     edx, 1
0x1143e:    mov     esi, 1
0x11443:    mov     rdi, r13
0x11446:    call    0xe480
0x1144b:    xor     eax, eax
0x1144d:    call    0xa820
0x11452:    mov     rdi, r13
0x11455:    call    0x18d10
0x1145a:    mov     rdi, qword ptr [rsp + 8]
0x1145f:    xor     esi, esi
0x11461:    xor     eax, eax
0x11463:    call    0x4d30
0x11468:    mov     r15d, eax
0x1146b:    cmp     eax, -1
0x1146e:    je      0x11a90
0x11474:    mov     ecx, dword ptr [rbp + 0x10]
0x11477:    test    ecx, ecx
0x11479:    je      0x1147e
0x1147b:    and     ebx, 0xffffffffb6
0x1147e:    mov     rdi, qword ptr [rsp + 0x18]

```

```

0x11483:    mov     esi, ebx
0x11485:    call   0x4e00
0x1148a:    mov     edi, r15d
0x1148d:    mov     dword ptr [rsp + 0x28], eax
0x11491:    cmp     eax, -1
0x11494:    je      0x11ae0
0x1149a:    call   0x182f0
0x1149f:    mov     qword ptr [rsp + 0x10], 0x8000
0x114a8:    mov     r13, rax
0x114ab:    cmp     rax, 0x8000
0x114b1:    jle     0x11b38
0x114b7:    mov     eax, ebx
0x114b9:    and     ebx, 0xb000
0x114bf:    and     eax, 0xf000
0x114c4:    mov     dword ptr [rsp + 0x2c], eax
0x114c8:    cmp     ebx, 0x2000
0x114ce:    je      0x11b00
0x114d4:    test    r13, r13
0x114d7:    je      0x11c58
0x114dd:    mov     rdi, qword ptr [rsp + 0x10]
0x114e2:    call   0x1d670
0x114e7:    mov     r14, rax
0x114ea:    test    r13, r13
0x114ed:    jle     0x11aa0
0x114f3:    movabs  rdx, 0xa3d70a3d70a3d70b
0x114fd:    mov     rax, r13
0x11500:    imul    rdx
0x11503:    mov     rax, r13
0x11506:    sar     rax, 0x3f
0x1150a:    add     rdx, r13
0x1150d:    sar     rdx, 6
0x11511:    sub     rdx, rax
0x11514:    mov     qword ptr [rsp + 0x38], rdx
0x11519:    xor     r12d, r12d
0x1151c:    jmp     0x11530
0x1151e:    nop
0x11520:    add     r12, 0x8000
0x11527:    cmp     r12, r13
0x1152a:    jge     0x11aa0
0x11530:    mov     rax, qword ptr [rsp + 0x10]
0x11535:    mov     rbp, r13
0x11538:    mov     edx, dword ptr [rip + 0x1a572]
0x1153e:    sub     rbp, r12
0x11541:    cmp     rbp, rax
0x11544:    cmovg   rbp, rax
0x11548:    test    edx, edx
0x1154a:    jne     0x11690
0x11550:    mov     edi, 1

```

```

0x11555:      call    0x18b70
0x1155a:      mov     edi, r15d
0x1155d:      movsxd  rdx, ebp
0x11560:      mov     rsi, r14
0x11563:      call    0x18d30
0x11568:      xor     edi, edi
0x1156a:      mov     ebx, eax
0x1156c:      call    0x18b70
0x11571:      cmp     ebp, ebx
0x11573:      je      0x11583
0x11575:      test    ebx, ebx
0x11577:      js      0x11bf0
0x1157d:      je      0x11aa0
0x11583:      mov     eax, dword ptr [rip + 0x1a527]
0x11589:      test    eax, eax
0x1158b:      jne     0x11778
0x11591:      movsxd  rbp, ebx
0x11594:      mov     edi, dword ptr [rsp + 0x28]
0x11598:      mov     rsi, r14
0x1159b:      mov     rdx, rbp
0x1159e:      call    0x18da0
0x115a3:      cmp     ebx, eax
0x115a5:      jne     0x11b50
0x115ab:      add     rbp, r12
0x115ae:      cmp     rbp, r13
0x115b1:      jg      0x11520
0x115b7:      mov     rdi, qword ptr [rsp + 8]
0x115bc:      call    0x49b0
0x115c1:      lea     rdi, [rax + 0x20]
0x115c5:      call    0x1d670
0x115ca:      cmp     dword ptr [rsp + 0x2c], 0x8000
0x115d2:      mov     rbx, rax
0x115d5:      je      0x117e0
0x115db:      mov     r9, qword ptr [rsp + 8]
0x115e0:      mov     rdi, rax
0x115e3:      mov     r8, rbp
0x115e6:      mov     esi, 1
0x115eb:      lea     rcx, [rip + 0xfc7b]
0x115f2:      mov     rdx, -1
0x115f9:      xor     eax, eax
0x115fb:      call    0x4f10
0x11600:      mov     rdi, rbx
0x11603:      mov     edx, 1
0x11608:      mov     esi, 1
0x1160d:      call    0xe480
0x11612:      xor     eax, eax
0x11614:      call    0xa820
0x11619:      mov     rdi, rbx

```



```

0x1161c:    call    0x18d10
0x11621:    jmp     0x11520
0x11626:    nop     word ptr cs:[rax + rax]
0x11630:    mov     eax, dword ptr [rsp + 0x58]
0x11634:    and     eax, 0xf000
0x11639:    cmp     eax, 0x4000
0x1163e:    jne     0x113c0
0x11644:    mov     r14, qword ptr [rsp + 8]
0x11649:    mov     rdi, r14
0x1164c:    call    0x49b0
0x11651:    mov     rdi, qword ptr [rsp + 0x18]
0x11656:    lea     rsi, [r13 + rax + 2]
0x1165b:    call    0x1d690
0x11660:    mov     rdi, rax
0x11663:    mov     r15, rax
0x11666:    mov     qword ptr [rsp + 0x18], rax
0x1166b:    call    0x49b0
0x11670:    mov     rsi, r14
0x11673:    mov     byte ptr [r15 + rax], 0x2f
0x11678:    lea     rdi, [r15 + rax + 1]
0x1167d:    call    0x48b0
0x11682:    jmp     0x113c0
0x11687:    nop     word ptr [rax + rax]
0x11690:    mov     dword ptr [rip + 0x1a416], 0
0x1169a:    xor     eax, eax
0x1169c:    call    0xc8c0
0x116a1:    mov     edx, 4
0x116a6:    lea     rsi, [rip + 0xee85]
0x116ad:    lea     rdi, [rip + 0xfb45]
0x116b4:    mov     qword ptr [rsp + 0x20], rax
0x116b9:    call    0x8070
0x116be:    mov     rdi, qword ptr [rsp + 0x20]
0x116c3:    mov     ebx, eax
0x116c5:    call    0xc970
0x116ca:    xor     eax, eax
0x116cc:    and     ebx, 0xffffffff
0x116cf:    call    0xd870
0x116d4:    xor     eax, eax
0x116d6:    call    0xd740
0x116db:    xor     eax, eax
0x116dd:    call    0xa820
0x116e2:    cmp     bl, 0x4e
0x116e5:    je      0x11550
0x116eb:    mov     edi, r15d
0x116ee:    call    0x4ab0
0x116f3:    mov     edi, dword ptr [rsp + 0x28]
0x116f7:    call    0x4ab0
0x116fc:    mov     rdi, qword ptr [rsp + 0x18]

```

```

0x11701:    lea     rsi, [rsp + 0xd0]
0x11709:    call    0x18f40
0x1170e:    test    eax, eax
0x11710:    jne     0x11729
0x11712:    mov     eax, dword ptr [rsp + 0xe8]
0x11719:    and     eax, 0xf000
0x1171e:    cmp     eax, 0x8000
0x11723:    je      0x11ced
0x11729:    mov     rdi, r14
0x1172c:    mov     r13d, 2
0x11732:    call    0x18d10
0x11737:    mov     rdi, qword ptr [rsp + 0x18]
0x1173c:    call    0x18d10
0x11741:    nop     dword ptr [rax]
0x11748:    mov     rax, qword ptr [rsp + 0x168]
0x11750:    sub     rax, qword ptr fs:[0x28]
0x11759:    jne     0x11cfc
0x1175f:    add     rsp, 0x178
0x11766:    mov     eax, r13d
0x11769:    pop     rbx
0x1176a:    pop     rbp
0x1176b:    pop     r12
0x1176d:    pop     r13
0x1176f:    pop     r14
0x11771:    pop     r15
0x11773:    ret
0x11774:    nop     dword ptr [rax]
0x11778:    mov     dword ptr [rip + 0x1a32e], 0
0x11782:    xor     eax, eax
0x11784:    call    0xc8c0
0x11789:    mov     edx, 4
0x1178e:    lea     rsi, [rip + 0xed9d]
0x11795:    lea     rdi, [rip + 0xfa5d]
0x1179c:    mov     qword ptr [rsp + 0x20], rax
0x117a1:    call    0x8070
0x117a6:    mov     rdi, qword ptr [rsp + 0x20]
0x117ab:    mov     ebp, eax
0x117ad:    call    0xc970
0x117b2:    xor     eax, eax
0x117b4:    and     ebp, 0xfffffffff
0x117b7:    call    0xd870
0x117bc:    xor     eax, eax
0x117be:    call    0xd740
0x117c3:    xor     eax, eax
0x117c5:    call    0xa820
0x117ca:    cmp     bpl, 0x4e
0x117ce:    je      0x11591
0x117d4:    jmp     0x116eb

```

```

0x117d9:      nop      dword ptr [rax]
0x117e0:      cmp      r13, 0x1869f
0x117e7:      jle      0x119f0
0x117ed:      mov      rax, rbp
0x117f0:      cqo
0x117f2:      idiv     qword ptr [rsp + 0x38]
0x117f7:      mov      r8, rax
0x117fa:      mov      r9, qword ptr [rsp + 8]
0x117ff:      mov      esi, 1
0x11804:      mov      rdi, rbx
0x11807:      xor      eax, eax
0x11809:      lea      rcx, [rip + 0xfa4b]
0x11810:      mov      rdx, -1
0x11817:      call     0x4f10
0x1181c:      jmp      0x11600
0x11821:      nop      dword ptr [rax]
0x11828:      lea      rsi, [rsp + 0x40]
0x1182d:      mov      rdi, r14
0x11830:      call     0x18f40
0x11835:      test     eax, eax
0x11837:      jne      0x118c3
0x1183d:      mov      eax, dword ptr [rsp + 0x58]
0x11841:      mov      r13d, 0xb
0x11847:      and      eax, 0xf000
0x1184c:      cmp      eax, 0x4000
0x11851:      jne      0x11748
0x11857:      mov      rdi, r14
0x1185a:      call     0x49b0
0x1185f:      mov      r15, qword ptr [rsp + 8]
0x11864:      mov      rbx, rax
0x11867:      mov      rdi, r15
0x1186a:      call     0x49b0
0x1186f:      lea      rdi, [rbx + rax + 2]
0x11874:      call     0x1d670
0x11879:      mov      r9, r15
0x1187c:      mov      r8, r14
0x1187f:      mov      esi, 1
0x11884:      mov      rdi, rax
0x11887:      mov      r12, rax
0x1188a:      xor      eax, eax
0x1188c:      mov      rdx, -1
0x11893:      lea      rcx, [rip + 0xee67]
0x1189a:      call     0x4f10
0x1189f:      mov      edi, dword ptr [rbp + 0x90]
0x118a5:      test     edi, edi
0x118a7:      je       0x118bb
0x118a9:      xor      esi, esi
0x118ab:      mov      rdi, r12

```

```

0x118ae:    call    0x4d90
0x118b3:    test    eax, eax
0x118b5:    je      0x11cbb
0x118bb:    mov     rdi, r12
0x118be:    call    0x18d10
0x118c3:    mov     r15, qword ptr [rsp + 8]
0x118c8:    mov     r13d, 6
0x118ce:    mov     rdi, r15
0x118d1:    call    0x49b0
0x118d6:    mov     rdi, r14
0x118d9:    mov     rbx, rax
0x118dc:    call    0x49b0
0x118e1:    lea     rdi, [rbx + rax + 0x21]
0x118e6:    call    0x1d670
0x118eb:    mov     rdi, r14
0x118ee:    mov     r12, rax
0x118f1:    call    0x190c0
0x118f6:    cmp     byte ptr [rax], 0
0x118f9:    je      0x11748
0x118ff:    mov     rdi, r15
0x11902:    call    0x49b0
0x11907:    mov     rdi, r14
0x1190a:    mov     rbx, rax
0x1190d:    call    0x49b0
0x11912:    lea     rdi, [rbx + rax + 0x20]
0x11917:    call    0x1d670
0x1191c:    mov     r9, r14
0x1191f:    mov     r8, r15
0x11922:    mov     esi, 1
0x11927:    lea     rcx, [rip + 0xf8ed]
0x1192e:    mov     r13, rax
0x11931:    mov     rdi, rax
0x11934:    xor     eax, eax
0x11936:    mov     rdx, -1
0x1193d:    call    0x4f10
0x11942:    mov     edx, 1
0x11947:    mov     esi, 1
0x1194c:    mov     rdi, r13
0x1194f:    call    0xe480
0x11954:    xor     eax, eax
0x11956:    call    0xa820
0x1195b:    mov     rdi, r13
0x1195e:    mov     r13d, 0xc
0x11964:    call    0x18d10
0x11969:    mov     r9, r14
0x1196c:    mov     r8, r15
0x1196f:    mov     esi, 1
0x11974:    lea     rcx, [rip + 0xf8a7]

```

```

0x1197b:    mov     rdx, -1
0x11982:    mov     rdi, r12
0x11985:    xor     eax, eax
0x11987:    call    0x4f10
0x1198c:    mov     esi, 1
0x11991:    mov     rdi, r12
0x11994:    call    0x193f0
0x11999:    mov     rdi, r12
0x1199c:    mov     ebx, eax
0x1199e:    call    0x18d10
0x119a3:    mov     rdi, qword ptr [rbp + 0x18]
0x119a7:    call    0xc4b0
0x119ac:    mov     eax, ebx
0x119ae:    and     eax, 0x7f
0x119b1:    add     eax, 1
0x119b4:    cmp     al, 1
0x119b6:    jg      0x11748
0x119bc:    mov     r13d, 1
0x119c2:    test    bl, 0x7f
0x119c5:    jne     0x11748
0x119cb:    and     bh, 0xff
0x119ce:    je      0x11748
0x119d4:    lea     rdi, [rip + 0xf857]
0x119db:    mov     r13d, 9
0x119e1:    call    0x195e0
0x119e6:    jmp     0x11748
0x119eb:    nop     dword ptr [rax + rax]
0x119f0:    lea     rax, [rbp + rbp*4]
0x119f5:    lea     r8, [rax + rax*4]
0x119f9:    shl     r8, 2
0x119fd:    mov     rax, r8
0x11a00:    cqo
0x11a02:    idiv    r13
0x11a05:    mov     r8, rax
0x11a08:    jmp     0x117fa
0x11a0d:    nop     dword ptr [rax]
0x11a10:    mov     rdi, qword ptr [rsp + 8]
0x11a15:    lea     rsi, [rsp + 0xd0]
0x11a1d:    call    0x18f40
0x11a22:    cmp     eax, -1
0x11a25:    je      0x11393
0x11a2b:    mov     ebx, dword ptr [rsp + 0xe8]
0x11a32:    mov     r12d, ebx
0x11a35:    and     r12d, 0xf000
0x11a3c:    jmp     0x11386
0x11a41:    nop     dword ptr [rax]
0x11a48:    mov     r8, qword ptr [rsp + 8]
0x11a4d:    lea     rcx, [rip + 0xf7e1]

```

```

0x11a54:      jmp      0x11423
0x11a59:      nop      dword ptr [rax]
0x11a60:      call     0xe4e0
0x11a65:      mov      rsi, r13
0x11a68:      xor      eax, eax
0x11a6a:      mov      rdi, rbp
0x11a6d:      call     0x11220
0x11a72:      mov      r13d, eax
0x11a75:      xor      eax, eax
0x11a77:      call     0xa820
0x11a7c:      test     r13d, r13d
0x11a7f:      je       0x113f4
0x11a85:      jmp      0x11748
0x11a8a:      nop      word ptr [rax + rax]
0x11a90:      mov      r13d, 4
0x11a96:      jmp      0x11748
0x11a9b:      nop      dword ptr [rax + rax]
0x11aa0:      call     0x49e0
0x11aa5:      test     eax, eax
0x11aa7:      je       0x11c90
0x11aad:      mov      edi, r15d
0x11ab0:      mov      r13d, 1
0x11ab6:      call     0x4ab0
0x11abb:      mov      edi, dword ptr [rsp + 0x28]
0x11abf:      call     0x4ab0
0x11ac4:      mov      rdi, r14
0x11ac7:      call     0x18d10
0x11acc:      mov      rdi, qword ptr [rsp + 0x18]
0x11ad1:      call     0x18d10
0x11ad6:      jmp      0x11748
0x11adb:      nop      dword ptr [rax + rax]
0x11ae0:      call     0x4ab0
0x11ae5:      mov      rdi, qword ptr [rsp + 0x18]
0x11aea:      mov      r13d, 6
0x11af0:      call     0x18d10
0x11af5:      jmp      0x11748
0x11afa:      nop      word ptr [rax + rax]
0x11b00:      movabs   r13, 0x7fffffffffffffff
0x11b0a:      mov      edi, 0x8000
0x11b0f:      call     0x1d670
0x11b14:      mov      qword ptr [rsp + 0x10], 0x8000
0x11b1d:      mov      r14, rax
0x11b20:      movabs   rax, 0x147ae147ae147ae
0x11b2a:      mov      qword ptr [rsp + 0x38], rax
0x11b2f:      jmp      0x11519
0x11b34:      nop      dword ptr [rax]
0x11b38:      mov      edi, r15d
0x11b3b:      call     0x182f0

```

```

0x11b40:    mov     qword ptr [rsp + 0x10], rax
0x11b45:    mov     r13, rax
0x11b48:    jmp     0x114b7
0x11b4d:    nop     dword ptr [rax]
0x11b50:    mov     dword ptr [rsp + 8], eax
0x11b54:    call    0x4860
0x11b59:    mov     edi, r15d
0x11b5c:    mov     ebx, dword ptr [rax]
0x11b5e:    call    0x4ab0
0x11b63:    mov     edi, dword ptr [rsp + 0x28]
0x11b67:    call    0x4ab0
0x11b6c:    mov     rdi, qword ptr [rsp + 0x18]
0x11b71:    lea     rsi, [rsp + 0xd0]
0x11b79:    call    0x18f40
0x11b7e:    mov     edx, dword ptr [rsp + 8]
0x11b82:    test    eax, eax
0x11b84:    jne     0x11bb0
0x11b86:    mov     eax, dword ptr [rsp + 0xe8]
0x11b8d:    and     eax, 0xf000
0x11b92:    cmp     eax, 0x8000
0x11b97:    jne     0x11bb0
0x11b99:    mov     rdi, qword ptr [rsp + 0x18]
0x11b9e:    call    0x4870
0x11ba3:    mov     edx, dword ptr [rsp + 8]
0x11ba7:    nop     word ptr [rax + rax]
0x11bb0:    mov     rdi, r14
0x11bb3:    mov     dword ptr [rsp + 8], edx
0x11bb7:    call    0x18d10
0x11bbc:    mov     rdi, qword ptr [rsp + 0x18]
0x11bc1:    call    0x18d10
0x11bc6:    mov     edx, dword ptr [rsp + 8]
0x11bca:    cmp     ebx, 0x1c
0x11bcd:    sete    r13b
0x11bd1:    test    edx, edx
0x11bd3:    setg    al
0x11bd6:    or      r13d, eax
0x11bd9:    movzx   r13d, r13b
0x11bdd:    add     r13d, 7
0x11be1:    jmp     0x11748
0x11be6:    nop     word ptr cs:[rax + rax]
0x11bf0:    mov     edi, r15d
0x11bf3:    call    0x4ab0
0x11bf8:    mov     edi, dword ptr [rsp + 0x28]
0x11bfc:    call    0x4ab0
0x11c01:    mov     rdi, qword ptr [rsp + 0x18]
0x11c06:    lea     rsi, [rsp + 0xd0]
0x11c0e:    call    0x18f40
0x11c13:    test    eax, eax

```

```

0x11c15:    jne     0x11c38
0x11c17:    mov     eax, dword ptr [rsp + 0xe8]
0x11c1e:    and     eax, 0xf000
0x11c23:    cmp     eax, 0x8000
0x11c28:    jne     0x11c38
0x11c2a:    mov     rdi, qword ptr [rsp + 0x18]
0x11c2f:    call    0x4870
0x11c34:    nop     dword ptr [rax]
0x11c38:    mov     rdi, r14
0x11c3b:    mov     r13d, 5
0x11c41:    call    0x18d10
0x11c46:    mov     rdi, qword ptr [rsp + 0x18]
0x11c4b:    call    0x18d10
0x11c50:    jmp     0x11748
0x11c55:    nop     dword ptr [rax]
0x11c58:    call    0x49e0
0x11c5d:    test    eax, eax
0x11c5f:    je      0x11ca7
0x11c61:    mov     edi, r15d
0x11c64:    mov     r13d, 1
0x11c6a:    call    0x4ab0
0x11c6f:    mov     edi, dword ptr [rsp + 0x28]
0x11c73:    call    0x4ab0
0x11c78:    mov     rdi, qword ptr [rsp + 0x18]
0x11c7d:    call    0x18d10
0x11c82:    jmp     0x11748
0x11c87:    nop     word ptr [rax + rax]
0x11c90:    mov     edx, dword ptr [rsp + 0x34]
0x11c94:    mov     esi, dword ptr [rsp + 0x30]
0x11c98:    mov     rdi, qword ptr [rsp + 0x18]
0x11c9d:    call    0x4cc0
0x11ca2:    jmp     0x11aad
0x11ca7:    mov     edx, dword ptr [rsp + 0x34]
0x11cab:    mov     esi, dword ptr [rsp + 0x30]
0x11caf:    mov     rdi, qword ptr [rsp + 0x18]
0x11cb4:    call    0x4cc0
0x11cb9:    jmp     0x11c61
0x11cbb:    call    0xe4e0
0x11cc0:    mov     rdi, rbp
0x11cc3:    mov     rsi, r12
0x11cc6:    xor     eax, eax
0x11cc8:    call    0x11220
0x11ccd:    mov     r13d, eax
0x11cd0:    xor     eax, eax
0x11cd2:    call    0xa820
0x11cd7:    mov     rdi, r12
0x11cda:    call    0x18d10
0x11cdf:    test    r13d, r13d

```



```

0x11ce2:      je      0x118c3
0x11ce8:      jmp     0x11748
0x11ced:      mov     rdi, qword ptr [rsp + 0x18]
0x11cf2:      call    0x4870
0x11cf7:      jmp     0x11729
0x11cfc:      call    0x49d0
0x11d01:      nop     word ptr cs:[rax + rax]
0x11d0c:      nop     dword ptr [rax]
0x11d10:      endbr64
0x11d14:      push    r15
0x11d16:      and     ecx, 0xf000
0x11d1c:      push    r14
0x11d1e:      mov     r14, rdi
0x11d21:      push    r13
0x11d23:      push    r12
0x11d25:      mov     r12, rdx
0x11d28:      push    rbp
0x11d29:      mov     rbp, rsi
0x11d2c:      push    rbx
0x11d2d:      sub     rsp, 0x148
0x11d34:      mov     rax, qword ptr fs:[0x28]
0x11d3d:      mov     qword ptr [rsp + 0x138], rax
0x11d45:      xor     eax, eax
0x11d47:      cmp     ecx, 0x4000
0x11d4d:      je      0x11f20
0x11d53:      mov     rdi, rdx
0x11d56:      lea     rbx, [rsp + 0x10]
0x11d5b:      call    0x49b0
0x11d60:      mov     rdi, r12
0x11d63:      mov     r13, rax
0x11d66:      call    0x1d7a0
0x11d6b:      mov     rsi, rbx
0x11d6e:      mov     rdi, rax
0x11d71:      mov     r12, rax
0x11d74:      call    0x18f40
0x11d79:      test    eax, eax
0x11d7b:      je      0x11ed0
0x11d81:      mov     rdi, r12
0x11d84:      call    0x190c0
0x11d89:      cmp     byte ptr [rax], 0
0x11d8c:      mov     r15, rax
0x11d8f:      je      0x12170
0x11d95:      cmp     r12, rax
0x11d98:      je      0x120e8
0x11d9e:      mov     edx, dword ptr [r14 + 0x90]
0x11da5:      test    edx, edx
0x11da7:      je      0x11dbb
0x11da9:      xor     esi, esi

```

```

0x11dab:    mov     rdi, r12
0x11dae:    call   0x4d90
0x11db3:    test   eax, eax
0x11db5:    je      0x12140
0x11dbb:    mov     rdi, rbp
0x11dbe:    call   0x49b0
0x11dc3:    lea     rdi, [rax + 0x20]
0x11dc7:    call   0x1d670
0x11dcc:    mov     r8, rbp
0x11dcf:    mov     esi, 1
0x11dd4:    lea     rcx, [rip + 0xf4c2]
0x11ddb:    mov     r13, rax
0x11dde:    mov     rdi, rax
0x11de1:    mov     rdx, -1
0x11de8:    xor     eax, eax
0x11dea:    call   0x4f10
0x11def:    mov     edx, 1
0x11df4:    mov     esi, 1
0x11df9:    mov     rdi, r13
0x11dfc:    call   0xe480
0x11e01:    xor     eax, eax
0x11e03:    call   0xa820
0x11e08:    mov     rdi, r13
0x11e0b:    mov     r13d, 6
0x11e11:    call   0x18d10
0x11e16:    lea     rsi, [rsp + 0xa0]
0x11e1e:    mov     rdi, rbp
0x11e21:    call   0x18f40
0x11e26:    cmp     eax, -1
0x11e29:    je      0x11ea1
0x11e2b:    mov     rsi, rbx
0x11e2e:    mov     rdi, r12
0x11e31:    call   0x18f40
0x11e36:    cmp     eax, -1
0x11e39:    je      0x121c8
0x11e3f:    mov     rax, qword ptr [rsp + 0xa0]
0x11e47:    cmp     qword ptr [rsp + 0x10], rax
0x11e4c:    jne     0x11e5a
0x11e4e:    mov     eax, dword ptr [r14 + 0x10]
0x11e52:    test    eax, eax
0x11e54:    je      0x12188
0x11e5a:    mov     r9d, dword ptr [rsp + 0xc0]
0x11e62:    mov     rdx, r12
0x11e65:    mov     rsi, rbp
0x11e68:    xor     eax, eax
0x11e6a:    mov     r8d, dword ptr [rsp + 0xbc]
0x11e72:    mov     ecx, dword ptr [rsp + 0xb8]
0x11e79:    mov     rdi, r14

```

```

0x11e7c:      call    0x11330
0x11e81:      mov     r13d, eax
0x11e84:      cmp     eax, 3
0x11e87:      je      0x11ea1
0x11e89:      jg      0x120d0
0x11e8f:      cmp     eax, 1
0x11e92:      je      0x121a8
0x11e98:      cmp     eax, 2
0x11e9b:      jne     0x12210
0x11ea1:      mov     rax, qword ptr [rsp + 0x138]
0x11ea9:      sub     rax, qword ptr fs:[0x28]
0x11eb2:      jne     0x122c1
0x11eb8:      add     rsp, 0x148
0x11ebf:      mov     eax, r13d
0x11ec2:      pop     rbx
0x11ec3:      pop     rbp
0x11ec4:      pop     r12
0x11ec6:      pop     r13
0x11ec8:      pop     r14
0x11eca:      pop     r15
0x11ecc:      ret
0x11ecd:      nop     dword ptr [rax]
0x11ed0:      mov     eax, dword ptr [rsp + 0x28]
0x11ed4:      and     eax, 0xf000
0x11ed9:      cmp     eax, 0x4000
0x11ede:      jne     0x11d81
0x11ee4:      mov     rdi, rbp
0x11ee7:      call    0x49b0
0x11eec:      mov     rdi, r12
0x11eef:      lea     rsi, [r13 + rax + 2]
0x11ef4:      call    0x1d690
0x11ef9:      mov     rdi, rax
0x11efc:      mov     r12, rax
0x11eff:      call    0x49b0
0x11f04:      mov     rsi, rbp
0x11f07:      mov     byte ptr [r12 + rax], 0x2f
0x11f0c:      lea     rdi, [r12 + rax + 1]
0x11f11:      call    0x48b0
0x11f16:      jmp     0x11d81
0x11f1b:      nop     dword ptr [rax + rax]
0x11f20:      lea     rsi, [rsp + 0x10]
0x11f25:      mov     rdi, rdx
0x11f28:      call    0x18f40
0x11f2d:      test    eax, eax
0x11f2f:      jne     0x11fb7
0x11f35:      mov     eax, dword ptr [rsp + 0x28]
0x11f39:      mov     r13d, 9
0x11f3f:      and     eax, 0xf000

```

```

0x11f44:      cmp     eax, 0x4000
0x11f49:      jne     0x11ea1
0x11f4f:      mov     rdi, r12
0x11f52:      call    0x49b0
0x11f57:      mov     rdi, rbp
0x11f5a:      mov     rbx, rax
0x11f5d:      call    0x49b0
0x11f62:      lea     rdi, [rbx + rax + 2]
0x11f67:      call    0x1d670
0x11f6c:      mov     esi, 1
0x11f71:      mov     r9, rbp
0x11f74:      mov     r8, r12
0x11f77:      mov     rdi, rax
0x11f7a:      mov     r15, rax
0x11f7d:      xor     eax, eax
0x11f7f:      mov     rdx, -1
0x11f86:      lea     rcx, [rip + 0xe774]
0x11f8d:      call    0x4f10
0x11f92:      mov     esi, dword ptr [r14 + 0x90]
0x11f99:      test    esi, esi
0x11f9b:      je      0x11faf
0x11f9d:      xor     esi, esi
0x11f9f:      mov     rdi, r15
0x11fa2:      call    0x4d90
0x11fa7:      test    eax, eax
0x11fa9:      je      0x12278
0x11faf:      mov     rdi, r15
0x11fb2:      call    0x18d10
0x11fb7:      mov     rdi, rbp
0x11fba:      call    0x49b0
0x11fbf:      mov     rdi, r12
0x11fc2:      mov     rbx, rax
0x11fc5:      call    0x49b0
0x11fca:      lea     rdi, [rbx + rax + 0x21]
0x11fcf:      call    0x1d670
0x11fd4:      mov     rdi, r12
0x11fd7:      mov     r13, rax
0x11fda:      call    0x190c0
0x11fdf:      cmp     byte ptr [rax], 0
0x11fe2:      je      0x12178
0x11fe8:      mov     rdi, rbp
0x11feb:      call    0x49b0
0x11ff0:      mov     rdi, r12
0x11ff3:      mov     rbx, rax
0x11ff6:      call    0x49b0
0x11ffb:      lea     rdi, [rbx + rax + 0x20]
0x12000:      call    0x1d670
0x12005:      mov     r9, r12

```

```

0x12008:      mov     r8, rbp
0x1200b:      mov     esi, 1
0x12010:      lea     rcx, [rip + 0xf26c]
0x12017:      mov     r15, rax
0x1201a:      mov     rdi, rax
0x1201d:      xor     eax, eax
0x1201f:      mov     rdx, -1
0x12026:      call    0x4f10
0x1202b:      mov     edx, 1
0x12030:      mov     esi, 1
0x12035:      mov     rdi, r15
0x12038:      call    0xe480
0x1203d:      xor     eax, eax
0x1203f:      call    0xa820
0x12044:      mov     rdi, r15
0x12047:      call    0x18d10
0x1204c:      mov     r9, r12
0x1204f:      mov     r8, rbp
0x12052:      mov     rdi, r13
0x12055:      lea     rcx, [rip + 0xf22e]
0x1205c:      mov     rdx, -1
0x12063:      mov     esi, 1
0x12068:      xor     eax, eax
0x1206a:      call    0x4f10
0x1206f:      mov     esi, 1
0x12074:      mov     rdi, r13
0x12077:      call    0x193f0
0x1207c:      mov     rdi, r13
0x1207f:      mov     r13d, 0xc
0x12085:      mov     ebx, eax
0x12087:      call    0x18d10
0x1208c:      mov     rdi, qword ptr [r14 + 0x18]
0x12090:      call    0xc4b0
0x12095:      mov     eax, ebx
0x12097:      and     eax, 0x7f
0x1209a:      add     eax, 1
0x1209d:      cmp     al, 1
0x1209f:      jg      0x11ea1
0x120a5:      mov     r13d, 8
0x120ab:      test    bl, 0x7f
0x120ae:      jne     0x11ea1
0x120b4:      and     bh, 0xff
0x120b7:      jne     0x122b0
0x120bd:      mov     r13d, 1
0x120c3:      jmp     0x11ea1
0x120c8:      nop     dword ptr [rax + rax]
0x120d0:      cmp     eax, 8
0x120d3:      jne     0x12210

```

```

0x120d9:      mov     r13d, 0xa
0x120df:      jmp     0x11ea1
0x120e4:      nop     dword ptr [rax]
0x120e8:      mov     rdi, r12
0x120eb:      call    0x49b0
0x120f0:      lea     rdi, [rax + 3]
0x120f4:      mov     r13, rax
0x120f7:      call    0x1d670
0x120fc:      mov     ecx, 0x2f2e
0x12101:      lea     rdx, [r13 + 1]
0x12105:      mov     rsi, r15
0x12108:      mov     word ptr [rax], cx
0x1210b:      lea     rcx, [rax + 2]
0x1210f:      mov     r12, rax
0x12112:      mov     rdi, rcx
0x12115:      mov     qword ptr [rsp + 8], rcx
0x1211a:      call    0x4bb0
0x1211f:      mov     rdi, r15
0x12122:      call    0x18d10
0x12127:      mov     edx, dword ptr [r14 + 0x90]
0x1212e:      mov     r15, qword ptr [rsp + 8]
0x12133:      test    edx, edx
0x12135:      jne     0x11da9
0x1213b:      jmp     0x11dbb
0x12140:      call    0xe4e0
0x12145:      xor     eax, eax
0x12147:      mov     rsi, r15
0x1214a:      mov     rdi, r14
0x1214d:      call    0x11220
0x12152:      mov     r13d, eax
0x12155:      xor     eax, eax
0x12157:      call    0xa820
0x1215c:      test    r13d, r13d
0x1215f:      je      0x11dbb
0x12165:      jmp     0x11ea1
0x1216a:      nop     word ptr [rax + rax]
0x12170:      mov     rdi, r12
0x12173:      call    0x18d10
0x12178:      mov     r13d, 4
0x1217e:      jmp     0x11ea1
0x12183:      nop     dword ptr [rax + rax]
0x12188:      xor     eax, eax
0x1218a:      call    0x18ab0
0x1218f:      mov     eax, dword ptr [rsp + 0x28]
0x12193:      and     eax, 0xf000
0x12198:      sub     eax, 0x4000
0x1219d:      test    eax, 0xffffb000
0x121a2:      je      0x12250

```

```

0x121a8:      mov     rdi, rbp
0x121ab:      mov     r13d, 5
0x121b1:      call    0x4870
0x121b6:      cmp     eax, -1
0x121b9:      jne     0x120bd
0x121bf:      jmp     0x11ea1
0x121c4:      nop     dword ptr [rax]
0x121c8:      movzx   r13d, byte ptr [r15 - 1]
0x121cd:      mov     byte ptr [r15 - 1], 0
0x121d2:      mov     rsi, rbx
0x121d5:      cmp     byte ptr [r12], 0
0x121da:      je      0x12200
0x121dc:      mov     rdi, r12
0x121df:      call    0x18f40
0x121e4:      mov     byte ptr [r15 - 1], r13b
0x121e8:      cmp     eax, -1
0x121eb:      jne     0x11e3f
0x121f1:      mov     r13d, 7
0x121f7:      jmp     0x11ea1
0x121fc:      nop     dword ptr [rax]
0x12200:      lea     rdi, [rip + 0xf0a0]
0x12207:      call    0x18f40
0x1220c:      jmp     0x121e4
0x1220e:      nop
0x12210:      sub     r13d, 1
0x12214:      lea     rax, [rip + 0x18205]
0x1221b:      xor     ecx, ecx
0x1221d:      mov     rsi, rbp
0x12220:      movsxd  r13, r13d
0x12223:      mov     r8d, 0xe
0x12229:      lea     rdi, [rip + 0xf079]
0x12230:      mov     rdx, qword ptr [rax + r13*8]
0x12234:      xor     eax, eax
0x12236:      mov     r13d, 0xb
0x1223c:      call    0xf7e0
0x12241:      jmp     0x11ea1
0x12246:      nop     word ptr cs:[rax + rax]
0x12250:      mov     rdi, r12
0x12253:      call    0x4870
0x12258:      mov     rsi, r12
0x1225b:      mov     rdi, rbp
0x1225e:      call    0x4e10
0x12263:      cmp     eax, -1
0x12266:      jne     0x121a8
0x1226c:      jmp     0x12178
0x12271:      nop     dword ptr [rax]
0x12278:      call    0xe4e0
0x1227d:      mov     rdi, r14

```

```

0x12280:    mov     rsi, r15
0x12283:    xor     eax, eax
0x12285:    call    0x11220
0x1228a:    mov     r13d, eax
0x1228d:    xor     eax, eax
0x1228f:    call    0xa820
0x12294:    mov     rdi, r15
0x12297:    call    0x18d10
0x1229c:    test    r13d, r13d
0x1229f:    je      0x11fb7
0x122a5:    jmp     0x11ea1
0x122aa:    nop     word ptr [rax + rax]
0x122b0:    lea     rdi, [rip + 0xefe3]
0x122b7:    call    0x195e0
0x122bc:    jmp     0x11ea1
0x122c1:    call    0x49d0
0x122c6:    nop     word ptr cs:[rax + rax]
0x122d0:    endbr64
0x122d4:    push    r15
0x122d6:    push    r14
0x122d8:    push    r13
0x122da:    mov     r13, rsi
0x122dd:    push    r12
0x122df:    mov     r12, rdi
0x122e2:    mov     rdi, rsi
0x122e5:    push    rbp
0x122e6:    push    rbx
0x122e7:    sub     rsp, 0x18
0x122eb:    call    0x49b0
0x122f0:    lea     rdi, [rax + 1]
0x122f4:    mov     rbx, rax
0x122f7:    call    0x1d670
0x122fc:    movsxd  r14, dword ptr [r12 + 0x30]
0x12301:    mov     rsi, r13
0x12304:    mov     rbp, rax
0x12307:    mov     rdi, rax
0x1230a:    cmp     r14, rbx
0x1230d:    cmova   r14, rbx
0x12311:    mov     rdx, r14
0x12314:    call    0x4880
0x12319:    mov     byte ptr [rbp + r14], 0
0x1231f:    mov     ebx, dword ptr [r12 + 0x68]
0x12324:    test    ebx, ebx
0x12326:    jle     0x12450
0x1232c:    mov     r15, qword ptr [r12 + 0x38]
0x12331:    xor     r12d, r12d
0x12334:    mov     qword ptr [rsp + 8], r15
0x12339:    mov     r13, r15

```



```

0x1233c:      jmp      0x1234d
0x1233e:      nop
0x12340:      add      r12d, 1
0x12344:      add      r15, 0x50
0x12348:      cmp      ebx, r12d
0x1234b:      je       0x12380
0x1234d:      mov      rsi, qword ptr [r15]
0x12350:      mov      rdi, rbp
0x12353:      call     0x4b30
0x12358:      test     eax, eax
0x1235a:      jne      0x12340
0x1235c:      mov      rdi, rbp
0x1235f:      call     0x18d10
0x12364:      add      rsp, 0x18
0x12368:      mov      eax, r12d
0x1236b:      pop      rbx
0x1236c:      pop      rbp
0x1236d:      pop      r12
0x1236f:      pop      r13
0x12371:      pop      r14
0x12373:      pop      r15
0x12375:      ret
0x12376:      nop      word ptr cs:[rax + rax]
0x12380:      xor      r12d, r12d
0x12383:      jmp      0x12395
0x12385:      nop      dword ptr [rax]
0x12388:      add      r12d, 1
0x1238c:      add      r13, 0x50
0x12390:      cmp      ebx, r12d
0x12393:      je       0x123b0
0x12395:      mov      rsi, qword ptr [r13]
0x12399:      mov      rdi, rbp
0x1239c:      call     0x4830
0x123a1:      test     eax, eax
0x123a3:      jne      0x12388
0x123a5:      cmp      ebx, r12d
0x123a8:      jne      0x1235c
0x123aa:      nop      word ptr [rax + rax]
0x123b0:      mov      r15, qword ptr [rsp + 8]
0x123b5:      xor      r12d, r12d
0x123b8:      jmp      0x123cd
0x123ba:      nop      word ptr [rax + rax]
0x123c0:      add      r12d, 1
0x123c4:      add      r15, 0x50
0x123c8:      cmp      r12d, ebx
0x123cb:      je       0x12400
0x123cd:      mov      r13, qword ptr [r15]
0x123d0:      mov      rdi, r13

```

```

0x123d3:      call    0x49b0
0x123d8:      mov     rsi, r13
0x123db:      mov     rdi, rbp
0x123de:      cmp     rax, r14
0x123e1:      cmova   rax, r14
0x123e5:      mov     rdx, rax
0x123e8:      call    0x4890
0x123ed:      test    eax, eax
0x123ef:      jne     0x123c0
0x123f1:      cmp     r12d, ebx
0x123f4:      jne     0x1235c
0x123fa:      nop     word ptr [rax + rax]
0x12400:      xor     r12d, r12d
0x12403:      jmp     0x1241a
0x12405:      nop     dword ptr [rax]
0x12408:      add     r15, 0x50
0x1240c:      add     r12d, 1
0x12410:      mov     qword ptr [rsp + 8], r15
0x12415:      cmp     r12d, ebx
0x12418:      je      0x12450
0x1241a:      mov     r15, qword ptr [rsp + 8]
0x1241f:      mov     r13, qword ptr [r15]
0x12422:      mov     rdi, r13
0x12425:      call    0x49b0
0x1242a:      mov     rsi, r13
0x1242d:      mov     rdi, rbp
0x12430:      cmp     rax, r14
0x12433:      cmova   rax, r14
0x12437:      mov     rdx, rax
0x1243a:      call    0x4c30
0x1243f:      test    eax, eax
0x12441:      jne     0x12408
0x12443:      cmp     r12d, ebx
0x12446:      jne     0x1235c
0x1244c:      nop     dword ptr [rax]
0x12450:      xor     r12d, r12d
0x12453:      jmp     0x1235c
0x12458:      nop     dword ptr [rax + rax]
0x12460:      endbr64
0x12464:      sub     rsp, 0xa8
0x1246b:      mov     rax, qword ptr fs:[0x28]
0x12474:      mov     qword ptr [rsp + 0x98], rax
0x1247c:      xor     eax, eax
0x1247e:      mov     rsi, rsp
0x12481:      call    0x4b90
0x12486:      mov     r8d, eax
0x12489:      xor     eax, eax
0x1248b:      test    r8d, r8d

```

```

0x1248e:      js      0x124a4
0x12490:      mov     eax, dword ptr [rsp + 0x18]
0x12494:      and     eax, 0xf000
0x12499:      cmp     eax, 0x4000
0x1249e:      sete    al
0x124a1:      movzx   eax, al
0x124a4:      mov     rdx, qword ptr [rsp + 0x98]
0x124ac:      sub     rdx, qword ptr fs:[0x28]
0x124b5:      jne     0x124bf
0x124b7:      add     rsp, 0xa8
0x124be:      ret
0x124bf:      call    0x49d0
0x124c4:      nop     word ptr cs:[rax + rax]
0x124cf:      nop
0x124d0:      endbr64
0x124d4:      push    r13
0x124d6:      push    r12
0x124d8:      mov     r12d, esi
0x124db:      push    rbp
0x124dc:      push    rbx
0x124dd:      sub     rsp, 0xa8
0x124e4:      mov     rax, qword ptr fs:[0x28]
0x124ed:      mov     qword ptr [rsp + 0x98], rax
0x124f5:      xor     eax, eax
0x124f7:      mov     r13, rsp
0x124fa:      call    0x1d7a0
0x124ff:      cmp     byte ptr [rax], 0
0x12502:      mov     rbp, rax
0x12505:      mov     rbx, rax
0x12508:      je      0x12560
0x1250a:      nop     word ptr [rax + rax]
0x12510:      movzx   eax, byte ptr [rbx + 1]
0x12514:      add     rbx, 1
0x12518:      test    al, al
0x1251a:      je      0x12560
0x1251c:      cmp     rbp, rbx
0x1251f:      je      0x12510
0x12521:      cmp     al, 0x2f
0x12523:      jne     0x12510
0x12525:      mov     byte ptr [rbx], 0
0x12528:      mov     rsi, r13
0x1252b:      mov     rdi, rbp
0x1252e:      call    0x4b90
0x12533:      test    eax, eax
0x12535:      js      0x12590
0x12537:      mov     eax, dword ptr [rsp + 0x18]
0x1253b:      and     eax, 0xf000
0x12540:      cmp     eax, 0x4000

```

```

0x12545:    jne     0x12590
0x12547:    mov     byte ptr [rbx], 0x2f
0x1254a:    movzx   eax, byte ptr [rbx + 1]
0x1254e:    add     rbx, 1
0x12552:    test    al, al
0x12554:    jne     0x1251c
0x12556:    nop     word ptr cs:[rax + rax]
0x12560:    mov     rdi, rbp
0x12563:    call    0x18d10
0x12568:    xor     eax, eax
0x1256a:    mov     rdx, qword ptr [rsp + 0x98]
0x12572:    sub     rdx, qword ptr fs:[0x28]
0x1257b:    jne     0x125ae
0x1257d:    add     rsp, 0xa8
0x12584:    pop     rbx
0x12585:    pop     rbp
0x12586:    pop     r12
0x12588:    pop     r13
0x1258a:    ret
0x1258b:    nop     dword ptr [rax + rax]
0x12590:    mov     esi, r12d
0x12593:    mov     rdi, rbp
0x12596:    call    0x48c0
0x1259b:    test    eax, eax
0x1259d:    je      0x12547
0x1259f:    mov     rdi, rbp
0x125a2:    call    0x18d10
0x125a7:    mov     eax, 0xffffffff
0x125ac:    jmp     0x1256a
0x125ae:    call    0x49d0
0x125b3:    nop     word ptr cs:[rax + rax]
0x125be:    nop
0x125c0:    endbr64
0x125c4:    push    r15
0x125c6:    movsxd  rsi, esi
0x125c9:    mov     r10, rdx
0x125cc:    movsxd  r8, r8d
0x125cf:    push    r14
0x125d1:    lea     rax, [rsi + rsi*4]
0x125d5:    mov     rsi, rdi
0x125d8:    push    r13
0x125da:    shl     rax, 4
0x125de:    push    r12
0x125e0:    push    rbp
0x125e1:    push    rbx
0x125e2:    sub     rsp, 0x68
0x125e6:    mov     qword ptr [rsp + 0x48], rdi
0x125eb:    mov     rdi, rax

```

```

0x125ee:    mov     qword ptr [rsp + 0x50], rax
0x125f3:    add     rax, qword ptr [rsi + 0x38]
0x125f7:    mov     qword ptr [rsp + 0x20], rdx
0x125fc:    mov     rdx, rax
0x125ff:    mov     rax, qword ptr [rax]
0x12602:    mov     qword ptr [rsp + 0x38], rcx
0x12607:    mov     rcx, qword ptr [rcx + 0x38]
0x1260b:    mov     qword ptr [rsp + 0x10], rax
0x12610:    lea     rax, [r8 + r8*4]
0x12614:    shl     rax, 4
0x12618:    mov     qword ptr [rsp + 0x28], r9
0x1261d:    lea     rsi, [rcx + rax]
0x12621:    mov     qword ptr [rsp + 0x58], rax
0x12626:    mov     eax, dword ptr [rdx + 0x10]
0x12629:    mov     ecx, dword ptr [rcx + rdi + 0x10]
0x1262d:    mov     rbx, qword ptr [rsi]
0x12630:    mov     qword ptr [r9], 0
0x12637:    mov     r8d, eax
0x1263a:    mov     qword ptr [r10], 0
0x12641:    mov     edi, ecx
0x12643:    and     r8d, 0xf000
0x1264a:    and     edi, 0xf000
0x12650:    cmp     r8d, 0x4000
0x12657:    je      0x12669
0x12659:    and     eax, 0xd000
0x1265e:    cmp     eax, 0x8000
0x12663:    jne     0x12928
0x12669:    mov     rax, qword ptr [rdx + 8]
0x1266d:    mov     qword ptr [rsp + 8], rax
0x12672:    cmp     edi, 0x4000
0x12678:    je      0x12ac0
0x1267e:    and     ecx, 0xd000
0x12684:    mov     dword ptr [rsp + 0x40], 0
0x1268c:    mov     dword ptr [rsp + 0x44], 1
0x12694:    cmp     ecx, 0x8000
0x1269a:    je      0x12ac0
0x126a0:    cmp     qword ptr [rsp + 8], 0
0x126a6:    je      0x12c90
0x126ac:    mov     r14, qword ptr [rsp + 0x38]
0x126b1:    mov     rdi, qword ptr [r14 + 0x18]
0x126b5:    call    0x49b0
0x126ba:    mov     rdi, rbx
0x126bd:    mov     rbp, rax
0x126c0:    call    0x49b0
0x126c5:    lea     rdi, [rbp + rax + 2]
0x126ca:    call    0x1d670
0x126cf:    mov     r8, qword ptr [r14 + 0x18]
0x126d3:    mov     r9, rbx

```

```

0x126d6:    lea     rcx, [rip + 0xe024]
0x126dd:    mov     rdx, -1
0x126e4:    mov     esi, 1
0x126e9:    mov     rdi, rax
0x126ec:    mov     r12, rax
0x126ef:    xor     eax, eax
0x126f1:    call    0x4f10
0x126f6:    mov     rbx, qword ptr [rsp + 0x10]
0x126fb:    mov     rdi, rbx
0x126fe:    call    0x49b0
0x12703:    lea     rdi, [rax + 0x21]
0x12707:    call    0x1d670
0x1270c:    mov     r8, rbx
0x1270f:    mov     esi, 1
0x12714:    lea     rcx, [rip + 0xebbd]
0x1271b:    mov     rbp, rax
0x1271e:    mov     rdi, rax
0x12721:    mov     rdx, -1
0x12728:    xor     eax, eax
0x1272a:    call    0x4f10
0x1272f:    mov     edx, 1
0x12734:    mov     esi, 1
0x12739:    mov     rdi, rbp
0x1273c:    call    0xe480
0x12741:    xor     eax, eax
0x12743:    call    0xa820
0x12748:    mov     rdi, rbp
0x1274b:    call    0x18d10
0x12750:    mov     rdi, qword ptr [rsp + 0x10]
0x12755:    xor     esi, esi
0x12757:    xor     eax, eax
0x12759:    call    0x4d30
0x1275e:    mov     r15d, eax
0x12761:    cmp     eax, -1
0x12764:    je      0x12c6b
0x1276a:    mov     rdi, r12
0x1276d:    xor     esi, esi
0x1276f:    xor     eax, eax
0x12771:    call    0x4d30
0x12776:    mov     rdi, r12
0x12779:    mov     dword ptr [rsp + 0x1c], eax
0x1277d:    cmp     eax, -1
0x12780:    je      0x12c77
0x12786:    call    0x18d10
0x1278b:    mov     edi, 0x8000
0x12790:    xor     r12d, r12d
0x12793:    call    0x1d670
0x12798:    mov     edi, 0x8000

```

```

0x1279d:    mov     rbp, rax
0x127a0:    call   0x1d670
0x127a5:    mov     rdi, qword ptr [rsp + 0x10]
0x127aa:    mov     r13, rax
0x127ad:    call   0x49b0
0x127b2:    lea     rdi, [rax + 0x21]
0x127b6:    call   0x1d670
0x127bb:    mov     dword ptr [rsp + 0x18], r15d
0x127c0:    movabs  rdx, 0xa3d70a3d70a3d70b
0x127ca:    mov     r14, rax
0x127cd:    mov     rax, qword ptr [rsp + 8]
0x127d2:    imul    rdx
0x127d5:    mov     rax, qword ptr [rsp + 8]
0x127da:    add     rdx, rax
0x127dd:    sar     rax, 0x3f
0x127e1:    sar     rdx, 6
0x127e5:    sub     rdx, rax
0x127e8:    mov     qword ptr [rsp + 0x30], rdx
0x127ed:    jmp     0x1282b
0x127ef:    nop
0x127f0:    mov     r8, qword ptr [rsp + 0x10]
0x127f5:    mov     esi, 1
0x127fa:    mov     rdi, r14
0x127fd:    xor     eax, eax
0x127ff:    lea     rcx, [rip + 0xeab8]
0x12806:    mov     rdx, -1
0x1280d:    call   0x4f10
0x12812:    mov     edx, 1
0x12817:    mov     esi, 1
0x1281c:    mov     rdi, r14
0x1281f:    call   0xe480
0x12824:    xor     eax, eax
0x12826:    call   0xa820
0x1282b:    mov     ecx, dword ptr [rip + 0x1927f]
0x12831:    test    ecx, ecx
0x12833:    jne     0x12a30
0x12839:    mov     edi, 1
0x1283e:    call   0x18b70
0x12843:    mov     edi, dword ptr [rsp + 0x18]
0x12847:    mov     edx, 0x8000
0x1284c:    mov     rsi, rbp
0x1284f:    call   0x47a0
0x12854:    xor     edi, edi
0x12856:    mov     rbx, rax
0x12859:    call   0x18b70
0x1285e:    test    ebx, ebx
0x12860:    js      0x12af0
0x12866:    mov     rcx, qword ptr [rsp + 0x20]

```

```

0x1286b:    movsxd   rax, ebx
0x1286e:    mov     edi, 1
0x12873:    add     qword ptr [rcx], rax
0x12876:    call    0x18b70
0x1287b:    mov     edi, dword ptr [rsp + 0x1c]
0x1287f:    mov     edx, 0x8000
0x12884:    mov     rsi, r13
0x12887:    call    0x47a0
0x1288c:    xor     edi, edi
0x1288e:    mov     r15, rax
0x12891:    call    0x18b70
0x12896:    test    r15d, r15d
0x12899:    js      0x12b20
0x1289f:    mov     rax, qword ptr [rsp + 0x28]
0x128a4:    movsxd   rdx, r15d
0x128a7:    add     qword ptr [rax], rdx
0x128aa:    cmp     r15d, ebx
0x128ad:    cmovle   ebx, r15d
0x128b1:    test    ebx, ebx
0x128b3:    je      0x12b40
0x128b9:    movsxd   r15, ebx
0x128bc:    mov     rsi, r13
0x128bf:    mov     rdi, rbp
0x128c2:    mov     rdx, r15
0x128c5:    call    0x4af0
0x128ca:    test    eax, eax
0x128cc:    jne     0x12b70
0x128d2:    add     r12, r15
0x128d5:    cmp     qword ptr [rsp + 8], 0
0x128db:    je      0x127f0
0x128e1:    cmp     qword ptr [rsp + 8], 0x1869f
0x128ea:    jle     0x12a10
0x128f0:    mov     rax, r12
0x128f3:    cqo
0x128f5:    idiv    qword ptr [rsp + 0x30]
0x128fa:    mov     r8, rax
0x128fd:    mov     r9, qword ptr [rsp + 0x10]
0x12902:    mov     esi, 1
0x12907:    mov     rdi, r14
0x1290a:    xor     eax, eax
0x1290c:    lea     rcx, [rip + 0xe9b4]
0x12913:    mov     rdx, -1
0x1291a:    call    0x4f10
0x1291f:    jmp     0x12812
0x12924:    nop     dword ptr [rax]
0x12928:    cmp     edi, 0x4000
0x1292e:    je      0x129f0
0x12934:    and     ecx, 0xd000

```



```

0x1293a:    cmp     ecx, 0x8000
0x12940:    je      0x129f0
0x12946:    mov     r14, qword ptr [rsp + 0x38]
0x1294b:    mov     rdi, qword ptr [r14 + 0x18]
0x1294f:    call    0x49b0
0x12954:    mov     rdi, rbx
0x12957:    mov     rbp, rax
0x1295a:    call    0x49b0
0x1295f:    lea     rdi, [rbp + rax + 2]
0x12964:    call    0x1d670
0x12969:    mov     r8, qword ptr [r14 + 0x18]
0x1296d:    mov     r9, rbx
0x12970:    lea     rcx, [rip + 0xdd8a]
0x12977:    mov     rdx, -1
0x1297e:    mov     esi, 1
0x12983:    mov     rdi, rax
0x12986:    mov     r12, rax
0x12989:    xor     eax, eax
0x1298b:    call    0x4f10
0x12990:    mov     rbx, qword ptr [rsp + 0x10]
0x12995:    mov     rdi, rbx
0x12998:    call    0x49b0
0x1299d:    lea     rdi, [rax + 0x21]
0x129a1:    call    0x1d670
0x129a6:    mov     r8, rbx
0x129a9:    mov     esi, 1
0x129ae:    lea     rcx, [rip + 0xe909]
0x129b5:    mov     rdi, rax
0x129b8:    mov     rbp, rax
0x129bb:    mov     rdx, -1
0x129c2:    xor     eax, eax
0x129c4:    call    0x4f10
0x129c9:    mov     dword ptr [rsp + 0x44], 1
0x129d1:    mov     qword ptr [rsp + 8], 0
0x129da:    mov     dword ptr [rsp + 0x40], 1
0x129e2:    jmp     0x1272f
0x129e7:    nop     word ptr [rax + rax]
0x129f0:    mov     rax, qword ptr [rsi + 8]
0x129f4:    mov     dword ptr [rsp + 0x40], 1
0x129fc:    mov     dword ptr [rsp + 0x44], 0
0x12a04:    mov     qword ptr [rsp + 8], rax
0x12a09:    jmp     0x126a0
0x12a0e:    nop
0x12a10:    lea     rax, [r12 + r12*4]
0x12a14:    lea     rax, [rax + rax*4]
0x12a18:    shl     rax, 2
0x12a1c:    cqo
0x12a1e:    idiv    qword ptr [rsp + 8]

```

```

0x12a23:    mov     r8, rax
0x12a26:    jmp     0x128fd
0x12a2b:    nop     dword ptr [rax + rax]
0x12a30:    mov     dword ptr [rip + 0x19076], 0
0x12a3a:    xor     eax, eax
0x12a3c:    call    0xc8c0
0x12a41:    mov     edx, 4
0x12a46:    lea     rsi, [rip + 0xdae5]
0x12a4d:    lea     rdi, [rip + 0xe7a5]
0x12a54:    mov     r15, rax
0x12a57:    call    0x8070
0x12a5c:    mov     rdi, r15
0x12a5f:    mov     ebx, eax
0x12a61:    call    0xc970
0x12a66:    xor     eax, eax
0x12a68:    and     ebx, 0xfffffffff
0x12a6b:    call    0xd870
0x12a70:    xor     eax, eax
0x12a72:    call    0xd740
0x12a77:    xor     eax, eax
0x12a79:    call    0xa820
0x12a7e:    cmp     bl, 0x4e
0x12a81:    je      0x12839
0x12a87:    mov     rdi, r14
0x12a8a:    mov     r15d, dword ptr [rsp + 0x18]
0x12a8f:    mov     r12, -1
0x12a96:    call    0x18d10
0x12a9b:    mov     rdi, rbp
0x12a9e:    call    0x18d10
0x12aa3:    mov     rdi, r13
0x12aa6:    call    0x18d10
0x12aab:    mov     edi, r15d
0x12aae:    call    0x4ab0
0x12ab3:    mov     edi, dword ptr [rsp + 0x1c]
0x12ab7:    call    0x4ab0
0x12abc:    jmp     0x12b09
0x12abe:    nop
0x12ac0:    mov     rcx, qword ptr [rsp + 8]
0x12ac5:    mov     rax, qword ptr [rsi + 8]
0x12ac9:    mov     dword ptr [rsp + 0x40], 0
0x12ad1:    mov     dword ptr [rsp + 0x44], 0
0x12ad9:    cmp     rcx, rax
0x12adc:    cmovge  rax, rcx
0x12ae0:    mov     qword ptr [rsp + 8], rax
0x12ae5:    jmp     0x126a0
0x12aea:    nop     word ptr [rax + rax]
0x12af0:    mov     edi, dword ptr [rsp + 0x18]
0x12af4:    mov     r12, -2

```

```

0x12afb:      call    0x4ab0
0x12b00:      mov     edi, dword ptr [rsp + 0x1c]
0x12b04:      call    0x4ab0
0x12b09:      add     rsp, 0x68
0x12b0d:      mov     rax, r12
0x12b10:      pop     rbx
0x12b11:      pop     rbp
0x12b12:      pop     r12
0x12b14:      pop     r13
0x12b16:      pop     r14
0x12b18:      pop     r15
0x12b1a:      ret
0x12b1b:      nop     dword ptr [rax + rax]
0x12b20:      mov     edi, dword ptr [rsp + 0x18]
0x12b24:      mov     r12, -3
0x12b2b:      call    0x4ab0
0x12b30:      mov     edi, dword ptr [rsp + 0x1c]
0x12b34:      call    0x4ab0
0x12b39:      jmp     0x12b09
0x12b3b:      nop     dword ptr [rax + rax]
0x12b40:      mov     r15d, dword ptr [rsp + 0x18]
0x12b45:      mov     rdi, r14
0x12b48:      call    0x18d10
0x12b4d:      mov     rdi, rbp
0x12b50:      call    0x18d10
0x12b55:      mov     rdi, r13
0x12b58:      call    0x18d10
0x12b5d:      mov     edi, r15d
0x12b60:      call    0x4ab0
0x12b65:      mov     edi, dword ptr [rsp + 0x1c]
0x12b69:      call    0x4ab0
0x12b6e:      jmp     0x12b09
0x12b70:      mov     r15d, dword ptr [rsp + 0x18]
0x12b75:      xor     eax, eax
0x12b77:      nop     word ptr [rax + rax]
0x12b80:      movzx   ecx, byte ptr [r13 + rax]
0x12b86:      mov     edx, eax
0x12b88:      cmp     byte ptr [rbp + rax], cl
0x12b8c:      jne     0x12b99
0x12b8e:      lea     edx, [rax + 1]
0x12b91:      add     rax, 1
0x12b95:      cmp     ebx, eax
0x12b97:      jg      0x12b80
0x12b99:      movsxd  rdx, edx
0x12b9c:      add     r12, rdx
0x12b9f:      mov     edx, dword ptr [rsp + 0x40]
0x12ba3:      test    edx, edx
0x12ba5:      jne     0x12bea

```

```

0x12ba7:    mov     rax, qword ptr [rsp + 0x48]
0x12bac:    mov     rcx, qword ptr [rsp + 0x50]
0x12bb1:    mov     rax, qword ptr [rax + 0x38]
0x12bb5:    mov     rax, qword ptr [rax + rcx + 8]
0x12bba:    mov     rcx, qword ptr [rsp + 0x20]
0x12bbf:    mov     qword ptr [rcx], rax
0x12bc2:    mov     eax, dword ptr [rsp + 0x44]
0x12bc6:    test    eax, eax
0x12bc8:    jne     0x12c24
0x12bca:    mov     rax, qword ptr [rsp + 0x38]
0x12bcf:    mov     rcx, qword ptr [rsp + 0x58]
0x12bd4:    mov     rax, qword ptr [rax + 0x38]
0x12bd8:    mov     rax, qword ptr [rax + rcx + 8]
0x12bdd:    mov     rcx, qword ptr [rsp + 0x28]
0x12be2:    mov     qword ptr [rcx], rax
0x12be5:    jmp     0x12b45
0x12bea:    mov     edi, 1
0x12bef:    call    0x18b70
0x12bf4:    mov     rbx, qword ptr [rsp + 0x20]
0x12bf9:    jmp     0x12c07
0x12bfb:    nop     dword ptr [rax + rax]
0x12c00:    cdqe
0x12c02:    add     qword ptr [rbx], rax
0x12c05:    je      0x12c1b
0x12c07:    mov     edx, 0x8000
0x12c0c:    mov     rsi, rbp
0x12c0f:    mov     edi, r15d
0x12c12:    call    0x47a0
0x12c17:    test    eax, eax
0x12c19:    jg      0x12c00
0x12c1b:    xor     edi, edi
0x12c1d:    call    0x18b70
0x12c22:    jmp     0x12bc2
0x12c24:    mov     edi, 1
0x12c29:    call    0x18b70
0x12c2e:    mov     qword ptr [rsp + 8], rbp
0x12c33:    mov     rbx, qword ptr [rsp + 0x28]
0x12c38:    mov     ebp, dword ptr [rsp + 0x1c]
0x12c3c:    jmp     0x12c47
0x12c3e:    nop
0x12c40:    cdqe
0x12c42:    add     qword ptr [rbx], rax
0x12c45:    je      0x12c5a
0x12c47:    mov     edx, 0x8000
0x12c4c:    mov     rsi, r13
0x12c4f:    mov     edi, ebp
0x12c51:    call    0x47a0
0x12c56:    test    eax, eax

```

```

0x12c58:      jg      0x12c40
0x12c5a:      xor     edi, edi
0x12c5c:      mov     rbp, qword ptr [rsp + 8]
0x12c61:      call    0x18b70
0x12c66:      jmp     0x12b45
0x12c6b:      mov     r12, -4
0x12c72:      jmp     0x12b09
0x12c77:      call    0x18d10
0x12c7c:      mov     edi, r15d
0x12c7f:      mov     r12, -5
0x12c86:      call    0x4ab0
0x12c8b:      jmp     0x12b09
0x12c90:      xor     r12d, r12d
0x12c93:      jmp     0x12b09
0x12c98:      nop     dword ptr [rax + rax]
0x12ca0:      endbr64
0x12ca4:      push    r15
0x12ca6:      push    r14
0x12ca8:      push    r13
0x12caa:      push    r12
0x12cac:      push    rbp
0x12cad:      mov     rbp, rsi
0x12cb0:      push    rbx
0x12cb1:      mov     rbx, rdi
0x12cb4:      sub     rsp, 0x38
0x12cb8:      mov     r13d, dword ptr [rdi + 0x28]
0x12cbc:      mov     rdi, qword ptr [rip + 0x1780d]
0x12cc3:      mov     rax, qword ptr fs:[0x28]
0x12ccc:      mov     qword ptr [rsp + 0x28], rax
0x12cd1:      xor     eax, eax
0x12cd3:      mov     r14d, dword ptr [rsi + 0x28]
0x12cd7:      call    0xdb60
0x12cdc:      xor     eax, eax
0x12cde:      call    0xa820
0x12ce3:      mov     r15, qword ptr [rbx + 0x18]
0x12ce7:      mov     rsi, qword ptr [rbp + 0x18]
0x12ceb:      mov     rdi, r15
0x12cee:      call    0x4b30
0x12cf3:      mov     rdx, qword ptr [rbx + 0x38]
0x12cf7:      mov     r8d, eax
0x12cfa:      movsxd  rax, r13d
0x12cfd:      lea     r12, [rax + rax*4]
0x12d01:      shl     r12, 4
0x12d05:      add     rdx, r12
0x12d08:      test    r8d, r8d
0x12d0b:      jne     0x12d39
0x12d0d:      movsxd  rax, r14d
0x12d10:      mov     rdi, qword ptr [rdx]

```

```

0x12d13:    mov     qword ptr [rsp + 8], rdx
0x12d18:    lea     rax, [rax + rax*4]
0x12d1c:    shl     rax, 4
0x12d20:    add     rax, qword ptr [rbp + 0x38]
0x12d24:    mov     rsi, qword ptr [rax]
0x12d27:    call    0x4b30
0x12d2c:    mov     rdx, qword ptr [rsp + 8]
0x12d31:    test    eax, eax
0x12d33:    je      0x12e40
0x12d39:    cmp     byte ptr [rdx + 0x48], 1
0x12d3d:    jne     0x12d57
0x12d3f:    movsxd  rax, r14d
0x12d42:    lea     r10, [rax + rax*4]
0x12d46:    mov     rax, qword ptr [rbp + 0x38]
0x12d4a:    shl     r10, 4
0x12d4e:    add     rax, r10
0x12d51:    cmp     byte ptr [rax + 0x48], 1
0x12d55:    je      0x12da0
0x12d57:    mov     ecx, 0x1e
0x12d5c:    xor     edx, edx
0x12d5e:    mov     rsi, r15
0x12d61:    xor     eax, eax
0x12d63:    lea     rdi, [rip + 0xede6]
0x12d6a:    call    0xf760
0x12d6f:    xor     eax, eax
0x12d71:    call    0xe4e0
0x12d76:    mov     rax, qword ptr [rsp + 0x28]
0x12d7b:    sub     rax, qword ptr fs:[0x28]
0x12d84:    jne     0x12fbc
0x12d8a:    add     rsp, 0x38
0x12d8e:    xor     eax, eax
0x12d90:    pop     rbx
0x12d91:    pop     rbp
0x12d92:    pop     r12
0x12d94:    pop     r13
0x12d96:    pop     r14
0x12d98:    pop     r15
0x12d9a:    jmp     0xa820
0x12d9f:    nop
0x12da0:    mov     rcx, qword ptr [rax + 8]
0x12da4:    cmp     qword ptr [rdx + 8], rcx
0x12da8:    je      0x12df2
0x12daa:    mov     edx, dword ptr [rdx + 0x10]
0x12dad:    mov     ecx, edx
0x12daf:    and     ecx, 0xf000
0x12db5:    cmp     ecx, 0x4000
0x12dbb:    je      0x12dcb
0x12dbd:    and     edx, 0xd000

```

```

0x12dc3:      cmp     edx, 0x8000
0x12dc9:      jne     0x12df2
0x12dcb:      mov     eax, dword ptr [rax + 0x10]
0x12dce:      mov     edx, eax
0x12dd0:      and     edx, 0xf000
0x12dd6:      cmp     edx, 0x4000
0x12ddc:      je      0x12ee8
0x12de2:      and     eax, 0xd000
0x12de7:      cmp     eax, 0x8000
0x12dec:      je      0x12ee8
0x12df2:      mov     esi, r13d
0x12df5:      lea     rdx, [rsp + 0x18]
0x12dfa:      lea     r9, [rsp + 0x20]
0x12dff:      mov     r8d, r14d
0x12e02:      mov     rcx, rbp
0x12e05:      mov     rdi, rbx
0x12e08:      xor     eax, eax
0x12e0a:      mov     qword ptr [rsp + 8], r10
0x12e0f:      call    0x125c0
0x12e14:      mov     r13, rax
0x12e17:      lea     eax, [rax + 5]
0x12e1a:      cmp     r13d, -5
0x12e1e:      jb      0x12f17
0x12e24:      lea     rdx, [rip + 0xeac5]
0x12e2b:      mov     r10, qword ptr [rsp + 8]
0x12e30:      movsxd  rax, dword ptr [rdx + rax*4]
0x12e34:      add     rax, rdx
0x12e37:      jmp     rax
0x12e3a:      -----

```

```

0x12e3e:      -----
0x12e3f:      -----
0x12e40:      mov     rax, qword ptr [rsp + 0x28]
0x12e45:      sub     rax, qword ptr fs:[0x28]
0x12e4e:      jne     0x12fbc
0x12e54:      add     rsp, 0x38
0x12e58:      mov     edx, 0x1c
0x12e5d:      xor     esi, esi
0x12e5f:      pop     rbx
0x12e60:      lea     rdi, [rip + 0xec11]
0x12e67:      pop     rbp
0x12e68:      pop     r12
0x12e6a:      pop     r13
0x12e6c:      pop     r14
0x12e6e:      pop     r15
0x12e70:      jmp     0x8070

```

```

0x12e75:      nop      dword ptr [rax]
0x12e78:      mov      rax, qword ptr [rbx + 0x38]
0x12e7c:      mov      rsi, qword ptr [rax + r12]
0x12e80:      mov      ecx, 0x1e
0x12e85:      xor      edx, edx
0x12e87:      lea      rdi, [rip + 0xec4a]
0x12e8e:      xor      eax, eax
0x12e90:      call     0xf760
0x12e95:      jmp      0x12d6f
0x12e9a:      nop      word ptr [rax + rax]
0x12ea0:      mov      rax, qword ptr [rbp + 0x38]
0x12ea4:      mov      rsi, qword ptr [rax + r10]
0x12ea8:      jmp      0x12e80
0x12eaa:      nop      word ptr [rax + rax]
0x12eb0:      mov      rax, qword ptr [rbx + 0x38]
0x12eb4:      mov      rsi, qword ptr [rax + r12]
0x12eb8:      mov      ecx, 0x1e
0x12ebd:      xor      edx, edx
0x12ebf:      lea      rdi, [rip + 0xe423]
0x12ec6:      xor      eax, eax
0x12ec8:      call     0xf760
0x12ecd:      jmp      0x12d6f
0x12ed2:      nop      word ptr [rax + rax]
0x12ed8:      mov      rax, qword ptr [rbp + 0x38]
0x12edc:      mov      rsi, qword ptr [rax + r10]
0x12ee0:      jmp      0x12eb8
0x12ee2:      nop      word ptr [rax + rax]
0x12ee8:      mov      edx, 0xc
0x12eed:      lea      rsi, [rip + 0xd63e]
0x12ef4:      mov      qword ptr [rsp + 8], r10
0x12ef9:      lea      rdi, [rip + 0xebb0]
0x12f00:      call     0x8070
0x12f05:      mov      r10, qword ptr [rsp + 8]
0x12f0a:      cmp      al, 0x79
0x12f0c:      jne      0x12d6f
0x12f12:      jmp      0x12df2
0x12f17:      mov      rax, qword ptr [rsp + 0x18]
0x12f1c:      mov      rdx, qword ptr [rsp + 0x20]
0x12f21:      cmp      rax, r13
0x12f24:      je       0x12f87
0x12f26:      cmp      rax, rdx
0x12f29:      cmovg    rax, rdx
0x12f2d:      cmp      rax, r13
0x12f30:      je       0x12fa4
0x12f32:      mov      edi, 0x80
0x12f37:      call     0x1d670
0x12f3c:      mov      r9, r13
0x12f3f:      mov      r8, r13

```



```

0x12f42:    mov     esi, 1
0x12f47:    mov     rbp, rax
0x12f4a:    mov     rdi, rax
0x12f4d:    xor     eax, eax
0x12f4f:    mov     rdx, -1
0x12f56:    lea     rcx, [rip + 0xe3af]
0x12f5d:    call    0x4f10
0x12f62:    mov     ecx, 0x14
0x12f67:    xor     edx, edx
0x12f69:    mov     rsi, rbp
0x12f6c:    lea     rdi, [rip + 0xe3a5]
0x12f73:    xor     eax, eax
0x12f75:    call    0xf760
0x12f7a:    mov     rdi, rbp
0x12f7d:    call    0x18d10
0x12f82:    jmp     0x12d6f
0x12f87:    cmp     rax, rdx
0x12f8a:    jne     0x12f26
0x12f8c:    mov     edx, 0x14
0x12f91:    xor     esi, esi
0x12f93:    lea     rdi, [rip + 0xe365]
0x12f9a:    call    0x8070
0x12f9f:    jmp     0x12d6f
0x12fa4:    mov     edx, 0x14
0x12fa9:    xor     esi, esi
0x12fab:    lea     rdi, [rip + 0xeb4e]
0x12fb2:    call    0x8070
0x12fb7:    jmp     0x12d6f
0x12fbc:    call    0x49d0
0x12fc1:    nop     word ptr cs:[rax + rax]
0x12fcc:    nop     dword ptr [rax]
0x12fd0:    endbr64
0x12fd4:    push    r15
0x12fd6:    push    r14
0x12fd8:    push    r13
0x12fda:    push    r12
0x12fdc:    mov     r12, rdi
0x12fdf:    push    rbp
0x12fe0:    mov     rbp, rsi
0x12fe3:    push    rbx
0x12fe4:    sub     rsp, 0x48
0x12fe8:    mov     rdi, qword ptr [rip + 0x174d9]
0x12fef:    mov     dword ptr [rsp + 0x14], edx
0x12ff3:    mov     rax, qword ptr fs:[0x28]
0x12ffc:    mov     qword ptr [rsp + 0x38], rax
0x13001:    xor     eax, eax
0x13003:    call    0xdb60
0x13008:    xor     eax, eax

```

```

0x1300a:    call    0xa820
0x1300f:    mov     rsi, qword ptr [rbp + 0x18]
0x13013:    mov     rdi, qword ptr [r12 + 0x18]
0x13018:    call    0x4b30
0x1301d:    test    eax, eax
0x1301f:    je      0x132d0
0x13025:    mov     ecx, dword ptr [r12 + 0x68]
0x1302a:    test    ecx, ecx
0x1302c:    jle     0x1306d
0x1302e:    lea     edx, [rcx - 1]
0x13031:    mov     rsi, qword ptr [r12 + 0x38]
0x13036:    lea     rdx, [rdx + rdx*4]
0x1303a:    shl     rdx, 4
0x1303e:    lea     rax, [rsi + 0x49]
0x13042:    lea     rdx, [rsi + rdx + 0x99]
0x1304a:    nop     word ptr [rax + rax]
0x13050:    cmp     byte ptr [rax], 0
0x13053:    jne     0x13064
0x13055:    cmp     byte ptr [rax - 1], 0
0x13059:    je      0x13064
0x1305b:    mov     byte ptr [rax], 1
0x1305e:    add     dword ptr [r12 + 0x70], 1
0x13064:    add     rax, 0x50
0x13068:    cmp     rax, rdx
0x1306b:    jne     0x13050
0x1306d:    mov     edx, dword ptr [rbp + 0x68]
0x13070:    test    edx, edx
0x13072:    jle     0x130ab
0x13074:    sub     edx, 1
0x13077:    mov     rsi, qword ptr [rbp + 0x38]
0x1307b:    lea     rdx, [rdx + rdx*4]
0x1307f:    shl     rdx, 4
0x13083:    lea     rax, [rsi + 0x49]
0x13087:    lea     rdx, [rsi + rdx + 0x99]
0x1308f:    nop
0x13090:    cmp     byte ptr [rax], 0
0x13093:    jne     0x130a2
0x13095:    cmp     byte ptr [rax - 1], 0
0x13099:    je      0x130a2
0x1309b:    mov     byte ptr [rax], 1
0x1309e:    add     dword ptr [rbp + 0x70], 1
0x130a2:    add     rax, 0x50
0x130a6:    cmp     rax, rdx
0x130a9:    jne     0x13090
0x130ab:    lea     rax, [rsp + 0x30]
0x130b0:    mov     dword ptr [rsp + 0x10], 1
0x130b8:    mov     ebx, 0x50
0x130bd:    mov     qword ptr [rsp + 0x18], rax

```

```

0x130c2:    cmp     ecx, 1
0x130c5:    jg      0x130e8
0x130c7:    jmp     0x13188
0x130cc:    nop     dword ptr [rax]
0x130d0:    add     dword ptr [rsp + 0x10], 1
0x130d5:    add     rbx, 0x50
0x130d9:    mov     eax, dword ptr [rsp + 0x10]
0x130dd:    cmp     dword ptr [r12 + 0x68], eax
0x130e2:    jle     0x13188
0x130e8:    mov     rax, qword ptr [r12 + 0x38]
0x130ed:    cmp     byte ptr [rax + rbx + 0x48], 1
0x130f2:    jne     0x130d0
0x130f4:    cmp     dword ptr [rbp + 0x68], 1
0x130f8:    jle     0x130d0
0x130fa:    mov     r14d, 0x50
0x13100:    mov     r13d, 1
0x13106:    jmp     0x1311e
0x13108:    nop     dword ptr [rax + rax]
0x13110:    add     r13d, 1
0x13114:    add     r14, 0x50
0x13118:    cmp     dword ptr [rbp + 0x68], r13d
0x1311c:    jle     0x130d0
0x1311e:    xor     eax, eax
0x13120:    call    0x18ab0
0x13125:    mov     r15, qword ptr [rbp + 0x38]
0x13129:    add     r15, r14
0x1312c:    cmp     byte ptr [r15 + 0x48], 1
0x13131:    jne     0x13110
0x13133:    mov     rdx, qword ptr [r12 + 0x38]
0x13138:    mov     rsi, qword ptr [r15]
0x1313b:    add     rdx, rbx
0x1313e:    mov     rdi, qword ptr [rdx]
0x13141:    mov     qword ptr [rsp + 8], rdx
0x13146:    call    0x4b30
0x1314b:    mov     rdx, qword ptr [rsp + 8]
0x13150:    test    eax, eax
0x13152:    jne     0x13110
0x13154:    mov     rax, qword ptr [r15 + 8]
0x13158:    cmp     qword ptr [rdx + 8], rax
0x1315c:    je      0x13440
0x13162:    mov     rsi, qword ptr [rdx + 0x20]
0x13166:    mov     rax, qword ptr [r15 + 0x20]
0x1316a:    cmp     rsi, rax
0x1316d:    jl      0x133f3
0x13173:    mov     byte ptr [r15 + 0x49], 0
0x13178:    sub     dword ptr [rbp + 0x70], 1
0x1317c:    jmp     0x130d0
0x13181:    nop     dword ptr [rax]

```

```

0x13188:    xor     eax, eax
0x1318a:    call    0xe4e0
0x1318f:    xor     eax, eax
0x13191:    mov     rdi, r12
0x13194:    call    0x10ec0
0x13199:    xor     eax, eax
0x1319b:    mov     rdi, r12
0x1319e:    call    0x10160
0x131a3:    cmp     byte ptr [r12 + 0x6d], 0
0x131a9:    jne     0x13290
0x131af:    xor     eax, eax
0x131b1:    mov     rdi, r12
0x131b4:    call    0x10240
0x131b9:    cmp     byte ptr [r12 + 0x6d], 0
0x131bf:    jne     0x13280
0x131c5:    xor     eax, eax
0x131c7:    mov     rdi, rbp
0x131ca:    call    0x10ec0
0x131cf:    xor     eax, eax
0x131d1:    mov     rdi, rbp
0x131d4:    call    0x10160
0x131d9:    cmp     byte ptr [rbp + 0x6d], 0
0x131dd:    jne     0x13258
0x131df:    xor     eax, eax
0x131e1:    mov     rdi, rbp
0x131e4:    call    0x10240
0x131e9:    cmp     byte ptr [rbp + 0x6d], 0
0x131ed:    jne     0x13238
0x131ef:    xor     eax, eax
0x131f1:    call    0xa820
0x131f6:    mov     edx, dword ptr [r12 + 0x70]
0x131fb:    test    edx, edx
0x131fd:    jne     0x132a0
0x13203:    mov     eax, dword ptr [rbp + 0x70]
0x13206:    test    eax, eax
0x13208:    jne     0x132a0
0x1320e:    mov     rax, qword ptr [rsp + 0x38]
0x13213:    sub     rax, qword ptr fs:[0x28]
0x1321c:    jne     0x134c3
0x13222:    add     rsp, 0x48
0x13226:    pop     rbx
0x13227:    pop     rbp
0x13228:    pop     r12
0x1322a:    pop     r13
0x1322c:    pop     r14
0x1322e:    pop     r15
0x13230:    ret
0x13231:    nop     dword ptr [rax]

```

```

0x13238:    mov     rdi, rbp
0x1323b:    xor     eax, eax
0x1323d:    call    0x10d70
0x13242:    xor     eax, eax
0x13244:    call    0xa820
0x13249:    mov     edx, dword ptr [r12 + 0x70]
0x1324e:    test    edx, edx
0x13250:    je      0x13203
0x13252:    jmp     0x132a0
0x13254:    nop     dword ptr [rax]
0x13258:    xor     eax, eax
0x1325a:    mov     rdi, rbp
0x1325d:    call    0xec50
0x13262:    xor     eax, eax
0x13264:    mov     rdi, rbp
0x13267:    call    0x10240
0x1326c:    cmp     byte ptr [rbp + 0x6d], 0
0x13270:    je      0x131ef
0x13276:    jmp     0x13238
0x13278:    nop     dword ptr [rax + rax]
0x13280:    mov     rdi, r12
0x13283:    xor     eax, eax
0x13285:    call    0x10d70
0x1328a:    jmp     0x131c5
0x1328f:    nop
0x13290:    mov     rdi, r12
0x13293:    xor     eax, eax
0x13295:    call    0xec50
0x1329a:    jmp     0x131af
0x1329f:    nop
0x132a0:    mov     rax, qword ptr [rsp + 0x38]
0x132a5:    sub     rax, qword ptr fs:[0x28]
0x132ae:    jne     0x134c3
0x132b4:    add     rsp, 0x48
0x132b8:    xor     eax, eax
0x132ba:    pop     rbx
0x132bb:    pop     rbp
0x132bc:    pop     r12
0x132be:    pop     r13
0x132c0:    pop     r14
0x132c2:    pop     r15
0x132c4:    jmp     0xa5a0
0x132c9:    nop     dword ptr [rax]
0x132d0:    mov     edx, 0x1c
0x132d5:    xor     esi, esi
0x132d7:    lea     rdi, [rip + 0xe89a]
0x132de:    call    0x8070
0x132e3:    mov     edx, dword ptr [r12 + 0x68]

```

```

0x132e8:    test    edx, edx
0x132ea:    jle     0x1331c
0x132ec:    sub     edx, 1
0x132ef:    mov     rcx, qword ptr [r12 + 0x38]
0x132f4:    lea     rdx, [rdx + rdx*4]
0x132f8:    shl     rdx, 4
0x132fc:    lea     rax, [rcx + 0x49]
0x13300:    lea     rdx, [rcx + rdx + 0x99]
0x13308:    nop     dword ptr [rax + rax]
0x13310:    mov     byte ptr [rax], 0
0x13313:    add     rax, 0x50
0x13317:    cmp     rdx, rax
0x1331a:    jne     0x13310
0x1331c:    mov     dword ptr [r12 + 0x70], 0
0x13325:    mov     edx, dword ptr [rbp + 0x68]
0x13328:    test    edx, edx
0x1332a:    jle     0x1335c
0x1332c:    sub     edx, 1
0x1332f:    mov     rcx, qword ptr [rbp + 0x38]
0x13333:    lea     rdx, [rdx + rdx*4]
0x13337:    shl     rdx, 4
0x1333b:    lea     rax, [rcx + 0x49]
0x1333f:    lea     rdx, [rcx + rdx + 0x99]
0x13347:    nop     word ptr [rax + rax]
0x13350:    mov     byte ptr [rax], 0
0x13353:    add     rax, 0x50
0x13357:    cmp     rdx, rax
0x1335a:    jne     0x13350
0x1335c:    mov     dword ptr [rbp + 0x70], 0
0x13363:    xor     eax, eax
0x13365:    mov     rdi, r12
0x13368:    call    0x10ec0
0x1336d:    xor     eax, eax
0x1336f:    mov     rdi, r12
0x13372:    call    0x10160
0x13377:    cmp     byte ptr [r12 + 0x6d], 0
0x1337d:    jne     0x13431
0x13383:    xor     eax, eax
0x13385:    mov     rdi, r12
0x13388:    call    0x10240
0x1338d:    cmp     byte ptr [r12 + 0x6d], 0
0x13393:    jne     0x13422
0x13399:    xor     eax, eax
0x1339b:    mov     rdi, rbp
0x1339e:    call    0x10ec0
0x133a3:    xor     eax, eax
0x133a5:    mov     rdi, rbp
0x133a8:    call    0x10160

```

```

0x133ad:    cmp     byte ptr [rbp + 0x6d], 0
0x133b1:    jne     0x13402
0x133b3:    xor     eax, eax
0x133b5:    mov     rdi, rbp
0x133b8:    call    0x10240
0x133bd:    cmp     byte ptr [rbp + 0x6d], 0
0x133c1:    je      0x1320e
0x133c7:    mov     rax, qword ptr [rsp + 0x38]
0x133cc:    sub     rax, qword ptr fs:[0x28]
0x133d5:    jne     0x134c3
0x133db:    add     rsp, 0x48
0x133df:    mov     rdi, rbp
0x133e2:    xor     eax, eax
0x133e4:    pop     rbx
0x133e5:    pop     rbp
0x133e6:    pop     r12
0x133e8:    pop     r13
0x133ea:    pop     r14
0x133ec:    pop     r15
0x133ee:    jmp     0x10d70
0x133f3:    mov     byte ptr [rdx + 0x49], 0
0x133f7:    sub     dword ptr [r12 + 0x70], 1
0x133fd:    jmp     0x130d0
0x13402:    xor     eax, eax
0x13404:    mov     rdi, rbp
0x13407:    call    0xec50
0x1340c:    xor     eax, eax
0x1340e:    mov     rdi, rbp
0x13411:    call    0x10240
0x13416:    cmp     byte ptr [rbp + 0x6d], 0
0x1341a:    je      0x1320e
0x13420:    jmp     0x133c7
0x13422:    mov     rdi, r12
0x13425:    xor     eax, eax
0x13427:    call    0x10d70
0x1342c:    jmp     0x13399
0x13431:    mov     rdi, r12
0x13434:    xor     eax, eax
0x13436:    call    0xec50
0x1343b:    jmp     0x13383
0x13440:    mov     ecx, dword ptr [rsp + 0x14]
0x13444:    test    ecx, ecx
0x13446:    je      0x13468
0x13448:    mov     rsi, qword ptr [rdx + 0x20]
0x1344c:    mov     rax, qword ptr [r15 + 0x20]
0x13450:    cmp     rsi, rax
0x13453:    jne     0x1316a
0x13459:    mov     byte ptr [rdx + 0x49], 0

```

```

0x1345d:    sub     dword ptr [r12 + 0x70], 1
0x13463:    jmp     0x13173
0x13468:    mov     r9, qword ptr [rsp + 0x18]
0x1346d:    mov     esi, dword ptr [rsp + 0x10]
0x13471:    xor     eax, eax
0x13473:    lea     rdx, [rsp + 0x28]
0x13478:    mov     r8d, r13d
0x1347b:    mov     rcx, rbp
0x1347e:    mov     rdi, r12
0x13481:    call    0x125c0
0x13486:    cmp     rax, -1
0x1348a:    je      0x13188
0x13490:    mov     rdx, qword ptr [r12 + 0x38]
0x13495:    add     rdx, rbx
0x13498:    cmp     qword ptr [rsp + 0x28], rax
0x1349d:    je      0x134b3
0x1349f:    add     r14, qword ptr [rbp + 0x38]
0x134a3:    mov     rsi, qword ptr [rdx + 0x20]
0x134a7:    mov     rax, qword ptr [r14 + 0x20]
0x134ab:    mov     r15, r14
0x134ae:    jmp     0x1316a
0x134b3:    cmp     qword ptr [rsp + 0x30], rax
0x134b8:    jne     0x1349f
0x134ba:    add     r14, qword ptr [rbp + 0x38]
0x134be:    mov     r15, r14
0x134c1:    jmp     0x13459
0x134c3:    call    0x49d0
0x134c8:    nop     dword ptr [rax + rax]
0x134d0:    endbr64
0x134d4:    push    r15
0x134d6:    movsxd  rsi, esi
0x134d9:    mov     r15d, edx
0x134dc:    push    r14
0x134de:    push    r13
0x134e0:    lea     r13, [rsi + rsi*4]
0x134e4:    push    r12
0x134e6:    shl     r13, 4
0x134ea:    mov     r12, rdi
0x134ed:    push    rbp
0x134ee:    push    rbx
0x134ef:    sub     rsp, 8
0x134f3:    mov     rax, qword ptr [rdi + 0x38]
0x134f7:    mov     r14, qword ptr [rax + r13]
0x134fb:    mov     rdi, r14
0x134fe:    call    0x1d7a0
0x13503:    mov     rdi, r14
0x13506:    mov     rbp, rax
0x13509:    call    0x49b0

```



```

0x1350e:      mov     rbx, rax
0x13511:      test    r15d, r15d
0x13514:      jne     0x13580
0x13516:      test    eax, eax
0x13518:      jle     0x13543
0x1351a:      call    0x4ef0
0x1351f:      lea     edi, [rbx - 1]
0x13522:      xor     edx, edx
0x13524:      nop     dword ptr [rax]
0x13528:      movsx   rsi, byte ptr [r14 + rdx]
0x1352d:      mov     rcx, qword ptr [rax]
0x13530:      mov     ecx, dword ptr [rcx + rsi*4]
0x13533:      mov     byte ptr [rbp + rdx], cl
0x13537:      mov     rcx, rdx
0x1353a:      add     rdx, 1
0x1353e:      cmp     rdi, rcx
0x13541:      jne     0x13528
0x13543:      mov     rsi, rbp
0x13546:      mov     rdi, r14
0x13549:      call    0x4b30
0x1354e:      test    eax, eax
0x13550:      jne     0x135ba
0x13552:      mov     rax, qword ptr [r12 + 0x38]
0x13557:      mov     byte ptr [rax + r13 + 0x49], 0
0x1355d:      mov     eax, 1
0x13562:      sub     dword ptr [r12 + 0x70], 1
0x13568:      add     rsp, 8
0x1356c:      pop     rbx
0x1356d:      pop     rbp
0x1356e:      pop     r12
0x13570:      pop     r13
0x13572:      pop     r14
0x13574:      pop     r15
0x13576:      ret
0x13577:      nop     word ptr [rax + rax]
0x13580:      test    eax, eax
0x13582:      jle     0x13543
0x13584:      call    0x47c0
0x13589:      lea     edi, [rbx - 1]
0x1358c:      xor     edx, edx
0x1358e:      nop
0x13590:      movsx   rsi, byte ptr [r14 + rdx]
0x13595:      mov     rcx, qword ptr [rax]
0x13598:      mov     ecx, dword ptr [rcx + rsi*4]
0x1359b:      mov     byte ptr [rbp + rdx], cl
0x1359f:      mov     rcx, rdx
0x135a2:      add     rdx, 1
0x135a6:      cmp     rcx, rdi

```

```

0x135a9:    jne     0x13590
0x135ab:    mov     rsi, rbp
0x135ae:    mov     rdi, r14
0x135b1:    call    0x4b30
0x135b6:    test    eax, eax
0x135b8:    je      0x13552
0x135ba:    lea     edi, [rbx + 0x21]
0x135bd:    mov     ebx, 1
0x135c2:    movsxd  rdi, edi
0x135c5:    call    0x1d670
0x135ca:    mov     r8, r14
0x135cd:    mov     esi, 1
0x135d2:    lea     rcx, [rip + 0xdd5b]
0x135d9:    mov     r15, rax
0x135dc:    mov     rdi, rax
0x135df:    mov     rdx, -1
0x135e6:    xor     eax, eax
0x135e8:    call    0x4f10
0x135ed:    mov     edx, 1
0x135f2:    mov     esi, 1
0x135f7:    mov     rdi, r15
0x135fa:    call    0xe480
0x135ff:    xor     eax, eax
0x13601:    call    0xa820
0x13606:    mov     rdi, r15
0x13609:    call    0x18d10
0x1360e:    mov     rax, qword ptr [r12 + 0x38]
0x13613:    cmp     byte ptr [rax + r13 + 0x48], 0
0x13619:    je      0x13698
0x1361b:    mov     rsi, rbp
0x1361e:    mov     rdi, r14
0x13621:    call    0x4e10
0x13626:    test    eax, eax
0x13628:    je      0x137a5
0x1362e:    call    0x4860
0x13633:    mov     r15d, dword ptr [rax]
0x13636:    xor     eax, eax
0x13638:    call    0xe4e0
0x1363d:    xor     eax, eax
0x1363f:    call    0xa820
0x13644:    cmp     r15d, 1
0x13648:    sete    al
0x1364b:    cmp     r15d, 0xd
0x1364f:    sete    dl
0x13652:    or      al, dl
0x13654:    jne     0x13745
0x1365a:    and     ebx, 1
0x1365d:    je      0x13745

```

```

0x13663:    cmp     dword ptr [r12 + 0x90], 1
0x1366c:    je      0x13760
0x13672:    mov     rdi, rbp
0x13675:    call    0x4870
0x1367a:    mov     ebx, eax
0x1367c:    test    eax, eax
0x1367e:    je      0x1361b
0x13680:    add     rsp, 8
0x13684:    mov     eax, 5
0x13689:    pop     rbx
0x1368a:    pop     rbp
0x1368b:    pop     r12
0x1368d:    pop     r13
0x1368f:    pop     r14
0x13691:    pop     r15
0x13693:    ret
0x13694:    nop     dword ptr [rax]
0x13698:    xor     esi, esi
0x1369a:    mov     rdi, rbp
0x1369d:    call    0x4d90
0x136a2:    test    eax, eax
0x136a4:    jne     0x13732
0x136aa:    cmp     dword ptr [r12 + 0x90], 1
0x136b3:    je      0x1378b
0x136b9:    mov     rdi, rbp
0x136bc:    call    0x49b0
0x136c1:    lea     rdi, [rax + 0x21]
0x136c5:    call    0x1d670
0x136ca:    mov     rdx, -1
0x136d1:    mov     r8, rbp
0x136d4:    lea     rcx, [rip + 0xdc6c]
0x136db:    mov     r15, rax
0x136de:    mov     rdi, rax
0x136e1:    mov     esi, 1
0x136e6:    xor     eax, eax
0x136e8:    call    0x4f10
0x136ed:    mov     esi, 1
0x136f2:    mov     rdi, r15
0x136f5:    call    0x193f0
0x136fa:    mov     rdi, r15
0x136fd:    mov     ebx, eax
0x136ff:    call    0x18d10
0x13704:    mov     rdi, qword ptr [r12 + 0x18]
0x13709:    call    0xc4b0
0x1370e:    mov     edx, ebx
0x13710:    mov     eax, 7
0x13715:    and     edx, 0x7f
0x13718:    add     edx, 1

```

```

0x1371b:    cmp     dl, 1
0x1371e:    jg      0x13568
0x13724:    test    bl, 0x7f
0x13727:    jne     0x13680
0x1372d:    and     bh, 0xff
0x13730:    jne     0x1377a
0x13732:    mov     rsi, rbp
0x13735:    mov     rdi, r14
0x13738:    call    0x4dc0
0x1373d:    test    eax, eax
0x1373f:    je      0x13552
0x13745:    add     rsp, 8
0x13749:    mov     eax, 6
0x1374e:    pop     rbx
0x1374f:    pop     rbp
0x13750:    pop     r12
0x13752:    pop     r13
0x13754:    pop     r14
0x13756:    pop     r15
0x13758:    ret
0x13759:    nop     dword ptr [rax]
0x13760:    xor     eax, eax
0x13762:    mov     rsi, rbp
0x13765:    mov     rdi, r12
0x13768:    call    0x11220
0x1376d:    test    eax, eax
0x1376f:    je      0x13672
0x13775:    jmp     0x13568
0x1377a:    lea     rdi, [rip + 0xd469]
0x13781:    call    0x195e0
0x13786:    jmp     0x13680
0x1378b:    xor     eax, eax
0x1378d:    mov     rsi, rbp
0x13790:    mov     rdi, r12
0x13793:    call    0x11220
0x13798:    test    eax, eax
0x1379a:    je      0x136b9
0x137a0:    jmp     0x13568
0x137a5:    mov     rdi, r14
0x137a8:    call    0x4870
0x137ad:    test    eax, eax
0x137af:    je      0x13552
0x137b5:    mov     eax, 4
0x137ba:    jmp     0x13568
0x137bf:    nop
0x137c0:    endbr64
0x137c4:    push    r15
0x137c6:    push    r14

```

```

0x137c8:      mov     r14, rdi
0x137cb:      push    r13
0x137cd:      push    r12
0x137cf:      mov     r12, rdx
0x137d2:      push    rbp
0x137d3:      push    rbx
0x137d4:      sub     rsp, 0x118
0x137db:      mov     rax, qword ptr fs:[0x28]
0x137e4:      mov     qword ptr [rsp + 0x108], rax
0x137ec:      xor     eax, eax
0x137ee:      test    rdx, rdx
0x137f1:      je       0x13be4
0x137f7:      movsxd   rcx, dword ptr [r14 + 0x68]
0x137fb:      test    ecx, ecx
0x137fd:      jle     0x1382c
0x137ff:      lea     edx, [rcx - 1]
0x13802:      mov     rsi, qword ptr [r14 + 0x38]
0x13806:      lea     rdx, [rdx + rdx*4]
0x1380a:      shl     rdx, 4
0x1380e:      lea     rax, [rsi + 0x49]
0x13812:      lea     rdx, [rsi + rdx + 0x99]
0x1381a:      nop     word ptr [rax + rax]
0x13820:      mov     byte ptr [rax], 0
0x13823:      add     rax, 0x50
0x13827:      cmp     rax, rdx
0x1382a:      jne     0x13820
0x1382c:      mov     dword ptr [r14 + 0x70], 0
0x13834:      lea     rdi, [rcx*8]
0x1383c:      call    0x1d670
0x13841:      mov     qword ptr [rsp], rax
0x13845:      movsxd   rax, dword ptr [r14 + 0x68]
0x13849:      test    eax, eax
0x1384b:      jle     0x13c59
0x13851:      lea     rdi, [rax + rax*4]
0x13855:      xor     r8d, r8d
0x13858:      xor     eax, eax
0x1385a:      xor     r13d, r13d
0x1385d:      shl     rdi, 4
0x13861:      mov     r9d, 0x400
0x13867:      jmp     0x13879
0x13869:      nop     dword ptr [rax]
0x13870:      add     rax, 0x50
0x13874:      cmp     rdi, rax
0x13877:      je       0x138f8
0x13879:      mov     rcx, qword ptr [r14 + 0x38]
0x1387d:      add     rcx, rax
0x13880:      mov     edx, dword ptr [rcx + 0x10]
0x13883:      and     edx, 0xf000

```

```

0x13889:      cmp     edx, 0x8000
0x1388f:      jne     0x13870
0x13891:      mov     rdx, qword ptr [rcx + 8]
0x13895:      test    edx, 0x3ff
0x1389b:      je      0x138be
0x1389d:      mov     r10, rdx
0x138a0:      sar     r10, 0x3f
0x138a4:      shr     r10, 0x36
0x138a8:      lea     rsi, [rdx + r10]
0x138ac:      and     esi, 0x3ff
0x138b2:      sub     rsi, r10
0x138b5:      mov     r10, r9
0x138b8:      sub     r10, rsi
0x138bb:      add     rdx, r10
0x138be:      test    rdx, rdx
0x138c1:      lea     rsi, [rdx + 0x3ff]
0x138c8:      cmovs   rdx, rsi
0x138cc:      sar     rdx, 0xa
0x138d0:      cmp     r12, rdx
0x138d3:      jl      0x13af0
0x138d9:      mov     rbx, qword ptr [rsp]
0x138dd:      movsxd  rdx, r13d
0x138e0:      add     rax, 0x50
0x138e4:      add     r13d, 1
0x138e8:      mov     qword ptr [rbx + rdx*8], rcx
0x138ec:      cmp     rdi, rax
0x138ef:      jne     0x13879
0x138f1:      nop     dword ptr [rax]
0x138f8:      mov     rbx, qword ptr [rsp]
0x138fc:      movsxd  rsi, r13d
0x138ff:      mov     edx, 8
0x13904:      lea     rcx, [rip - 0x53cb]
0x1390b:      mov     dword ptr [rsp + 0x10], r8d
0x13910:      mov     rdi, rbx
0x13913:      call    0x48e0
0x13918:      test    r13d, r13d
0x1391b:      je      0x13c86
0x13921:      lea     eax, [r13 - 1]
0x13925:      mov     r8d, r13d
0x13928:      xor     r15d, r15d
0x1392b:      xor     r11d, r11d
0x1392e:      lea     r10, [rbx + rax*8 + 8]
0x13933:      mov     rax, r12
0x13936:      xor     r9d, r9d
0x13939:      mov     edx, 0x400
0x1393e:      mov     r13, r10
0x13941:      mov     r12, rbx
0x13944:      mov     rcx, rax

```

```

0x13947:      nop      word ptr [rax + rax]
0x13950:      mov      rdi, qword ptr [r12]
0x13954:      mov      rax, qword ptr [rdi + 8]
0x13958:      test     eax, 0x3ff
0x1395d:      je        0x13980
0x1395f:      mov      r10, rax
0x13962:      sar      r10, 0x3f
0x13966:      shr      r10, 0x36
0x1396a:      lea      rsi, [rax + r10]
0x1396e:      and      esi, 0x3ff
0x13974:      sub      rsi, r10
0x13977:      mov      r10, rdx
0x1397a:      sub      r10, rsi
0x1397d:      add      rax, r10
0x13980:      test     rax, rax
0x13983:      lea      rbp, [rax + 0x3ff]
0x1398a:      cmovns   rbp, rax
0x1398e:      sar      rbp, 0xa
0x13992:      test     r15d, r15d
0x13995:      jle      0x13af9
0x1399b:      mov      rax, r9
0x1399e:      xor      ebx, ebx
0x139a0:      jmp      0x139b8
0x139a2:      nop      word ptr [rax + rax]
0x139a8:      add      ebx, 1
0x139ab:      add      rax, 8
0x139af:      cmp      ebx, r15d
0x139b2:      je        0x13b10
0x139b8:      mov      rsi, qword ptr [rax]
0x139bb:      cmp      rsi, rbp
0x139be:      jl        0x139a8
0x139c0:      sub      rsi, rbp
0x139c3:      mov      qword ptr [rax], rsi
0x139c6:      test     ebx, ebx
0x139c8:      jne      0x13b00
0x139ce:      mov      byte ptr [rdi + 0x49], 1
0x139d2:      add      dword ptr [r14 + 0x70], 1
0x139d7:      add      r12, 8
0x139db:      cmp      r13, r12
0x139de:      jne      0x13950
0x139e4:      mov      r13d, r8d
0x139e7:      mov      r12, rcx
0x139ea:      xor      eax, eax
0x139ec:      mov      rdi, r14
0x139ef:      mov      qword ptr [rsp + 8], r9
0x139f4:      call     0x10ec0
0x139f9:      xor      eax, eax
0x139fb:      mov      rdi, r14

```

```

0x139fe:      call    0x10160
0x13a03:      cmp     byte ptr [r14 + 0x6d], 0
0x13a08:      mov     r9, qword ptr [rsp + 8]
0x13a0d:      je      0x13a1e
0x13a0f:      mov     rdi, r14
0x13a12:      xor     eax, eax
0x13a14:      call    0xec50
0x13a19:      mov     r9, qword ptr [rsp + 8]
0x13a1e:      xor     eax, eax
0x13a20:      mov     rdi, r14
0x13a23:      mov     qword ptr [rsp + 8], r9
0x13a28:      call    0x10240
0x13a2d:      cmp     byte ptr [r14 + 0x6d], 0
0x13a32:      mov     r9, qword ptr [rsp + 8]
0x13a37:      je      0x13a48
0x13a39:      mov     rdi, r14
0x13a3c:      xor     eax, eax
0x13a3e:      call    0x10d70
0x13a43:      mov     r9, qword ptr [rsp + 8]
0x13a48:      test    r9, r9
0x13a4b:      je      0x13a55
0x13a4d:      mov     rdi, r9
0x13a50:      call    0x18d10
0x13a55:      mov     rdi, qword ptr [rsp]
0x13a59:      call    0x18d10
0x13a5e:      test    r13d, r13d
0x13a61:      je      0x13bad
0x13a67:      mov     eax, dword ptr [rsp + 0x10]
0x13a6b:      lea     rbp, [rsp + 0x60]
0x13a70:      test    eax, eax
0x13a72:      je      0x13b7b
0x13a78:      sub     rsp, 8
0x13a7c:      mov     r9d, r13d
0x13a7f:      mov     edx, 0xa0
0x13a84:      mov     rdi, rbp
0x13a87:      push    rax
0x13a88:      lea     rax, [rip + 0xd8d2]
0x13a8f:      mov     esi, 1
0x13a94:      lea     r8, [rip + 0xd8ba]
0x13a9b:      push    r15
0x13a9d:      lea     rcx, [rip + 0xe104]
0x13aa4:      push    rax
0x13aa5:      xor     eax, eax
0x13aa7:      call    0x4f10
0x13aac:      add     rsp, 0x20
0x13ab0:      xor     esi, esi
0x13ab2:      mov     edx, 0x36
0x13ab7:      mov     rdi, rbp

```



```

0x13aba:    call    0x8070
0x13abf:    mov     rax, qword ptr [rsp + 0x108]
0x13ac7:    sub     rax, qword ptr fs:[0x28]
0x13ad0:    jne     0x13c91
0x13ad6:    add     rsp, 0x118
0x13add:    pop     rbx
0x13ade:    pop     rbp
0x13adf:    pop     r12
0x13ae1:    pop     r13
0x13ae3:    pop     r14
0x13ae5:    pop     r15
0x13ae7:    ret
0x13ae8:    nop     dword ptr [rax + rax]
0x13af0:    add     r8d, 1
0x13af4:    jmp     0x13870
0x13af9:    xor     ebx, ebx
0x13afb:    nop     dword ptr [rax + rax]
0x13b00:    cmp     r15d, ebx
0x13b03:    jne     0x139d7
0x13b09:    nop     dword ptr [rax]
0x13b10:    cmp     r11d, r15d
0x13b13:    je      0x13b40
0x13b15:    mov     rsi, rcx
0x13b18:    movsxd  rax, r15d
0x13b1b:    add     r15d, 1
0x13b1f:    sub     rsi, rbp
0x13b22:    mov     qword ptr [r9 + rax*8], rsi
0x13b26:    test    ebx, ebx
0x13b28:    jne     0x139d7
0x13b2e:    mov     rax, qword ptr [r12]
0x13b32:    mov     byte ptr [rax + 0x49], 1
0x13b36:    add     dword ptr [r14 + 0x70], 1
0x13b3b:    jmp     0x139d7
0x13b40:    add     r11d, 0x10
0x13b44:    mov     rdi, r9
0x13b47:    mov     qword ptr [rsp + 0x18], rcx
0x13b4c:    movsxd  rsi, r11d
0x13b4f:    mov     dword ptr [rsp + 0x14], r8d
0x13b54:    shl     rsi, 3
0x13b58:    mov     dword ptr [rsp + 8], r11d
0x13b5d:    call    0x1d690
0x13b62:    mov     rcx, qword ptr [rsp + 0x18]
0x13b67:    mov     r8d, dword ptr [rsp + 0x14]
0x13b6c:    mov     edx, 0x400
0x13b71:    mov     r11d, dword ptr [rsp + 8]
0x13b76:    mov     r9, rax
0x13b79:    jmp     0x13b15
0x13b7b:    sub     rsp, 8

```

```

0x13b7f:    lea     rcx, [rip + 0xe05a]
0x13b86:    mov     r9d, r13d
0x13b89:    xor     eax, eax
0x13b8b:    push    r15
0x13b8d:    mov     edx, 0xa0
0x13b92:    mov     esi, 1
0x13b97:    mov     rdi, rbp
0x13b9a:    lea     r8, [rip + 0xd7b4]
0x13ba1:    call    0x4f10
0x13ba6:    pop     rdx
0x13ba7:    pop     rcx
0x13ba8:    jmp     0x13ab0
0x13bad:    mov     eax, dword ptr [rsp + 0x10]
0x13bb1:    lea     rbp, [rsp + 0x60]
0x13bb6:    test    eax, eax
0x13bb8:    je      0x13c32
0x13bba:    mov     r9, r12
0x13bbd:    lea     r8, [rip + 0xd791]
0x13bc4:    mov     rdi, rbp
0x13bc7:    xor     eax, eax
0x13bc9:    lea     rcx, [rip + 0xe048]
0x13bd0:    mov     edx, 0xa0
0x13bd5:    mov     esi, 1
0x13bda:    call    0x4f10
0x13bdf:    jmp     0x13ab0
0x13be4:    mov     rdi, qword ptr [rsi + 0x18]
0x13be8:    lea     rdx, [rsp + 0x20]
0x13bed:    xor     esi, esi
0x13bef:    mov     qword ptr [rsp + 0x28], -1
0x13bf8:    call    0x1c490
0x13bfd:    test    eax, eax
0x13bff:    js      0x137f7
0x13c05:    cmp     qword ptr [rsp + 0x28], -1
0x13c0b:    je      0x137f7
0x13c11:    call    0x4a60
0x13c16:    mov     r12, qword ptr [rsp + 0x30]
0x13c1b:    test    eax, eax
0x13c1d:    cmovne  r12, qword ptr [rsp + 0x38]
0x13c23:    imul    r12, qword ptr [rsp + 0x20]
0x13c29:    shr     r12, 0xa
0x13c2d:    jmp     0x137f7
0x13c32:    mov     edx, 0xa0
0x13c37:    mov     esi, 1
0x13c3c:    mov     rdi, rbp
0x13c3f:    xor     eax, eax
0x13c41:    lea     r8, [rip + 0xd70d]
0x13c48:    lea     rcx, [rip + 0xd729]
0x13c4f:    call    0x4f10

```

```

0x13c54:      jmp      0x13ab0
0x13c59:      mov     rdi, qword ptr [rsp]
0x13c5d:      lea     rcx, [rip - 0x5724]
0x13c64:      xor     esi, esi
0x13c66:      xor     r13d, r13d
0x13c69:      mov     edx, 8
0x13c6e:      xor     r15d, r15d
0x13c71:      call    0x48e0
0x13c76:      mov     dword ptr [rsp + 0x10], 0
0x13c7e:      xor     r9d, r9d
0x13c81:      jmp     0x139ea
0x13c86:      xor     r15d, r15d
0x13c89:      xor     r9d, r9d
0x13c8c:      jmp     0x139ea
0x13c91:      call    0x49d0
0x13c96:      nop     word ptr cs:[rax + rax]
0x13ca0:      endbr64
0x13ca4:      push    r15
0x13ca6:      xor     r15d, r15d
0x13ca9:      push    r14
0x13cab:      push    r13
0x13cad:      push    r12
0x13caf:      xor     r12d, r12d
0x13cb2:      push    rbp
0x13cb3:      mov     rbp, rdi
0x13cb6:      push    rbx
0x13cb7:      sub     rsp, 0x18
0x13cbb:      call    0x49b0
0x13cc0:      lea     rdi, [rax + 1]
0x13cc4:      call    0x1d670
0x13cc9:      mov     esi, 8
0x13cce:      mov     edi, 2
0x13cd3:      mov     r13, rax
0x13cd6:      call    0x1d740
0x13cdb:      mov     r14, rax
0x13cde:      xor     eax, eax
0x13ce0:      movzx   ebx, byte ptr [rbp]
0x13ce4:      add     rbp, 1
0x13ce8:      cmp     bl, 0x5c
0x13ceb:      je      0x13d25
0x13ced:      jg      0x13cf9
0x13cef:      cmp     bl, 9
0x13cf2:      je      0x13d40
0x13cf4:      test    bl, 0xdf
0x13cf7:      je      0x13d40
0x13cf9:      test    r12d, r12d
0x13cfc:      je      0x13d0a
0x13cfe:      movsxd  rcx, eax

```

```

0x13d01:    add     eax, 1
0x13d04:    mov     byte ptr [r13 + rcx], 0x5c
0x13d0a:    movsxd  rcx, eax
0x13d0d:    add     rbp, 1
0x13d11:    add     eax, 1
0x13d14:    xor     r12d, r12d
0x13d17:    mov     byte ptr [r13 + rcx], bl
0x13d1c:    movzx   ebx, byte ptr [rbp - 1]
0x13d20:    cmp     bl, 0x5c
0x13d23:    jne     0x13ced
0x13d25:    test    r12d, r12d
0x13d28:    je      0x13d36
0x13d2a:    movsxd  rcx, eax
0x13d2d:    add     eax, 1
0x13d30:    mov     byte ptr [r13 + rcx], 0x5c
0x13d36:    xor     r12d, 1
0x13d3a:    jmp     0x13ce0
0x13d3c:    nop     dword ptr [rax]
0x13d40:    test    bl, bl
0x13d42:    je      0x13d60
0x13d44:    test    r12b, 1
0x13d48:    je      0x13d60
0x13d4a:    movsxd  rcx, eax
0x13d4d:    mov     r12d, 1
0x13d53:    add     eax, 1
0x13d56:    mov     byte ptr [r13 + rcx], bl
0x13d5b:    jmp     0x13ce0
0x13d5d:    nop     dword ptr [rax]
0x13d60:    test    eax, eax
0x13d62:    je      0x13da6
0x13d64:    cdqe
0x13d66:    movsxd  rdx, r15d
0x13d69:    mov     rdi, r13
0x13d6c:    mov     byte ptr [r13 + rax], 0
0x13d72:    lea     eax, [r15 + 1]
0x13d76:    lea     r15, [rdx*8]
0x13d7e:    lea     rcx, [r14 + r15]
0x13d82:    mov     dword ptr [rsp + 4], eax
0x13d86:    mov     qword ptr [rsp + 8], rcx
0x13d8b:    call    0x1d7a0
0x13d90:    mov     rcx, qword ptr [rsp + 8]
0x13d95:    mov     qword ptr [rcx], rax
0x13d98:    mov     qword ptr [r14 + r15 + 8], 0
0x13da1:    mov     r15d, dword ptr [rsp + 4]
0x13da6:    test    bl, bl
0x13da8:    je      0x13dd0
0x13daa:    lea     esi, [r15 + 2]
0x13dae:    mov     rdi, r14

```

```

0x13db1:      movsxd  rsi, esi
0x13db4:      shl     rsi, 3
0x13db8:      call    0x1d690
0x13dbd:      mov     r14, rax
0x13dc0:      xor     eax, eax
0x13dc2:      jmp     0x13ce0
0x13dc7:      nop     word ptr [rax + rax]
0x13dd0:      mov     rdi, r13
0x13dd3:      call    0x18d10
0x13dd8:      add     rsp, 0x18
0x13ddc:      mov     rax, r14
0x13ddf:      pop     rbx
0x13de0:      pop     rbp
0x13de1:      pop     r12
0x13de3:      pop     r13
0x13de5:      pop     r14
0x13de7:      pop     r15
0x13de9:      ret
0x13dea:      nop     word ptr [rax + rax]
0x13df0:      endbr64
0x13df4:      push    rbp
0x13df5:      mov     rbp, rdi
0x13df8:      push    rbx
0x13df9:      sub     rsp, 8
0x13dfd:      mov     rdi, qword ptr [rdi]
0x13e00:      test    rdi, rdi
0x13e03:      je      0x13e22
0x13e05:      mov     rbx, rbp
0x13e08:      nop     dword ptr [rax + rax]
0x13e10:      call    0x18d10
0x13e15:      mov     rdi, qword ptr [rbx + 8]
0x13e19:      add     rbx, 8
0x13e1d:      test    rdi, rdi
0x13e20:      jne     0x13e10
0x13e22:      add     rsp, 8
0x13e26:      mov     rdi, rbp
0x13e29:      pop     rbx
0x13e2a:      pop     rbp
0x13e2b:      jmp     0x18d10
0x13e30:      endbr64
0x13e34:      push    r15
0x13e36:      push    r14
0x13e38:      push    r13
0x13e3a:      push    r12
0x13e3c:      push    rbp
0x13e3d:      push    rbx
0x13e3e:      sub     rsp, 0x48
0x13e42:      mov     rax, qword ptr fs:[0x28]

```

```

0x13e4b:    mov     qword ptr [rsp + 0x38], rax
0x13e50:    xor     eax, eax
0x13e52:    cmp     esi, 0x33
0x13e55:    ja      0x14857
0x13e5b:    mov     r12, rdx
0x13e5e:    mov     eax, esi
0x13e60:    lea     rdx, [rip + 0xda9d]
0x13e67:    mov     rbp, rdi
0x13e6a:    movsxd  rax, dword ptr [rdx + rax*4]
0x13e6e:    mov     r15d, esi
0x13e71:    mov     r13, rcx
0x13e74:    add     rax, rdx
0x13e77:    jmp     rax
0x13e7a:    -----

0x13e82:    -----
0x13e83:    -----
0x13e84:    movsxd  rax, dword ptr [rbp + 0x28]
0x13e88:    mov     rdx, qword ptr [rbp + 0x18]
0x13e8c:    mov     dword ptr [rbp + 0x84], r15d
0x13e93:    mov     rdi, qword ptr [rbp + 0x38]
0x13e97:    mov     dword ptr [rip + 0x1778a], r15d
0x13e9e:    lea     rax, [rax + rax*4]
0x13ea2:    shl     rax, 4
0x13ea6:    cmp     byte ptr [rdx], 0x2f
0x13ea9:    mov     rbx, qword ptr [rdi + rax]
0x13ead:    mov     eax, dword ptr [rbp + 0x68]
0x13eb0:    jne     0x13ebc
0x13eb2:    cmp     byte ptr [rdx + 1], 0
0x13eb6:    je      0x1498b
0x13ebc:    sub     eax, 1
0x13ebf:    add     rdi, 0x50
0x13ec3:    mov     edx, 0x50
0x13ec8:    movsxd  rsi, eax
0x13ecb:    lea     rcx, [rip - 0x5852]
0x13ed2:    call    0x48e0
0x13ed7:    mov     ecx, dword ptr [rbp + 0x68]
0x13eda:    test    ecx, ecx
0x13edc:    jle     0x14cab
0x13ee2:    mov     rdx, qword ptr [rbp + 0x38]
0x13ee6:    xor     eax, eax
0x13ee8:    jmp     0x13efb
0x13eea:    nop     word ptr [rax + rax]
0x13ef0:    add     eax, 1
0x13ef3:    add     rdx, 0x50
0x13ef7:    cmp     eax, ecx

```

```

0x13ef9:      je      0x13f00
0x13efb:      cmp     qword ptr [rdx], rbx
0x13efe:      jne     0x13ef0
0x13f00:      mov     dword ptr [rbp + 0x28], eax
0x13f03:      mov     rdi, rbp
0x13f06:      xor     eax, eax
0x13f08:      call    0xf110
0x13f0d:      mov     dword ptr [rbp + 0x2c], eax
0x13f10:      jmp     0x13f40
0x13f12:      nop     word ptr [rax + rax]
0x13f18:      sub     r15d, 9
0x13f1c:      cmp     dword ptr [rbp + 0x64], 0x4f
0x13f20:      mov     dword ptr [rbp + 0x80], r15d
0x13f27:      jg      0x13f40
0x13f29:      cmp     r15d, 5
0x13f2d:      jne     0x13f40
0x13f2f:      mov     dword ptr [rbp + 0x80], 0
0x13f39:      nop     dword ptr [rax]
0x13f40:      cmp     byte ptr [rbp + 0x6d], 0
0x13f44:      je      0x13f50
0x13f46:      mov     rdi, rbp
0x13f49:      xor     eax, eax
0x13f4b:      call    0x10d70
0x13f50:      mov     rax, qword ptr [rsp + 0x38]
0x13f55:      sub     rax, qword ptr fs:[0x28]
0x13f5e:      jne     0x14ce9
0x13f64:      add     rsp, 0x48
0x13f68:      xor     eax, eax
0x13f6a:      pop     rbx
0x13f6b:      pop     rbp
0x13f6c:      pop     r12
0x13f6e:      pop     r13
0x13f70:      pop     r14
0x13f72:      pop     r15
0x13f74:      ret
0x13f75:      nop     dword ptr [rax]
0x13f78:      movsxd  rax, dword ptr [rdi + 0x28]
0x13f7c:      mov     esi, 0x2e
0x13f81:      lea     rax, [rax + rax*4]
0x13f85:      shl     rax, 4
0x13f89:      add     rax, qword ptr [rdi + 0x38]
0x13f8d:      mov     r12, qword ptr [rax]
0x13f90:      mov     rdi, r12
0x13f93:      call    0x4a20
0x13f98:      mov     rbx, rax
0x13f9b:      test    rax, rax
0x13f9e:      je      0x13f50
0x13fa0:      cmp     r12, rax

```

```

0x13fa3:    je     0x13f50
0x13fa5:    mov    rdi, rax
0x13fa8:    call   0x49b0
0x13fad:    lea     rdi, [rax + 2]
0x13fb1:    call   0x4c20
0x13fb6:    mov    r8, rbx
0x13fb9:    mov    esi, 1
0x13fbe:    xor     ebx, ebx
0x13fc0:    mov    rdi, rax
0x13fc3:    mov    r13, rax
0x13fc6:    xor     eax, eax
0x13fc8:    mov    rdx, -1
0x13fcf:    lea     rcx, [rip + 0xd3be]
0x13fd6:    call   0x4f10
0x13fdb:    mov     eax, dword ptr [rbp + 0x68]
0x13fde:    test    eax, eax
0x13fe0:    jle     0x14132
0x13fe6:    nop     word ptr cs:[rax + rax]
0x13ff0:    mov     rax, qword ptr [rbp + 0x38]
0x13ff4:    lea     r12, [rbx + rbx*4]
0x13ff8:    shl     r12, 4
0x13ffc:    add     rax, r12
0x13fff:    cmp     byte ptr [rax + 0x48], 0
0x14003:    je      0x14048
0x14005:    cmp     dword ptr [rip + 0x17618], 1
0x1400c:    mov     rsi, qword ptr [rax]
0x1400f:    mov     rdi, r13
0x14012:    sbb     edx, edx
0x14014:    and     edx, 4
0x14017:    add     edx, 1
0x1401a:    call   0x4a40
0x1401f:    test    eax, eax
0x14021:    jne     0x14048
0x14023:    add     r12, qword ptr [rbp + 0x38]
0x14027:    movzx   eax, byte ptr [r12 + 0x49]
0x1402d:    cmp     r15d, 0x32
0x14031:    je      0x149b3
0x14037:    test    al, al
0x14039:    je      0x14048
0x1403b:    mov     byte ptr [r12 + 0x49], 0
0x14041:    sub     dword ptr [rbp + 0x70], 1
0x14045:    nop     dword ptr [rax]
0x14048:    add     rbx, 1
0x1404c:    cmp     dword ptr [rbp + 0x68], ebx
0x1404f:    jg      0x13ff0
0x14051:    jmp     0x14132
0x14056:    nop     word ptr cs:[rax + rax]
0x14060:    xor     eax, eax

```



```

0x14062:    mov     rdi, rcx
0x14065:    call   0x13ca0
0x1406a:    cmp     qword ptr [rax], 0
0x1406e:    mov     r14, rax
0x14071:    mov     r13, rax
0x14074:    je      0x1412a
0x1407a:    nop     word ptr [rax + rax]
0x14080:    mov     edx, dword ptr [rbp + 0x68]
0x14083:    xor     ebx, ebx
0x14085:    test    edx, edx
0x14087:    jle     0x140f9
0x14089:    nop     dword ptr [rax]
0x14090:    mov     rax, qword ptr [rbp + 0x38]
0x14094:    lea     r12, [rbx + rbx*4]
0x14098:    shl     r12, 4
0x1409c:    add     rax, r12
0x1409f:    cmp     byte ptr [rax + 0x48], 0
0x140a3:    je      0x140f0
0x140a5:    cmp     dword ptr [rip + 0x17578], 1
0x140ac:    mov     rsi, qword ptr [rax]
0x140af:    sbb     edx, edx
0x140b1:    mov     rdi, qword ptr [r13]
0x140b5:    and     edx, 4
0x140b8:    add     edx, 1
0x140bb:    call   0x4a40
0x140c0:    test    eax, eax
0x140c2:    jne     0x140f0
0x140c4:    add     r12, qword ptr [rbp + 0x38]
0x140c8:    movzx   eax, byte ptr [r12 + 0x49]
0x140ce:    cmp     r15d, 0x24
0x140d2:    je      0x14840
0x140d8:    test    al, al
0x140da:    je      0x140f0
0x140dc:    mov     byte ptr [r12 + 0x49], 0
0x140e2:    sub     dword ptr [rbp + 0x70], 1
0x140e6:    nop     word ptr cs:[rax + rax]
0x140f0:    add     rbx, 1
0x140f4:    cmp     dword ptr [rbp + 0x68], ebx
0x140f7:    jg      0x14090
0x140f9:    add     r13, 8
0x140fd:    cmp     qword ptr [r13], 0
0x14102:    jne     0x14080
0x14108:    mov     rdi, qword ptr [r14]
0x1410b:    test    rdi, rdi
0x1410e:    je      0x1412a
0x14110:    mov     rbx, r14
0x14113:    nop     dword ptr [rax + rax]
0x14118:    call   0x18d10

```

```

0x1411d:    mov     rdi, qword ptr [rbx + 8]
0x14121:    add     rbx, 8
0x14125:    test    rdi, rdi
0x14128:    jne     0x14118
0x1412a:    mov     rdi, r14
0x1412d:    call    0x18d10
0x14132:    cmp     byte ptr [rbp + 0x6d], 0
0x14136:    je      0x13f50
0x1413c:    mov     rdi, rbp
0x1413f:    xor     eax, eax
0x14141:    call    0x10d70
0x14146:    cmp     byte ptr [rbp + 0x6d], 0
0x1414a:    je      0x13f50
0x14150:    mov     rdi, rbp
0x14153:    xor     eax, eax
0x14155:    call    0xec50
0x1415a:    jmp     0x13f50
0x1415f:    mov     rax, qword ptr [rdi + 0x18]
0x14163:    cmp     byte ptr [rax], 0x2f
0x14166:    je      0x149a4
0x1416c:    mov     esi, dword ptr [rbp + 0x28]
0x1416f:    test    esi, esi
0x14171:    je      0x14187
0x14173:    mov     dword ptr [rbp + 0x28], 0
0x1417a:    mov     rdi, rbp
0x1417d:    xor     eax, eax
0x1417f:    call    0xf110
0x14184:    mov     dword ptr [rbp + 0x2c], eax
0x14187:    mov     rax, qword ptr [rsp + 0x38]
0x1418c:    sub     rax, qword ptr fs:[0x28]
0x14195:    jne     0x14ce9
0x1419b:    add     rsp, 0x48
0x1419f:    mov     rsi, r12
0x141a2:    mov     rdi, rbp
0x141a5:    xor     eax, eax
0x141a7:    pop     rbx
0x141a8:    pop     rbp
0x141a9:    pop     r12
0x141ab:    pop     r13
0x141ad:    pop     r14
0x141af:    pop     r15
0x141b1:    jmp     0x162b0
0x141b6:    mov     edx, dword ptr [rdi + 0x68]
0x141b9:    test    edx, edx
0x141bb:    jle     0x141ec
0x141bd:    sub     edx, 1
0x141c0:    mov     rcx, qword ptr [rdi + 0x38]
0x141c4:    lea     rdx, [rdx + rdx*4]

```

```

0x141c8:      shl     rdx, 4
0x141cc:      lea     rax, [rcx + 0x49]
0x141d0:      lea     rdx, [rcx + rdx + 0x99]
0x141d8:      nop     dword ptr [rax + rax]
0x141e0:      mov     byte ptr [rax], 0
0x141e3:      add     rax, 0x50
0x141e7:      cmp     rdx, rax
0x141ea:      jne     0x141e0
0x141ec:      cmp     byte ptr [rbp + 0x6d], 0
0x141f0:      mov     dword ptr [rbp + 0x70], 0
0x141f7:      je      0x13f50
0x141fd:      jmp     0x1413c
0x14202:      movzx    edx, byte ptr [rdi + 0x6d]
0x14206:      mov     ebx, r8d
0x14209:      xor     r13d, r13d
0x1420c:      xor     r12d, r12d
0x1420f:      test    r8d, r8d
0x14212:      jne     0x14249
0x14214:      jmp     0x1488f
0x14219:      nop     dword ptr [rax]
0x14220:      test    r12d, r12d
0x14223:      jne     0x14240
0x14225:      mov     r12d, 1
0x1422b:      test    dl, dl
0x1422d:      je      0x14240
0x1422f:      mov     rdi, rbp
0x14232:      call    0x10760
0x14237:      movzx    edx, byte ptr [rbp + 0x6d]
0x1423b:      nop     dword ptr [rax + rax]
0x14240:      sub     ebx, 1
0x14243:      je      0x14870
0x14249:      mov     esi, dword ptr [rbp + 0x28]
0x1424c:      test    esi, esi
0x1424e:      je      0x14870
0x14254:      lea     eax, [rsi - 1]
0x14257:      mov     dword ptr [rbp + 0x28], eax
0x1425a:      cmp     esi, dword ptr [rbp + 0x2c]
0x1425d:      jne     0x14220
0x1425f:      sub     esi, dword ptr [rbp + 0x88]
0x14265:      mov     eax, 0
0x1426a:      mov     r13d, 1
0x14270:      cmovs   esi, eax
0x14273:      mov     dword ptr [rbp + 0x2c], esi
0x14276:      jmp     0x14240
0x14278:      mov     rsi, r12
0x1427b:      xor     eax, eax
0x1427d:      call    0x15bb0
0x14282:      jmp     0x13f50

```

```

0x14287:      mov     rsi, r12
0x1428a:      xor     eax, eax
0x1428c:      call    0x15910
0x14291:      jmp     0x13f50
0x14296:      mov     edx, dword ptr [rcx]
0x14298:      mov     rsi, r12
0x1429b:      xor     eax, eax
0x1429d:      call    0x167a0
0x142a2:      jmp     0x13f50
0x142a7:      mov     edx, dword ptr [rcx]
0x142a9:      mov     rsi, r12
0x142ac:      xor     eax, eax
0x142ae:      call    0x12fd0
0x142b3:      jmp     0x13f50
0x142b8:      mov     rdi, qword ptr [rdi + 0x48]
0x142bc:      call    0x19120
0x142c1:      mov     qword ptr [rbp + 0x48], 0
0x142c9:      jmp     0x13f50
0x142ce:      mov     eax, dword ptr [rdi + 0x30]
0x142d1:      mov     edx, dword ptr [rdi + 0x64]
0x142d4:      mov     esi, 4
0x142d9:      add     r8d, dword ptr [rdi + 0x74]
0x142dd:      add     eax, 1
0x142e0:      cmp     edx, 0x27
0x142e3:      jle     0x14300
0x142e5:      movsxd  rcx, dword ptr [rdi + 0x80]
0x142ec:      cmp     edx, 0x49
0x142ef:      jg      0x142f6
0x142f1:      cmp     ecx, 5
0x142f4:      je      0x14300
0x142f6:      lea     rsi, [rip + 0xd6e3]
0x142fd:      mov     esi, dword ptr [rsi + rcx*4]
0x14300:      sub     edx, esi
0x14302:      sub     eax, edx
0x14304:      mov     edx, 0
0x14309:      cmovs   eax, edx
0x1430c:      cmp     eax, r8d
0x1430f:      cmovg   eax, r8d
0x14313:      mov     dword ptr [rbp + 0x74], eax
0x14316:      mov     rdi, rbp
0x14319:      xor     eax, eax
0x1431b:      call    0x10ec0
0x14320:      xor     eax, eax
0x14322:      mov     rdi, rbp
0x14325:      call    0x10160
0x1432a:      cmp     byte ptr [rbp + 0x6d], 0
0x1432e:      je      0x1433a
0x14330:      mov     rdi, rbp

```

```

0x14333:    xor     eax, eax
0x14335:    call   0xec50
0x1433a:    mov     rdi, rbp
0x1433d:    xor     eax, eax
0x1433f:    call   0x10240
0x14344:    jmp     0x13f40
0x14349:    sub     dword ptr [rdi + 0x74], r8d
0x1434d:    mov     eax, dword ptr [rdi + 0x74]
0x14350:    xor     edx, edx
0x14352:    test    eax, eax
0x14354:    cmovs   eax, edx
0x14357:    jmp     0x14313
0x14359:    xor     esi, esi
0x1435b:    mov     rdi, rcx
0x1435e:    mov     edx, 0xa
0x14363:    call   0x4ba0
0x14368:    mov     rsi, r12
0x1436b:    mov     rdi, rbp
0x1436e:    movsxd  rdx, eax
0x14371:    xor     eax, eax
0x14373:    call   0x137c0
0x14378:    jmp     0x13f50
0x1437d:    mov     rsi, r12
0x14380:    xor     eax, eax
0x14382:    call   0x12ca0
0x14387:    jmp     0x13f50
0x1438c:    mov     eax, dword ptr [rdi + 0x60]
0x1438f:    mov     edx, dword ptr [r12 + 0x60]
0x14394:    mov     dword ptr [rdi + 0x60], edx
0x14397:    mov     edx, dword ptr [r12 + 0x64]
0x1439c:    mov     dword ptr [r12 + 0x60], eax
0x143a1:    mov     eax, dword ptr [rdi + 0x64]
0x143a4:    mov     dword ptr [rdi + 0x64], edx
0x143a7:    mov     edx, dword ptr [r12 + 0x58]
0x143ac:    mov     dword ptr [r12 + 0x64], eax
0x143b1:    mov     eax, dword ptr [rdi + 0x58]
0x143b4:    mov     dword ptr [rdi + 0x58], edx
0x143b7:    mov     edx, dword ptr [r12 + 0x5c]
0x143bc:    mov     dword ptr [r12 + 0x58], eax
0x143c1:    mov     eax, dword ptr [rdi + 0x5c]
0x143c4:    mov     dword ptr [rdi + 0x5c], edx
0x143c7:    mov     dword ptr [r12 + 0x5c], eax
0x143cc:    movsxd  rcx, dword ptr [rdi + 0x60]
0x143d0:    movsxd  rdx, dword ptr [rdi + 0x5c]
0x143d4:    movsxd  rsi, dword ptr [rdi + 0x58]
0x143d8:    movsxd  r8, dword ptr [rdi + 0x64]
0x143dc:    call   0xf590
0x143e1:    movsxd  rcx, dword ptr [r12 + 0x60]

```

```

0x143e6:      movsxd  rdx, dword ptr [r12 + 0x5c]
0x143eb:      mov     rdi, r12
0x143ee:      movsxd  rsi, dword ptr [r12 + 0x58]
0x143f3:      movsxd  r8, dword ptr [r12 + 0x64]
0x143f8:      call   0xf590
0x143fd:      jmp     0x13f50
0x14402:      mov     rsi, r12
0x14405:      xor     eax, eax
0x14407:      call   0x14e20
0x1440c:      jmp     0x13f50
0x14411:      mov     eax, dword ptr [rdi + 0x84]
0x14417:      add     eax, 1
0x1441a:      movsxd  r15, eax
0x1441d:      cdq
0x1441e:      imul    r15, r15, 0x38e38e39
0x14425:      sar     r15, 0x21
0x14429:      sub     r15d, edx
0x1442c:      lea     edx, [r15 + r15*8]
0x14430:      sub     eax, edx
0x14432:      mov     r15d, eax
0x14435:      jmp     0x13e84
0x1443a:      mov     eax, dword ptr [rdi + 0x80]
0x14440:      add     eax, 1
0x14443:      movsxd  r15, eax
0x14446:      cdq
0x14447:      imul    r15, r15, 0x2aaaaaab
0x1444e:      shr     r15, 0x20
0x14452:      sub     r15d, edx
0x14455:      lea     edx, [r15 + r15*2]
0x14459:      add     edx, edx
0x1445b:      sub     eax, edx
0x1445d:      mov     r15d, eax
0x14460:      jmp     0x13f1c
0x14465:      mov     edx, dword ptr [rdi + 0x68]
0x14468:      mov     dword ptr [rdi + 0x70], 0
0x1446f:      test    edx, edx
0x14471:      jle     0x14132
0x14477:      sub     edx, 1
0x1447a:      mov     rcx, qword ptr [rdi + 0x38]
0x1447e:      lea     rdx, [rdx + rdx*4]
0x14482:      shl     rdx, 4
0x14486:      lea     rax, [rcx + 0x49]
0x1448a:      lea     rcx, [rcx + rdx + 0x99]
0x14492:      nop     word ptr [rax + rax]
0x14498:      cmp     byte ptr [rax - 1], 0
0x1449c:      je      0x144ac
0x1449e:      cmp     byte ptr [rax], 0
0x144a1:      sete    dl

```

```

0x144a4:    mov     byte ptr [rax], dl
0x144a6:    movzx   edx, dl
0x144a9:    add     dword ptr [rbp + 0x70], edx
0x144ac:    add     rax, 0x50
0x144b0:    cmp     rcx, rax
0x144b3:    jne     0x14498
0x144b5:    jmp     0x14132
0x144ba:    mov     rsi, rcx
0x144bd:    xor     eax, eax
0x144bf:    call    0x161d0
0x144c4:    jmp     0x13f50
0x144c9:    mov     rdx, rcx
0x144cc:    mov     rsi, r12
0x144cf:    xor     eax, eax
0x144d1:    call    0x165f0
0x144d6:    jmp     0x13f50
0x144db:    mov     rcx, qword ptr [rcx + 8]
0x144df:    mov     rdi, rcx
0x144e2:    mov     qword ptr [rsp + 8], rcx
0x144e7:    call    0x49b0
0x144ec:    mov     rcx, qword ptr [rsp + 8]
0x144f1:    mov     qword ptr [rsp + 0x10], rax
0x144f6:    mov     r14, rax
0x144f9:    mov     eax, dword ptr [r13]
0x144fd:    cmp     eax, 2
0x14500:    je      0x14bb4
0x14506:    cmp     eax, 3
0x14509:    je      0x14aea
0x1450f:    cmp     eax, 1
0x14512:    je      0x148db
0x14518:    mov     rax, qword ptr [rbp + 0x40]
0x1451c:    mov     qword ptr [r13 + 0x10], rax
0x14520:    jmp     0x13f50
0x14525:    mov     rdi, qword ptr [rcx + 8]
0x14529:    mov     qword ptr [rsp + 8], rdi
0x1452e:    call    0x49b0
0x14533:    mov     rdi, qword ptr [rsp + 8]
0x14538:    mov     qword ptr [rsp + 0x10], rax
0x1453d:    mov     r14, rax
0x14540:    mov     eax, dword ptr [r13]
0x14544:    cmp     eax, 2
0x14547:    je      0x14a96
0x1454d:    cmp     eax, 3
0x14550:    je      0x149de
0x14556:    cmp     eax, 1
0x14559:    je      0x148db
0x1455f:    mov     rax, qword ptr [rbp + 0x40]
0x14563:    mov     qword ptr [rsp + 0x10], rax

```

```

0x14568:    mov     rax, qword ptr [rsp + 0x10]
0x1456d:    mov     qword ptr [r13 + 0x10], rax
0x14571:    jmp     0x13f50
0x14576:    mov     edi, 0x10
0x1457b:    call    0x19100
0x14580:    lea     rsi, [rsp + 0x20]
0x14585:    mov     qword ptr [rsp + 0x28], 0
0x1458e:    mov     qword ptr [rbp + 0x48], rax
0x14592:    mov     rdi, rax
0x14595:    mov     eax, dword ptr [rbp + 0x28]
0x14598:    mov     dword ptr [rsp + 0x20], eax
0x1459c:    call    0x19150
0x145a1:    mov     qword ptr [rbp + 0x40], 0
0x145a9:    mov     byte ptr [rbp + 0x6e], 0
0x145ad:    jmp     0x13f50
0x145b2:    xor     esi, esi
0x145b4:    mov     edx, 0xa
0x145b9:    mov     rdi, rcx
0x145bc:    call    0x4ba0
0x145c1:    test    eax, eax
0x145c3:    jle     0x13f50
0x145c9:    mov     ecx, dword ptr [rbp + 0x60]
0x145cc:    lea     edx, [rcx - 1]
0x145cf:    cmp     edx, eax
0x145d1:    jle     0x13f50
0x145d7:    mov     dword ptr [r12 + 0x88], eax
0x145df:    mov     dword ptr [rbp + 0x88], eax
0x145e5:    jmp     0x13f50
0x145ea:    mov     eax, dword ptr [rdi + 0x68]
0x145ed:    sub     eax, 1
0x145f0:    cmp     dword ptr [rdi + 0x28], eax
0x145f3:    je      0x13f50
0x145f9:    mov     dword ptr [rdi + 0x28], eax
0x145fc:    mov     rdi, rbp
0x145ff:    xor     eax, eax
0x14601:    call    0xf110
0x14606:    mov     dword ptr [rbp + 0x2c], eax
0x14609:    jmp     0x14132
0x1460e:    mov     ecx, dword ptr [rdi + 0x28]
0x14611:    test    ecx, ecx
0x14613:    je      0x13f50
0x14619:    mov     dword ptr [rdi + 0x28], 0
0x14620:    jmp     0x145fc
0x14622:    mov     r9d, dword ptr [rdi + 0x68]
0x14626:    mov     r10d, dword ptr [rdi + 0x28]
0x1462a:    lea     eax, [r9 - 1]
0x1462e:    cmp     r10d, eax
0x14631:    je      0x13f50

```



```

0x14637:      mov     r8d, dword ptr [rdi + 0x60]
0x1463b:      mov     ecx, eax
0x1463d:      sub     ecx, dword ptr [rdi + 0x2c]
0x14640:      lea     edx, [r8 - 2]
0x14644:      cmp     ecx, edx
0x14646:      jl      0x14656
0x14648:      mov     ecx, eax
0x1464a:      sub     ecx, r10d
0x1464d:      cmp     edx, ecx
0x1464f:      jg      0x14656
0x14651:      lea     eax, [r10 + r8 - 2]
0x14656:      mov     dword ptr [rbp + 0x28], eax
0x14659:      mov     rdi, rbp
0x1465c:      xor     eax, eax
0x1465e:      sub     r8d, 1
0x14662:      call    0xf110
0x14667:      mov     dword ptr [rbp + 0x2c], eax
0x1466a:      movzx   eax, byte ptr [rbp + 0x6d]
0x1466e:      cmp     r9d, r8d
0x14671:      jge     0x148ce
0x14677:      test    al, al
0x14679:      je      0x13f50
0x1467f:      mov     esi, r10d
0x14682:      mov     rdi, rbp
0x14685:      call    0x10760
0x1468a:      cmp     byte ptr [rbp + 0x6d], 0
0x1468e:      mov     esi, dword ptr [rbp + 0x28]
0x14691:      je      0x13f50
0x14697:      mov     rdi, rbp
0x1469a:      call    0x10760
0x1469f:      jmp     0x14146
0x146a4:      mov     r8d, dword ptr [rdi + 0x28]
0x146a8:      test    r8d, r8d
0x146ab:      je      0x13f50
0x146b1:      mov     r9d, dword ptr [rdi + 0x60]
0x146b5:      lea     eax, [r9 - 2]
0x146b9:      cmp     r8d, eax
0x146bc:      jge     0x149ca
0x146c2:      mov     dword ptr [rdi + 0x28], 0
0x146c9:      xor     eax, eax
0x146cb:      call    0xf110
0x146d0:      mov     dword ptr [rbp + 0x2c], eax
0x146d3:      sub     r9d, 1
0x146d7:      movzx   eax, byte ptr [rbp + 0x6d]
0x146db:      cmp     r9d, dword ptr [rbp + 0x68]
0x146df:      jle     0x148ce
0x146e5:      mov     esi, r8d
0x146e8:      test    al, al

```

```

0x146ea:      je      0x13f50
0x146f0:      jmp     0x14682
0x146f2:      movzx   edi, byte ptr [rdi + 0x6d]
0x146f6:      mov     ebx, r8d
0x146f9:      xor     r13d, r13d
0x146fc:      xor     r12d, r12d
0x146ff:      test    r8d, r8d
0x14702:      jne     0x14739
0x14704:      jmp     0x148c0
0x14709:      nop     dword ptr [rax]
0x14710:      test    r12d, r12d
0x14713:      jne     0x14730
0x14715:      mov     r12d, 1
0x1471b:      test    dil, dil
0x1471e:      je      0x14730
0x14720:      mov     rdi, rbp
0x14723:      mov     esi, r9d
0x14726:      call    0x10760
0x1472b:      movzx   edi, byte ptr [rbp + 0x6d]
0x1472f:      nop
0x14730:      sub     ebx, 1
0x14733:      je      0x148a0
0x14739:      mov     eax, dword ptr [rbp + 0x68]
0x1473c:      mov     r9d, dword ptr [rbp + 0x28]
0x14740:      sub     eax, 1
0x14743:      cmp     r9d, eax
0x14746:      jge     0x148a0
0x1474c:      mov     esi, dword ptr [rbp + 0x2c]
0x1474f:      mov     ecx, dword ptr [rbp + 0x60]
0x14752:      lea     edx, [r9 + 1]
0x14756:      mov     dword ptr [rbp + 0x28], edx
0x14759:      sub     ecx, 2
0x1475c:      sub     edx, esi
0x1475e:      cmp     edx, ecx
0x14760:      jl      0x14710
0x14762:      sub     eax, ecx
0x14764:      add     esi, dword ptr [rbp + 0x88]
0x1476a:      mov     r13d, 1
0x14770:      add     eax, 1
0x14773:      cmp     esi, eax
0x14775:      cmovg   esi, eax
0x14778:      mov     dword ptr [rbp + 0x2c], esi
0x1477b:      jmp     0x14730
0x1477d:      mov     rax, qword ptr [rdi + 0x18]
0x14781:      mov     esi, dword ptr [rdi + 0x28]
0x14784:      cmp     byte ptr [rax], 0x2f
0x14787:      je      0x1497c
0x1478d:      test    esi, esi

```

```

0x1478f:      je      0x147d0
0x14791:      movsxd  rax, esi
0x14794:      lea     rax, [rax + rax*4]
0x14798:      shl     rax, 4
0x1479c:      add     rax, qword ptr [rbp + 0x38]
0x147a0:      movzx   edx, byte ptr [rax + 0x49]
0x147a4:      test    dl, dl
0x147a6:      sete    byte ptr [rax + 0x49]
0x147aa:      cmp     dl, 1
0x147ad:      sbb     eax, eax
0x147af:      and     eax, 2
0x147b2:      sub     eax, 1
0x147b5:      add     dword ptr [rbp + 0x70], eax
0x147b8:      cmp     byte ptr [rbp + 0x6d], 0
0x147bc:      je      0x147d0
0x147be:      mov     rdi, rbp
0x147c1:      mov     dword ptr [rsp + 8], r8d
0x147c6:      call    0x10760
0x147cb:      mov     r8d, dword ptr [rsp + 8]
0x147d0:      xor     ecx, ecx
0x147d2:      mov     rdx, r12
0x147d5:      mov     esi, 0x1c
0x147da:      mov     rdi, rbp
0x147dd:      call    0x13e30
0x147e2:      jmp     0x13f50
0x147e7:      mov     rsi, r12
0x147ea:      xor     eax, eax
0x147ec:      call    0x15400
0x147f1:      jmp     0x13f50
0x147f6:      mov     edx, dword ptr [rdi + 0x68]
0x147f9:      test    edx, edx
0x147fb:      jle     0x14132
0x14801:      sub     edx, 1
0x14804:      mov     rcx, qword ptr [rdi + 0x38]
0x14808:      lea     rdx, [rdx + rdx*4]
0x1480c:      shl     rdx, 4
0x14810:      lea     rax, [rcx + 0x49]
0x14814:      lea     rdx, [rcx + rdx + 0x99]
0x1481c:      nop     dword ptr [rax]
0x14820:      cmp     byte ptr [rax], 0
0x14823:      jne     0x14832
0x14825:      cmp     byte ptr [rax - 1], 0
0x14829:      je      0x14832
0x1482b:      mov     byte ptr [rax], 1
0x1482e:      add     dword ptr [rbp + 0x70], 1
0x14832:      add     rax, 0x50
0x14836:      cmp     rax, rdx
0x14839:      jne     0x14820

```

```

0x1483b:      jmp      0x14132
0x14840:      test     al, al
0x14842:      jne      0x140f0
0x14848:      mov      byte ptr [r12 + 0x49], 1
0x1484e:      add      dword ptr [rbp + 0x70], 1
0x14852:      jmp      0x140f0
0x14857:      lea      rdi, [rip + 0xcb3a]
0x1485e:      call     0x8b00
0x14863:      jmp      0x13f50
0x14868:      nop      dword ptr [rax + rax]
0x14870:      test     r13d, r13d
0x14873:      je       0x14931
0x14879:      test     dl, dl
0x1487b:      je       0x13f50
0x14881:      mov      rdi, rbp
0x14884:      xor      eax, eax
0x14886:      call     0x10d70
0x1488b:      movzx    edx, byte ptr [rbp + 0x6d]
0x1488f:      test     dl, dl
0x14891:      jne      0x14150
0x14897:      jmp      0x13f50
0x1489c:      nop      dword ptr [rax]
0x148a0:      test     r13d, r13d
0x148a3:      je       0x14956
0x148a9:      test     dil, dil
0x148ac:      je       0x13f50
0x148b2:      mov      rdi, rbp
0x148b5:      xor      eax, eax
0x148b7:      call     0x10d70
0x148bc:      movzx    edi, byte ptr [rbp + 0x6d]
0x148c0:      test     dil, dil
0x148c3:      jne      0x14150
0x148c9:      jmp      0x13f50
0x148ce:      test     al, al
0x148d0:      je       0x13f50
0x148d6:      jmp      0x1413c
0x148db:      mov      rdi, qword ptr [rbp + 0x48]
0x148df:      lea      r14, [rsp + 0x20]
0x148e4:      mov      rsi, r14
0x148e7:      call     0x191a0
0x148ec:      mov      edx, dword ptr [rsp + 0x20]
0x148f0:      mov      rax, qword ptr [rbp + 0x40]
0x148f4:      cmp      rax, qword ptr [rsp + 0x10]
0x148f9:      jne      0x1451c
0x148ff:      mov      r8d, dword ptr [rbp + 0x28]
0x14903:      cmp      r8d, edx
0x14906:      jl       0x14cb2
0x1490c:      jg       0x14ccf

```

```

0x14912:    mov     rdi, qword ptr [rbp + 0x48]
0x14916:    mov     edx, 1
0x1491b:    mov     rsi, r14
0x1491e:    call    0x19200
0x14923:    mov     rax, qword ptr [rsp + 0x28]
0x14928:    mov     qword ptr [rbp + 0x40], rax
0x1492c:    jmp     0x1451c
0x14931:    test    r12d, r12d
0x14934:    je      0x1488f
0x1493a:    mov     esi, dword ptr [rbp + 0x28]
0x1493d:    test    dl, dl
0x1493f:    je      0x13f50
0x14945:    mov     rdi, rbp
0x14948:    call    0x10760
0x1494d:    movzx   edx, byte ptr [rbp + 0x6d]
0x14951:    jmp     0x1488f
0x14956:    test    r12d, r12d
0x14959:    je      0x148c0
0x1495f:    mov     esi, dword ptr [rbp + 0x28]
0x14962:    test    dil, dil
0x14965:    je      0x13f50
0x1496b:    mov     rdi, rbp
0x1496e:    call    0x10760
0x14973:    movzx   edi, byte ptr [rbp + 0x6d]
0x14977:    jmp     0x148c0
0x1497c:    cmp     byte ptr [rax + 1], 0
0x14980:    je      0x14791
0x14986:    jmp     0x1478d
0x1498b:    movsxd  rsi, eax
0x1498e:    lea     rcx, [rip - 0x6315]
0x14995:    mov     edx, 0x50
0x1499a:    call    0x48e0
0x1499f:    jmp     0x13ed7
0x149a4:    cmp     byte ptr [rax + 1], 0
0x149a8:    je      0x13f50
0x149ae:    jmp     0x1416c
0x149b3:    test    al, al
0x149b5:    jne     0x14048
0x149bb:    mov     byte ptr [r12 + 0x49], 1
0x149c1:    add     dword ptr [rbp + 0x70], 1
0x149c5:    jmp     0x14048
0x149ca:    mov     edx, r8d
0x149cd:    sub     edx, eax
0x149cf:    xor     eax, eax
0x149d1:    mov     dword ptr [rdi + 0x28], edx
0x149d4:    call    0xf110
0x149d9:    jmp     0x146d0
0x149de:    mov     eax, dword ptr [rbp + 0x28]

```

```

0x149e1:    mov     rdi, qword ptr [rbp + 0x48]
0x149e5:    lea     rsi, [rsp + 0x20]
0x149ea:    mov     qword ptr [rsp + 0x28], r14
0x149ef:    mov     dword ptr [rsp + 0x20], eax
0x149f3:    call    0x19150
0x149f8:    mov     rax, qword ptr [r13 + 8]
0x149fc:    mov     ebx, dword ptr [rbp + 0x28]
0x149ff:    mov     qword ptr [rsp + 8], rax
0x14a04:    test    ebx, ebx
0x14a06:    js      0x14a80
0x14a08:    movsxd  rax, ebx
0x14a0b:    lea     r8, [rax + rax*4]
0x14a0f:    shl     r8, 4
0x14a13:    add     r8, qword ptr [rbp + 0x38]
0x14a17:    mov     r15, r8
0x14a1a:    jmp     0x14a2c
0x14a1c:    nop     dword ptr [rax]
0x14a20:    sub     ebx, 1
0x14a23:    sub     r15, 0x50
0x14a27:    cmp     ebx, -1
0x14a2a:    je      0x14a80
0x14a2c:    mov     rsi, qword ptr [r15]
0x14a2f:    mov     rdi, qword ptr [rsp + 8]
0x14a34:    mov     rdx, r14
0x14a37:    call    0x4c30
0x14a3c:    test    eax, eax
0x14a3e:    jne     0x14a20
0x14a40:    cmp     byte ptr [rbp + 0x6e], 0
0x14a44:    mov     qword ptr [rbp + 0x40], r14
0x14a48:    je      0x14c8d
0x14a4e:    mov     dword ptr [rbp + 0x28], ebx
0x14a51:    xor     eax, eax
0x14a53:    mov     rdi, rbp
0x14a56:    call    0xf110
0x14a5b:    cmp     byte ptr [rbp + 0x6d], 0
0x14a5f:    mov     dword ptr [rbp + 0x2c], eax
0x14a62:    je      0x14a77
0x14a64:    xor     eax, eax
0x14a66:    mov     rdi, rbp
0x14a69:    call    0x10d70
0x14a6e:    mov     rax, qword ptr [rbp + 0x40]
0x14a72:    mov     qword ptr [rsp + 0x10], rax
0x14a77:    mov     byte ptr [rbp + 0x6e], 0
0x14a7b:    jmp     0x14568
0x14a80:    mov     rax, qword ptr [rbp + 0x40]
0x14a84:    mov     dword ptr [r13], 0xffffffff
0x14a8c:    mov     qword ptr [rsp + 0x10], rax
0x14a91:    jmp     0x14568

```

```

0x14a96:    cmp     byte ptr [rbp + 0x6e], 0
0x14a9a:    je      0x14c0e
0x14aa0:    mov     eax, dword ptr [rbp + 0x68]
0x14aa3:    lea     ebx, [rax - 1]
0x14aa6:    test    ebx, ebx
0x14aa8:    js      0x14a80
0x14aaa:    movsxd  rax, ebx
0x14aad:    lea     r15, [rax + rax*4]
0x14ab1:    shl     r15, 4
0x14ab5:    add     r15, qword ptr [rbp + 0x38]
0x14ab9:    jmp     0x14ad1
0x14abb:    nop     dword ptr [rax + rax]
0x14ac0:    sub     ebx, 1
0x14ac3:    sub     r15, 0x50
0x14ac7:    mov     rdi, qword ptr [rsp + 8]
0x14acc:    cmp     ebx, -1
0x14acf:    je      0x14a80
0x14ad1:    mov     rsi, qword ptr [r15]
0x14ad4:    mov     rdx, r14
0x14ad7:    mov     qword ptr [rsp + 8], rdi
0x14adc:    call    0x4c30
0x14ae1:    test    eax, eax
0x14ae3:    jne     0x14ac0
0x14ae5:    jmp     0x14a40
0x14aea:    mov     eax, dword ptr [rbp + 0x28]
0x14aed:    mov     rdi, qword ptr [rbp + 0x48]
0x14af1:    lea     rsi, [rsp + 0x20]
0x14af6:    mov     qword ptr [rsp + 0x28], r14
0x14afb:    mov     dword ptr [rsp + 0x20], eax
0x14aff:    call    0x19150
0x14b04:    mov     rax, qword ptr [r13 + 8]
0x14b08:    mov     ebx, dword ptr [rbp + 0x28]
0x14b0b:    mov     qword ptr [rsp + 8], rax
0x14b10:    mov     eax, dword ptr [rbp + 0x68]
0x14b13:    mov     dword ptr [rsp + 0x1c], eax
0x14b17:    cmp     ebx, eax
0x14b19:    jge     0x14ba3
0x14b1f:    movsxd  rax, ebx
0x14b22:    lea     rcx, [rax + rax*4]
0x14b26:    shl     rcx, 4
0x14b2a:    add     rcx, qword ptr [rbp + 0x38]
0x14b2e:    mov     r15, rcx
0x14b31:    jmp     0x14b45
0x14b33:    nop     dword ptr [rax + rax]
0x14b38:    add     ebx, 1
0x14b3b:    add     r15, 0x50
0x14b3f:    cmp     ebx, dword ptr [rsp + 0x1c]
0x14b43:    je      0x14ba3

```

```

0x14b45:    mov     rsi, qword ptr [r15]
0x14b48:    mov     rdi, qword ptr [rsp + 8]
0x14b4d:    mov     rdx, r14
0x14b50:    call    0x4c30
0x14b55:    test    eax, eax
0x14b57:    jne     0x14b38
0x14b59:    cmp     ebx, -1
0x14b5c:    je      0x14ba3
0x14b5e:    cmp     byte ptr [rbp + 0x6e], 0
0x14b62:    mov     qword ptr [rbp + 0x40], r14
0x14b66:    je      0x14c70
0x14b6c:    mov     dword ptr [rbp + 0x28], ebx
0x14b6f:    xor     eax, eax
0x14b71:    mov     rdi, rbp
0x14b74:    call    0xf110
0x14b79:    cmp     byte ptr [rbp + 0x6d], 0
0x14b7d:    mov     dword ptr [rbp + 0x2c], eax
0x14b80:    je      0x14b95
0x14b82:    xor     eax, eax
0x14b84:    mov     rdi, rbp
0x14b87:    call    0x10d70
0x14b8c:    mov     rax, qword ptr [rbp + 0x40]
0x14b90:    mov     qword ptr [rsp + 0x10], rax
0x14b95:    mov     byte ptr [rbp + 0x6e], 0
0x14b99:    mov     rax, qword ptr [rsp + 0x10]
0x14b9e:    jmp     0x1451c
0x14ba3:    mov     dword ptr [r13], 0xffffffff
0x14bab:    mov     rax, qword ptr [rbp + 0x40]
0x14baf:    jmp     0x1451c
0x14bb4:    xor     ebx, ebx
0x14bb6:    cmp     byte ptr [rbp + 0x6e], 0
0x14bba:    je      0x14c41
0x14bc0:    mov     eax, dword ptr [rbp + 0x68]
0x14bc3:    mov     dword ptr [rsp + 0x1c], eax
0x14bc7:    cmp     ebx, eax
0x14bc9:    jge     0x14ba3
0x14bcb:    movsxd  rax, ebx
0x14bce:    lea     r15, [rax + rax*4]
0x14bd2:    shl     r15, 4
0x14bd6:    add     r15, qword ptr [rbp + 0x38]
0x14bda:    jmp     0x14bf2
0x14bdc:    nop     dword ptr [rax]
0x14be0:    add     ebx, 1
0x14be3:    add     r15, 0x50
0x14be7:    cmp     ebx, dword ptr [rsp + 0x1c]
0x14beb:    mov     rcx, qword ptr [rsp + 8]
0x14bf0:    je      0x14ba3
0x14bf2:    mov     rsi, qword ptr [r15]

```



```

0x14bf5:    mov     rdx, r14
0x14bf8:    mov     rdi, rcx
0x14bfb:    mov     qword ptr [rsp + 8], rcx
0x14c00:    call    0x4c30
0x14c05:    test    eax, eax
0x14c07:    jne     0x14be0
0x14c09:    jmp     0x14b59
0x14c0e:    mov     eax, dword ptr [rbp + 0x28]
0x14c11:    mov     rdi, qword ptr [rbp + 0x48]
0x14c15:    lea     rsi, [rsp + 0x20]
0x14c1a:    mov     qword ptr [rsp + 0x28], r14
0x14c1f:    mov     dword ptr [rsp + 0x20], eax
0x14c23:    call    0x19150
0x14c28:    cmp     byte ptr [rbp + 0x6e], 0
0x14c2c:    jne     0x14cee
0x14c32:    mov     eax, dword ptr [rbp + 0x28]
0x14c35:    mov     rdi, qword ptr [r13 + 8]
0x14c39:    lea     ebx, [rax - 1]
0x14c3c:    jmp     0x14aa6
0x14c41:    mov     eax, dword ptr [rbp + 0x28]
0x14c44:    mov     rdi, qword ptr [rbp + 0x48]
0x14c48:    lea     rsi, [rsp + 0x20]
0x14c4d:    mov     qword ptr [rsp + 0x28], r14
0x14c52:    mov     dword ptr [rsp + 0x20], eax
0x14c56:    call    0x19150
0x14c5b:    cmp     byte ptr [rbp + 0x6e], 0
0x14c5f:    jne     0x14c67
0x14c61:    mov     eax, dword ptr [rbp + 0x28]
0x14c64:    lea     ebx, [rax + 1]
0x14c67:    mov     rcx, qword ptr [r13 + 8]
0x14c6b:    jmp     0x14bc0
0x14c70:    sub     ebx, dword ptr [rbp + 0x28]
0x14c73:    xor     ecx, ecx
0x14c75:    mov     rdx, r12
0x14c78:    mov     esi, 0x1c
0x14c7d:    mov     r8d, ebx
0x14c80:    mov     rdi, rbp
0x14c83:    call    0x13e30
0x14c88:    jmp     0x14518
0x14c8d:    mov     r8d, dword ptr [rbp + 0x28]
0x14c91:    xor     ecx, ecx
0x14c93:    mov     rdx, r12
0x14c96:    mov     esi, 0x1b
0x14c9b:    mov     rdi, rbp
0x14c9e:    sub     r8d, ebx
0x14ca1:    call    0x13e30
0x14ca6:    jmp     0x1455f
0x14cab:    xor     eax, eax

```

```

0x14cad:      jmp      0x13f00
0x14cb2:      sub      edx, r8d
0x14cb5:      xor      ecx, ecx
0x14cb7:      mov      esi, 0x1c
0x14cbc:      mov      rdi, rbp
0x14cbf:      mov      r8d, edx
0x14cc2:      mov      rdx, r12
0x14cc5:      call     0x13e30
0x14cca:      jmp      0x14912
0x14ccf:      sub      r8d, edx
0x14cd2:      xor      ecx, ecx
0x14cd4:      mov      rdx, r12
0x14cd7:      mov      esi, 0x1b
0x14cdc:      mov      rdi, rbp
0x14cdf:      call     0x13e30
0x14ce4:      jmp      0x14912
0x14ce9:      call     0x49d0
0x14cee:      mov      rdi, qword ptr [r13 + 8]
0x14cf2:      jmp      0x14aa0
0x14cf7:      nop      word ptr [rax + rax]
0x14d00:      endbr64
0x14d04:      push     rbp
0x14d05:      xor      eax, eax
0x14d07:      mov      rbp, rdi
0x14d0a:      mov      dword ptr [rdi + 0x28], 0
0x14d11:      call     0xf110
0x14d16:      mov      rsi, qword ptr [rbp + 0x18]
0x14d1a:      xor      edx, edx
0x14d1c:      mov      ecx, 0x1e
0x14d21:      mov      dword ptr [rbp + 0x2c], eax
0x14d24:      lea      rdi, [rip + 0xc677]
0x14d2b:      xor      eax, eax
0x14d2d:      call     0xf760
0x14d32:      mov      rax, qword ptr [rbp + 0x18]
0x14d36:      cmp      byte ptr [rax], 0x2f
0x14d39:      jne      0x14d51
0x14d3b:      cmp      byte ptr [rax + 1], 0
0x14d3f:      jne      0x14d51
0x14d41:      lea      rdi, [rip + 0xc672]
0x14d48:      call     0x8b00
0x14d4d:      mov      rax, qword ptr [rbp + 0x18]
0x14d51:      mov      edx, 0x2f
0x14d56:      mov      word ptr [rax], dx
0x14d59:      mov      rdi, qword ptr [rbp + 0x18]
0x14d5d:      mov      qword ptr [rbp + 0x78], 1
0x14d65:      call     0x49c0
0x14d6a:      mov      rdi, rbp
0x14d6d:      mov      r8d, 1

```

```

0x14d73:    xor     ecx, ecx
0x14d75:    xor     edx, edx
0x14d77:    mov     esi, 0x22
0x14d7c:    pop     rbp
0x14d7d:    jmp     0x13e30
0x14d82:    nop     word ptr cs:[rax + rax]
0x14d8d:    nop     dword ptr [rax]
0x14d90:    endbr64
0x14d94:    push    rbp
0x14d95:    cmp     byte ptr [rdi + 0x6d], 0
0x14d99:    mov     rbp, rdi
0x14d9c:    mov     eax, esi
0x14d9e:    mov     byte ptr [rdi + 0x6c], sil
0x14da2:    je      0x14db0
0x14da4:    mov     esi, dword ptr [rdi + 0x28]
0x14da7:    call    0x10760
0x14dac:    movzx   eax, byte ptr [rbp + 0x6c]
0x14db0:    test    al, al
0x14db2:    jne     0x14dc0
0x14db4:    pop     rbp
0x14db5:    ret
0x14db6:    nop     word ptr cs:[rax + rax]
0x14dc0:    mov     rdi, qword ptr [rbp + 0x18]
0x14dc4:    call    0x49c0
0x14dc9:    cmp     eax, -1
0x14dcc:    jne     0x14db4
0x14dce:    xor     eax, eax
0x14dd0:    mov     rdi, rbp
0x14dd3:    call    0x14d00
0x14dd8:    xor     eax, eax
0x14dda:    mov     rdi, rbp
0x14ddd:    call    0x10ec0
0x14de2:    xor     eax, eax
0x14de4:    mov     rdi, rbp
0x14de7:    call    0x10160
0x14dec:    cmp     byte ptr [rbp + 0x6d], 0
0x14df0:    jne     0x14e10
0x14df2:    xor     eax, eax
0x14df4:    mov     rdi, rbp
0x14df7:    call    0x10240
0x14dfc:    cmp     byte ptr [rbp + 0x6d], 0
0x14e00:    je      0x14db4
0x14e02:    mov     rdi, rbp
0x14e05:    xor     eax, eax
0x14e07:    pop     rbp
0x14e08:    jmp     0x10d70
0x14e0d:    nop     dword ptr [rax]
0x14e10:    mov     rdi, rbp

```

```

0x14e13:    xor     eax, eax
0x14e15:    call   0xec50
0x14e1a:    jmp     0x14df2
0x14e1c:    nop     dword ptr [rax]
0x14e20:    endbr64
0x14e24:    push    r15
0x14e26:    mov     r15, rsi
0x14e29:    push    r14
0x14e2b:    push    r13
0x14e2d:    push    r12
0x14e2f:    push    rbp
0x14e30:    mov     rbp, rdi
0x14e33:    push    rbx
0x14e34:    sub     rsp, 0x58
0x14e38:    mov     edx, dword ptr [rdi + 0x70]
0x14e3b:    mov     rax, qword ptr fs:[0x28]
0x14e44:    mov     qword ptr [rsp + 0x48], rax
0x14e49:    xor     eax, eax
0x14e4b:    mov     qword ptr [rsp + 0x38], 0
0x14e54:    mov     qword ptr [rsp + 0x40], 0
0x14e5d:    mov     dword ptr [rdi + 0x90], 1
0x14e67:    test    edx, edx
0x14e69:    jne     0x15138
0x14e6f:    movsxd  rax, dword ptr [rdi + 0x28]
0x14e73:    mov     rdx, rax
0x14e76:    lea     rax, [rax + rax*4]
0x14e7a:    shl     rax, 4
0x14e7e:    add     rax, qword ptr [rdi + 0x38]
0x14e82:    mov     r12, qword ptr [rax]
0x14e85:    test    edx, edx
0x14e87:    jne     0x14ea0
0x14e89:    mov     rax, qword ptr [rdi + 0x18]
0x14e8d:    cmp     byte ptr [rax], 0x2f
0x14e90:    jne     0x15110
0x14e96:    cmp     byte ptr [rax + 1], 0
0x14e9a:    jne     0x15110
0x14ea0:    mov     rdi, r12
0x14ea3:    call    0x49b0
0x14ea8:    lea     rdi, [rax + 0x11]
0x14eac:    call    0x1d670
0x14eb1:    mov     edx, 0xe
0x14eb6:    mov     rsi, r12
0x14eb9:    lea     rdi, [rip + 0x16720]
0x14ec0:    mov     r13, rax
0x14ec3:    call    0x4880
0x14ec8:    lea     rcx, [rip + 0xc500]
0x14ecf:    mov     esi, 1
0x14ed4:    mov     rdi, r13

```

```

0x14ed7:    mov     r8, rax
0x14eda:    mov     rdx, -1
0x14ee1:    xor     eax, eax
0x14ee3:    call    0x4f10
0x14ee8:    mov     rdi, r12
0x14eeb:    call    0x49b0
0x14ef0:    mov     rdi, qword ptr [r15 + 0x18]
0x14ef4:    mov     rbx, rax
0x14ef7:    call    0x49b0
0x14efc:    lea     rdi, [rbx + rax + 2]
0x14f01:    call    0x1d670
0x14f06:    mov     r8, qword ptr [r15 + 0x18]
0x14f0a:    mov     r9, r12
0x14f0d:    lea     rcx, [rip + 0xb7ed]
0x14f14:    mov     rdi, rax
0x14f17:    mov     r14, rax
0x14f1a:    mov     esi, 1
0x14f1f:    xor     eax, eax
0x14f21:    mov     rdx, -1
0x14f28:    call    0x4f10
0x14f2d:    mov     rdi, r13
0x14f30:    lea     rsi, [rsp + 0x40]
0x14f35:    mov     rdx, r14
0x14f38:    mov     rcx, qword ptr [rip + 0x166d9]
0x14f3f:    call    0x83e0
0x14f44:    mov     rdi, r13
0x14f47:    test    rax, rax
0x14f4a:    je      0x153c3
0x14f50:    call    0x18d10
0x14f55:    movsxd  rax, dword ptr [rbp + 0x28]
0x14f59:    lea     rax, [rax + rax*4]
0x14f5d:    shl     rax, 4
0x14f61:    add     rax, qword ptr [rbp + 0x38]
0x14f65:    mov     eax, dword ptr [rax + 0x10]
0x14f68:    and     eax, 0xf000
0x14f6d:    cmp     eax, 0x4000
0x14f72:    je      0x152f0
0x14f78:    mov     rdi, qword ptr [rip + 0x15509]
0x14f7f:    call    0xdb60
0x14f84:    xor     eax, eax
0x14f86:    call    0xa820
0x14f8b:    mov     rdi, qword ptr [rsp + 0x40]
0x14f90:    xor     eax, eax
0x14f92:    call    0x19940
0x14f97:    mov     rdi, qword ptr [rsp + 0x40]
0x14f9c:    mov     r13, rax
0x14f9f:    call    0x18d10
0x14fa4:    mov     rdi, r12

```

```

0x14fa7:    xor     eax, eax
0x14fa9:    mov     rsi, r13
0x14fac:    mov     qword ptr [rsp + 0x40], r13
0x14fb1:    call    0x11190
0x14fb6:    mov     rdi, r14
0x14fb9:    mov     ebx, eax
0x14fbb:    call    0x18d10
0x14fc0:    test    ebx, ebx
0x14fc2:    jne     0x153a0
0x14fc8:    call    0x4bf0
0x14fcd:    mov     ebx, eax
0x14fcf:    call    0x49e0
0x14fd4:    movsxd  rdx, dword ptr [rbp + 0x28]
0x14fd8:    mov     r9d, ebx
0x14fdb:    mov     rsi, r12
0x14fde:    mov     r8d, eax
0x14fe1:    mov     rdi, rbp
0x14fe4:    xor     eax, eax
0x14fe6:    lea     rdx, [rdx + rdx*4]
0x14fea:    shl     rdx, 4
0x14fee:    add     rdx, qword ptr [rbp + 0x38]
0x14ff2:    mov     ecx, dword ptr [rdx + 0x10]
0x14ff5:    mov     rdx, qword ptr [rsp + 0x40]
0x14ffa:    call    0x11330
0x14fff:    mov     rdi, qword ptr [rsp + 0x40]
0x15004:    mov     ebx, eax
0x15006:    call    0x18d10
0x1500b:    lea     eax, [rbx - 1]
0x1500e:    cmp     eax, 1
0x15011:    jbe     0x15039
0x15013:    cdqe
0x15015:    lea     rdx, [rip + 0x15404]
0x1501c:    xor     ecx, ecx
0x1501e:    mov     rsi, r12
0x15021:    mov     rdx, qword ptr [rdx + rax*8]
0x15025:    mov     r8d, 0x1e
0x1502b:    lea     rdi, [rip + 0xc277]
0x15032:    xor     eax, eax
0x15034:    call    0xf7e0
0x15039:    xor     eax, eax
0x1503b:    call    0xe4e0
0x15040:    mov     rdi, rbp
0x15043:    xor     eax, eax
0x15045:    call    0x10240
0x1504a:    xor     eax, eax
0x1504c:    mov     rdi, r15
0x1504f:    call    0x10240
0x15054:    xor     eax, eax

```

```

0x15056:      call    0xa820
0x1505b:      mov     rsi, qword ptr [r15 + 0x18]
0x1505f:      xor     eax, eax
0x15061:      mov     edx, 1
0x15066:      mov     rdi, r15
0x15069:      call    0xf880
0x1506e:      test    eax, eax
0x15070:      je      0x153e0
0x15076:      xor     eax, eax
0x15078:      mov     rdi, r15
0x1507b:      call    0x10ec0
0x15080:      xor     eax, eax
0x15082:      mov     rdi, r15
0x15085:      call    0x10160
0x1508a:      cmp     byte ptr [r15 + 0x6d], 0
0x1508f:      je      0x1509b
0x15091:      mov     rdi, r15
0x15094:      xor     eax, eax
0x15096:      call    0xec50
0x1509b:      xor     eax, eax
0x1509d:      mov     rdi, r15
0x150a0:      call    0x10240
0x150a5:      cmp     byte ptr [r15 + 0x6d], 0
0x150aa:      je      0x150b6
0x150ac:      mov     rdi, r15
0x150af:      xor     eax, eax
0x150b1:      call    0x10d70
0x150b6:      mov     rsi, qword ptr [rbp + 0x18]
0x150ba:      xor     eax, eax
0x150bc:      mov     edx, 1
0x150c1:      mov     rdi, rbp
0x150c4:      call    0xf880
0x150c9:      test    eax, eax
0x150cb:      je      0x153d0
0x150d1:      xor     eax, eax
0x150d3:      mov     rdi, rbp
0x150d6:      call    0x10ec0
0x150db:      xor     eax, eax
0x150dd:      mov     rdi, rbp
0x150e0:      call    0x10160
0x150e5:      cmp     byte ptr [rbp + 0x6d], 0
0x150e9:      je      0x150f5
0x150eb:      mov     rdi, rbp
0x150ee:      xor     eax, eax
0x150f0:      call    0xec50
0x150f5:      xor     eax, eax
0x150f7:      mov     rdi, rbp
0x150fa:      call    0x10240

```

```

0x150ff:    cmp     byte ptr [rbp + 0x6d], 0
0x15103:    je      0x15110
0x15105:    mov     rdi, rbp
0x15108:    xor     eax, eax
0x1510a:    call    0x10d70
0x1510f:    nop
0x15110:    mov     rax, qword ptr [rsp + 0x48]
0x15115:    sub     rax, qword ptr fs:[0x28]
0x1511e:    jne     0x153ed
0x15124:    add     rsp, 0x58
0x15128:    pop     rbx
0x15129:    pop     rbp
0x1512a:    pop     r12
0x1512c:    pop     r13
0x1512e:    pop     r14
0x15130:    pop     r15
0x15132:    ret
0x15133:    nop     dword ptr [rax + rax]
0x15138:    mov     rdx, qword ptr [rsi + 0x18]
0x1513c:    mov     rcx, qword ptr [rip + 0x164d5]
0x15143:    lea     rsi, [rsp + 0x38]
0x15148:    lea     rdi, [rip + 0xc28d]
0x1514f:    call    0x83e0
0x15154:    test    rax, rax
0x15157:    je      0x15110
0x15159:    mov     rdi, qword ptr [rbp + 0x18]
0x1515d:    mov     rsi, qword ptr [rsp + 0x38]
0x15162:    xor     eax, eax
0x15164:    call    0x11190
0x15169:    test    eax, eax
0x1516b:    jne     0x15308
0x15171:    mov     r12, qword ptr [rsp + 0x38]
0x15176:    mov     rdi, r12
0x15179:    call    0x49b0
0x1517e:    mov     rdi, r12
0x15181:    mov     rbx, rax
0x15184:    lea     rax, [rax + 1]
0x15188:    mov     qword ptr [rsp + 0x18], rax
0x1518d:    lea     rax, [rbx + 2]
0x15191:    mov     rsi, rax
0x15194:    mov     qword ptr [rsp + 0x10], rax
0x15199:    call    0x1d690
0x1519e:    mov     qword ptr [rsp + 0x38], rax
0x151a3:    mov     byte ptr [rax + rbx], 0x2f
0x151a7:    mov     rax, qword ptr [rsp + 0x38]
0x151ac:    mov     byte ptr [rax + rbx + 1], 0
0x151b1:    mov     eax, dword ptr [rbp + 0x70]
0x151b4:    mov     dword ptr [rbp + 0x50], 0xffffffff

```



```

0x151bb:    mov     dword ptr [rbp + 0x54], eax
0x151be:    mov     rdi, rbp
0x151c1:    call    0x100e0
0x151c6:    cmp     eax, -1
0x151c9:    je      0x152ce
0x151cf:    cdq     rax
0x151d1:    lea     rax, [rax + rax*4]
0x151d5:    shl     rax, 4
0x151d9:    mov     qword ptr [rsp + 8], rax
0x151de:    add     rax, qword ptr [rbp + 0x38]
0x151e2:    mov     r12, qword ptr [rax]
0x151e5:    mov     r14d, dword ptr [rax + 0x10]
0x151e9:    call    0x49e0
0x151ee:    mov     ebx, eax
0x151f0:    call    0x4bf0
0x151f5:    mov     rdi, r12
0x151f8:    mov     r13d, eax
0x151fb:    call    0x49b0
0x15200:    mov     rcx, qword ptr [rsp + 0x10]
0x15205:    mov     rdi, qword ptr [rsp + 0x38]
0x1520a:    lea     rsi, [rcx + rax]
0x1520e:    call    0x1d690
0x15213:    mov     rcx, qword ptr [rsp + 0x18]
0x15218:    mov     rsi, r12
0x1521b:    mov     qword ptr [rsp + 0x38], rax
0x15220:    lea     rdi, [rax + rcx]
0x15224:    call    0x48b0
0x15229:    mov     eax, dword ptr [rip + 0x16881]
0x1522f:    test    eax, eax
0x15231:    jne     0x15338
0x15237:    mov     rdi, qword ptr [rip + 0x15252]
0x1523e:    call    0xdb60
0x15243:    xor     eax, eax
0x15245:    call    0xa820
0x1524a:    mov     rdx, qword ptr [rsp + 0x38]
0x1524f:    xor     eax, eax
0x15251:    mov     r9d, r13d
0x15254:    mov     r8d, ebx
0x15257:    mov     ecx, r14d
0x1525a:    mov     rsi, r12
0x1525d:    mov     rdi, rbp
0x15260:    call    0x11330
0x15265:    cmp     eax, 1
0x15268:    je      0x15320
0x1526e:    cmp     eax, 2
0x15271:    je      0x152ce
0x15273:    cmp     eax, 3
0x15276:    je      0x151be

```

```

0x1527c:      sub     eax, 1
0x1527f:      lea     rbx, [rip + 0x1519a]
0x15286:      xor     ecx, ecx
0x15288:      mov     rsi, r12
0x1528b:      cdqe
0x1528d:      mov     r8d, 0xe
0x15293:      lea     rdi, [rip + 0xc00f]
0x1529a:      mov     rdx, qword ptr [rbx + rax*8]
0x1529e:      xor     eax, eax
0x152a0:      call    0xf7e0
0x152a5:      test    al, al
0x152a7:      je      0x152ce
0x152a9:      xor     eax, eax
0x152ab:      mov     rdi, rbp
0x152ae:      call    0x10240
0x152b3:      mov     rdi, r15
0x152b6:      xor     eax, eax
0x152b8:      call    0x10240
0x152bd:      mov     rdi, rbp
0x152c0:      call    0x100e0
0x152c5:      cmp     eax, -1
0x152c8:      jne     0x151cf
0x152ce:      mov     rdi, qword ptr [rsp + 0x38]
0x152d3:      test    rdi, rdi
0x152d6:      je      0x152dd
0x152d8:      call    0x18d10
0x152dd:      xor     eax, eax
0x152df:      call    0xe4e0
0x152e4:      xor     eax, eax
0x152e6:      call    0xa820
0x152eb:      jmp     0x1505b
0x152f0:      mov     rdi, qword ptr [rip + 0x151a1]
0x152f7:      call    0xdb60
0x152fc:      jmp     0x14f84
0x15301:      nop     dword ptr [rax]
0x15308:      mov     edx, 0xc
0x1530d:      xor     esi, esi
0x1530f:      lea     rdi, [rip + 0x14fea]
0x15316:      call    0x8070
0x1531b:      jmp     0x15110
0x15320:      mov     rax, qword ptr [rbp + 0x38]
0x15324:      mov     rbx, qword ptr [rsp + 8]
0x15329:      mov     byte ptr [rax + rbx + 0x49], 0
0x1532e:      jmp     0x152a9
0x15333:      nop     dword ptr [rax + rax]
0x15338:      mov     dword ptr [rip + 0x1676e], 0
0x15342:      xor     eax, eax
0x15344:      call    0xc8c0

```

```

0x15349:    mov     edx, 4
0x1534e:    lea     rsi, [rip + 0xb1dd]
0x15355:    lea     rdi, [rip + 0xbe9d]
0x1535c:    mov     qword ptr [rsp + 0x28], rax
0x15361:    call    0x8070
0x15366:    mov     rdi, qword ptr [rsp + 0x28]
0x1536b:    mov     byte ptr [rsp + 0x27], al
0x1536f:    call    0xc970
0x15374:    xor     eax, eax
0x15376:    call    0xd870
0x1537b:    xor     eax, eax
0x1537d:    call    0xd740
0x15382:    xor     eax, eax
0x15384:    call    0xa820
0x15389:    movzx   eax, byte ptr [rsp + 0x27]
0x1538e:    and     eax, 0xfffffffff
0x15391:    cmp     al, 0x4e
0x15393:    je      0x15237
0x15399:    jmp     0x152ce
0x1539e:    nop
0x153a0:    mov     rdx, qword ptr [rsp + 0x40]
0x153a5:    lea     rdi, [rip + 0xc8a4]
0x153ac:    xor     ecx, ecx
0x153ae:    xor     eax, eax
0x153b0:    mov     r8d, 0x1e
0x153b6:    mov     rsi, r12
0x153b9:    call    0xf7e0
0x153be:    mov     rdi, qword ptr [rsp + 0x40]
0x153c3:    call    0x18d10
0x153c8:    jmp     0x15110
0x153cd:    nop     dword ptr [rax]
0x153d0:    mov     rdi, rbp
0x153d3:    call    0x14d00
0x153d8:    jmp     0x150d1
0x153dd:    nop     dword ptr [rax]
0x153e0:    mov     rdi, r15
0x153e3:    call    0x14d00
0x153e8:    jmp     0x15076
0x153ed:    call    0x49d0
0x153f2:    nop     word ptr cs:[rax + rax]
0x153fd:    nop     dword ptr [rax]
0x15400:    endbr64
0x15404:    push    r15
0x15406:    push    r14
0x15408:    push    r13
0x1540a:    push    r12
0x1540c:    push    rbp
0x1540d:    mov     rbp, rdi

```

```

0x15410:    push    rbx
0x15411:    sub     rsp, 0x28
0x15415:    mov     qword ptr [rsp + 0x10], rsi
0x1541a:    mov     esi, dword ptr [rdi + 0x70]
0x1541d:    test    esi, esi
0x1541f:    jne     0x15450
0x15421:    mov     ecx, dword ptr [rdi + 0x28]
0x15424:    test    ecx, ecx
0x15426:    jne     0x15450
0x15428:    mov     rax, qword ptr [rdi + 0x18]
0x1542c:    cmp     byte ptr [rax], 0x2f
0x1542f:    je      0x15440
0x15431:    add     rsp, 0x28
0x15435:    pop     rbx
0x15436:    pop     rbp
0x15437:    pop     r12
0x15439:    pop     r13
0x1543b:    pop     r14
0x1543d:    pop     r15
0x1543f:    ret
0x15440:    cmp     byte ptr [rax + 1], 0
0x15444:    jne     0x15431
0x15446:    nop     word ptr cs:[rax + rax]
0x15450:    xor     edx, edx
0x15452:    lea     rsi, [rip + 0xb0d9]
0x15459:    lea     rdi, [rip + 0xbf97]
0x15460:    call    0x8070
0x15465:    cmp     al, 0x79
0x15467:    jne     0x15431
0x15469:    mov     edx, dword ptr [rbp + 0x68]
0x1546c:    test    edx, edx
0x1546e:    jle     0x158fc
0x15474:    mov     rax, qword ptr [rbp + 0x38]
0x15478:    xor     ecx, ecx
0x1547a:    add     rax, 0x49
0x1547e:    jmp     0x1548b
0x15480:    add     ecx, 1
0x15483:    add     rax, 0x50
0x15487:    cmp     ecx, edx
0x15489:    je      0x15490
0x1548b:    cmp     byte ptr [rax], 0
0x1548e:    je      0x15480
0x15490:    mov     dword ptr [rsp + 0x1c], ecx
0x15494:    mov     eax, dword ptr [rbp + 0x70]
0x15497:    mov     dword ptr [rbp + 0x50], 0xffffffff
0x1549e:    xor     r13d, r13d
0x154a1:    lea     r15, [rip + 0xbf69]
0x154a8:    mov     dword ptr [rsp + 0x18], 1

```

```

0x154b0:    mov     dword ptr [rbp + 0x54], eax
0x154b3:    nop     dword ptr [rax + rax]
0x154b8:    mov     rdi, rbp
0x154bb:    call    0x100e0
0x154c0:    cmp     eax, -1
0x154c3:    je      0x155c6
0x154c9:    cdqe
0x154cb:    lea     rbx, [rax + rax*4]
0x154cf:    mov     rax, qword ptr [rbp + 0x38]
0x154d3:    shl     rbx, 4
0x154d7:    mov     r12, qword ptr [rax + rbx]
0x154db:    xor     eax, eax
0x154dd:    call    0x18ab0
0x154e2:    mov     rdi, r12
0x154e5:    call    0x49b0
0x154ea:    lea     rdi, [rax + 0x20]
0x154ee:    call    0x1d670
0x154f3:    mov     r8, r12
0x154f6:    mov     rcx, r15
0x154f9:    mov     esi, 1
0x154fe:    mov     r14, rax
0x15501:    mov     rdi, rax
0x15504:    mov     rdx, -1
0x1550b:    xor     eax, eax
0x1550d:    call    0x4f10
0x15512:    mov     edx, 1
0x15517:    mov     esi, 2
0x1551c:    mov     rdi, r14
0x1551f:    call    0xe480
0x15524:    xor     eax, eax
0x15526:    call    0xa820
0x1552b:    mov     rdi, r14
0x1552e:    call    0x18d10
0x15533:    mov     edx, dword ptr [rip + 0x16577]
0x15539:    test    edx, edx
0x1553b:    jne     0x15760
0x15541:    mov     eax, dword ptr [rsp + 0x18]
0x15545:    test    eax, eax
0x15547:    jne     0x15738
0x1554d:    mov     rdi, qword ptr [rip + 0x14f54]
0x15554:    call    0xdb60
0x15559:    xor     eax, eax
0x1555b:    call    0xa820
0x15560:    cmp     r13d, 0x61
0x15564:    je      0x15728
0x1556a:    cmp     r13d, 0x73
0x1556e:    je      0x154b8
0x15574:    cmp     r13d, 0x79

```

```

0x15578:    jne     0x155c6
0x1557a:    mov     rax, qword ptr [rbp + 0x38]
0x1557e:    cmp     byte ptr [rax + rbx + 0x48], 0
0x15583:    jne     0x156e8
0x15589:    mov     rdi, qword ptr [rip + 0x14f10]
0x15590:    call    0xdb60
0x15595:    xor     eax, eax
0x15597:    call    0xa820
0x1559c:    mov     rdi, r12
0x1559f:    call    0x4990
0x155a4:    test    eax, eax
0x155a6:    jne     0x157c8
0x155ac:    mov     rax, qword ptr [rbp + 0x38]
0x155b0:    mov     rdi, rbp
0x155b3:    mov     byte ptr [rax + rbx + 0x49], 0
0x155b8:    call    0x100e0
0x155bd:    cmp     eax, -1
0x155c0:    jne     0x154c9
0x155c6:    mov     eax, dword ptr [rsp + 0x1c]
0x155ca:    cmp     dword ptr [rbp + 0x68], eax
0x155cd:    je      0x155e3
0x155cf:    mov     eax, dword ptr [rsp + 0x1c]
0x155d3:    mov     rdi, rbp
0x155d6:    mov     dword ptr [rbp + 0x28], eax
0x155d9:    xor     eax, eax
0x155db:    call    0xf110
0x155e0:    mov     dword ptr [rbp + 0x2c], eax
0x155e3:    xor     eax, eax
0x155e5:    mov     rdi, rbp
0x155e8:    call    0x10240
0x155ed:    mov     rdi, qword ptr [rsp + 0x10]
0x155f2:    xor     eax, eax
0x155f4:    call    0x10240
0x155f9:    xor     eax, eax
0x155fb:    call    0xe4e0
0x15600:    xor     eax, eax
0x15602:    call    0xa820
0x15607:    mov     rsi, qword ptr [rbp + 0x18]
0x1560b:    xor     eax, eax
0x1560d:    mov     edx, 1
0x15612:    mov     rdi, rbp
0x15615:    call    0xf880
0x1561a:    test    eax, eax
0x1561c:    je      0x158e0
0x15622:    xor     eax, eax
0x15624:    mov     rdi, rbp
0x15627:    call    0x10ec0
0x1562c:    xor     eax, eax

```

```

0x1562e:    mov     rdi, rbp
0x15631:    call   0x10160
0x15636:    cmp     byte ptr [rbp + 0x6d], 0
0x1563a:    jne     0x15890
0x15640:    xor     eax, eax
0x15642:    mov     rdi, rbp
0x15645:    call   0x10240
0x1564a:    cmp     byte ptr [rbp + 0x6d], 0
0x1564e:    jne     0x15880
0x15654:    mov     rax, qword ptr [rsp + 0x10]
0x15659:    mov     rdi, qword ptr [rbp + 0x18]
0x1565d:    mov     r12, qword ptr [rax + 0x18]
0x15661:    mov     rsi, r12
0x15664:    call   0x4b30
0x15669:    test    eax, eax
0x1566b:    jne     0x15431
0x15671:    mov     rdi, qword ptr [rsp + 0x10]
0x15676:    mov     edx, 1
0x1567b:    mov     rsi, r12
0x1567e:    call   0xf880
0x15683:    test    eax, eax
0x15685:    je      0x158ed
0x1568b:    mov     rbx, qword ptr [rsp + 0x10]
0x15690:    xor     eax, eax
0x15692:    mov     rdi, rbx
0x15695:    call   0x10ec0
0x1569a:    xor     eax, eax
0x1569c:    mov     rdi, rbx
0x1569f:    call   0x10160
0x156a4:    cmp     byte ptr [rbx + 0x6d], 0
0x156a8:    je      0x156b4
0x156aa:    mov     rdi, rbx
0x156ad:    xor     eax, eax
0x156af:    call   0xec50
0x156b4:    mov     rbx, qword ptr [rsp + 0x10]
0x156b9:    xor     eax, eax
0x156bb:    mov     rdi, rbx
0x156be:    call   0x10240
0x156c3:    cmp     byte ptr [rbx + 0x6d], 0
0x156c7:    je      0x15431
0x156cd:    add     rsp, 0x28
0x156d1:    mov     rdi, rbx
0x156d4:    xor     eax, eax
0x156d6:    pop     rbx
0x156d7:    pop     rbp
0x156d8:    pop     r12
0x156da:    pop     r13
0x156dc:    pop     r14

```

```

0x156de:    pop     r15
0x156e0:    jmp     0x10d70
0x156e5:    nop     dword ptr [rax]
0x156e8:    mov     rdi, r12
0x156eb:    call    0x4870
0x156f0:    test    eax, eax
0x156f2:    je      0x155ac
0x156f8:    xor     eax, eax
0x156fa:    mov     ecx, 0xe
0x156ff:    mov     rsi, r12
0x15702:    lea     rdx, [rip + 0xae29]
0x15709:    lea     rdi, [rip + 0xc5e8]
0x15710:    call    0xf760
0x15715:    cmp     al, 0x79
0x15717:    je      0x154b8
0x1571d:    jmp     0x155c6
0x15722:    nop     word ptr [rax + rax]
0x15728:    mov     dword ptr [rsp + 0x18], 0
0x15730:    jmp     0x1557a
0x15735:    nop     dword ptr [rax]
0x15738:    mov     ecx, 2
0x1573d:    lea     rdx, [rip + 0xbcd9]
0x15744:    mov     rsi, r12
0x15747:    xor     eax, eax
0x15749:    lea     rdi, [rip + 0xc528]
0x15750:    call    0xf760
0x15755:    movsx   r13d, al
0x15759:    jmp     0x1554d
0x1575e:    nop
0x15760:    mov     dword ptr [rip + 0x16346], 0
0x1576a:    xor     eax, eax
0x1576c:    call    0xc8c0
0x15771:    mov     edx, 4
0x15776:    lea     rsi, [rip + 0xad5]
0x1577d:    lea     rdi, [rip + 0xba75]
0x15784:    mov     qword ptr [rsp + 8], rax
0x15789:    call    0x8070
0x1578e:    mov     rdi, qword ptr [rsp + 8]
0x15793:    mov     r14d, eax
0x15796:    call    0xc970
0x1579b:    xor     eax, eax
0x1579d:    and     r14d, 0xffffffff
0x157a1:    call    0xd870
0x157a6:    xor     eax, eax
0x157a8:    call    0xd740
0x157ad:    xor     eax, eax
0x157af:    call    0xa820
0x157b4:    cmp     r14b, 0x4e

```



```

0x157b8:      je      0x15541
0x157be:      jmp     0x155c6
0x157c3:      nop     dword ptr [rax + rax]
0x157c8:      xor     eax, eax
0x157ca:      mov     ecx, 0x12
0x157cf:      mov     rsi, r12
0x157d2:      lea     rdx, [rip + 0xad59]
0x157d9:      lea     rdi, [rip + 0xc4c0]
0x157e0:      call    0xf760
0x157e5:      cmp     al, 0x79
0x157e7:      jne     0x156f8
0x157ed:      mov     rdi, r12
0x157f0:      call    0x49b0
0x157f5:      lea     rdi, [rax + 0x21]
0x157f9:      call    0x1d670
0x157fe:      mov     rdx, -1
0x15805:      mov     r8, r12
0x15808:      lea     rcx, [rip + 0xbb38]
0x1580f:      mov     r14, rax
0x15812:      mov     rdi, rax
0x15815:      mov     esi, 1
0x1581a:      xor     eax, eax
0x1581c:      call    0x4f10
0x15821:      mov     esi, 1
0x15826:      mov     rdi, r14
0x15829:      call    0x193f0
0x1582e:      mov     rdi, r14
0x15831:      mov     dword ptr [rsp + 8], eax
0x15835:      call    0x18d10
0x1583a:      mov     rdi, qword ptr [rbp + 0x18]
0x1583e:      call    0xc4b0
0x15843:      mov     eax, dword ptr [rsp + 8]
0x15847:      mov     edx, eax
0x15849:      and     edx, 0x7f
0x1584c:      add     edx, 1
0x1584f:      cmp     dl, 1
0x15852:      jg      0x158b0
0x15854:      test    al, 0x7f
0x15856:      jne     0x156f8
0x1585c:      test    ah, 0xff
0x1585f:      je      0x155ac
0x15865:      lea     rdi, [rip + 0xb37e]
0x1586c:      call    0x195e0
0x15871:      jmp     0x156f8
0x15876:      nop     word ptr cs:[rax + rax]
0x15880:      mov     rdi, rbp
0x15883:      xor     eax, eax
0x15885:      call    0x10d70

```

```

0x1588a:      jmp      0x15654
0x1588f:      nop
0x15890:      xor      eax, eax
0x15892:      mov      rdi, rbp
0x15895:      call     0xec50
0x1589a:      xor      eax, eax
0x1589c:      mov      rdi, rbp
0x1589f:      call     0x10240
0x158a4:      cmp      byte ptr [rbp + 0x6d], 0
0x158a8:      je        0x15654
0x158ae:      jmp      0x15880
0x158b0:      xor      eax, eax
0x158b2:      mov      ecx, 0xe
0x158b7:      mov      rsi, r12
0x158ba:      lea      rdx, [rip + 0xac71]
0x158c1:      lea      rdi, [rip + 0xc408]
0x158c8:      call     0xf760
0x158cd:      cmp      al, 0x79
0x158cf:      je        0x154b8
0x158d5:      jmp      0x155c6
0x158da:      nop      word ptr [rax + rax]
0x158e0:      mov      rdi, rbp
0x158e3:      call     0x14d00
0x158e8:      jmp      0x15622
0x158ed:      mov      rdi, qword ptr [rsp + 0x10]
0x158f2:      call     0x14d00
0x158f7:      jmp      0x1568b
0x158fc:      mov      dword ptr [rsp + 0x1c], 0
0x15904:      jmp      0x15494
0x15909:      nop      dword ptr [rax]
0x15910:      endbr64
0x15914:      push     r13
0x15916:      xor      edx, edx
0x15918:      push     r12
0x1591a:      push     rbp
0x1591b:      mov      rbp, rsi
0x1591e:      push     rbx
0x1591f:      mov      rbx, rdi
0x15922:      lea      rdi, [rip + 0xbaf9]
0x15929:      sub      rsp, 0x18
0x1592d:      mov      rcx, qword ptr [rip + 0x15cd4]
0x15934:      mov      rax, qword ptr fs:[0x28]
0x1593d:      mov      qword ptr [rsp + 8], rax
0x15942:      xor      eax, eax
0x15944:      mov      rsi, rsp
0x15947:      mov      qword ptr [rsp], 0
0x1594f:      call     0x83e0
0x15954:      test     rax, rax

```

```

0x15957:      je      0x15a0b
0x1595d:      mov     rdi, qword ptr [rsp]
0x15961:      cmp     byte ptr [rdi], 0
0x15964:      je      0x15a06
0x1596a:      xor     eax, eax
0x1596c:      call    0x19940
0x15971:      mov     rdi, qword ptr [rsp]
0x15975:      mov     r13, rax
0x15978:      call    0x18d10
0x1597d:      mov     rdi, r13
0x15980:      mov     qword ptr [rsp], r13
0x15984:      call    0x49b0
0x15989:      mov     rdi, r13
0x1598c:      lea     rsi, [rax + 2]
0x15990:      mov     r12, rax
0x15993:      call    0x4ce0
0x15998:      mov     esi, 0x41ff
0x1599d:      mov     qword ptr [rsp], rax
0x159a1:      mov     byte ptr [rax + r12], 0x2f
0x159a6:      mov     rax, qword ptr [rsp]
0x159aa:      mov     byte ptr [rax + r12 + 1], 0
0x159b0:      mov     rdi, qword ptr [rsp]
0x159b4:      xor     eax, eax
0x159b6:      call    0x124d0
0x159bb:      cmp     eax, -1
0x159be:      je      0x15a90
0x159c4:      mov     rsi, qword ptr [rbx + 0x18]
0x159c8:      xor     eax, eax
0x159ca:      mov     edx, 1
0x159cf:      mov     rdi, rbx
0x159d2:      call    0xf880
0x159d7:      test    eax, eax
0x159d9:      jne     0x15a30
0x159db:      mov     rdi, rbx
0x159de:      call    0x14d00
0x159e3:      mov     rsi, qword ptr [rbp + 0x18]
0x159e7:      mov     rdi, qword ptr [rbx + 0x18]
0x159eb:      call    0x4b30
0x159f0:      test    eax, eax
0x159f2:      je      0x15af0
0x159f8:      mov     rdi, rbp
0x159fb:      xor     eax, eax
0x159fd:      call    0x10240
0x15a02:      mov     rdi, qword ptr [rsp]
0x15a06:      call    0x18d10
0x15a0b:      mov     rax, qword ptr [rsp + 8]
0x15a10:      sub     rax, qword ptr fs:[0x28]
0x15a19:      jne     0x15ba4

```

```

0x15a1f:      add     rsp, 0x18
0x15a23:      pop     rbx
0x15a24:      pop     rbp
0x15a25:      pop     r12
0x15a27:      pop     r13
0x15a29:      ret
0x15a2a:      nop     word ptr [rax + rax]
0x15a30:      mov     rsi, qword ptr [rsp]
0x15a34:      mov     rdi, rbx
0x15a37:      xor     eax, eax
0x15a39:      call    0x122d0
0x15a3e:      mov     edx, dword ptr [rbx + 0x60]
0x15a41:      mov     esi, dword ptr [rbx + 0x68]
0x15a44:      mov     edi, 0
0x15a49:      mov     dword ptr [rbx + 0x28], eax
0x15a4c:      sub     edx, 2
0x15a4f:      mov     ecx, edx
0x15a51:      shr     ecx, 0x1f
0x15a54:      add     ecx, edx
0x15a56:      sar     ecx, 1
0x15a58:      sub     eax, ecx
0x15a5a:      lea     ecx, [rdx + rax]
0x15a5d:      cmovs   eax, edi
0x15a60:      cmp     esi, ecx
0x15a62:      jg      0x15a70
0x15a64:      mov     eax, esi
0x15a66:      sub     eax, edx
0x15a68:      mov     edx, 0
0x15a6d:      cmovs   eax, edx
0x15a70:      cmp     byte ptr [rbx + 0x6d], 0
0x15a74:      mov     dword ptr [rbx + 0x2c], eax
0x15a77:      jne     0x15ac0
0x15a79:      mov     rdi, rbx
0x15a7c:      xor     eax, eax
0x15a7e:      call    0x10240
0x15a83:      jmp     0x159e3
0x15a88:      nop     dword ptr [rax + rax]
0x15a90:      mov     rsi, qword ptr [rsp]
0x15a94:      mov     ecx, 0x1c
0x15a99:      xor     edx, edx
0x15a9b:      xor     eax, eax
0x15a9d:      lea     rdi, [rip + 0xb993]
0x15aa4:      call    0xf760
0x15aa9:      mov     rdi, qword ptr [rsp]
0x15aad:      call    0x18d10
0x15ab2:      jmp     0x15a0b
0x15ab7:      nop     word ptr [rax + rax]
0x15ac0:      xor     eax, eax

```

```

0x15ac2:    mov     rdi, rbx
0x15ac5:    call    0x10d70
0x15aca:    cmp     byte ptr [rbx + 0x6d], 0
0x15ace:    je      0x15a79
0x15ad0:    mov     rdi, rbx
0x15ad3:    xor     eax, eax
0x15ad5:    call    0xec50
0x15ada:    mov     rdi, rbx
0x15add:    xor     eax, eax
0x15adf:    call    0x10240
0x15ae4:    jmp     0x159e3
0x15ae9:    nop     dword ptr [rax]
0x15af0:    movsxd  rax, dword ptr [rbp + 0x28]
0x15af4:    lea     rax, [rax + rax*4]
0x15af8:    shl     rax, 4
0x15afc:    add     rax, qword ptr [rbp + 0x38]
0x15b00:    mov     rdi, qword ptr [rax]
0x15b03:    call    0x1d7a0
0x15b08:    mov     rsi, qword ptr [rbp + 0x18]
0x15b0c:    mov     edx, 1
0x15b11:    mov     rdi, rbp
0x15b14:    mov     r12, rax
0x15b17:    xor     eax, eax
0x15b19:    call    0xf880
0x15b1e:    test    eax, eax
0x15b20:    jne     0x15b40
0x15b22:    mov     rdi, rbp
0x15b25:    call    0x14d00
0x15b2a:    mov     rdi, r12
0x15b2d:    call    0x18d10
0x15b32:    jmp     0x159f8
0x15b37:    nop     word ptr [rax + rax]
0x15b40:    mov     rsi, r12
0x15b43:    mov     rdi, rbp
0x15b46:    xor     eax, eax
0x15b48:    call    0x122d0
0x15b4d:    mov     edx, dword ptr [rbp + 0x60]
0x15b50:    mov     esi, dword ptr [rbp + 0x68]
0x15b53:    mov     edi, 0
0x15b58:    mov     dword ptr [rbp + 0x28], eax
0x15b5b:    sub     edx, 2
0x15b5e:    mov     ecx, edx
0x15b60:    shr     ecx, 0x1f
0x15b63:    add     ecx, edx
0x15b65:    sar     ecx, 1
0x15b67:    sub     eax, ecx
0x15b69:    lea     ecx, [rdx + rax]
0x15b6c:    cmovs   eax, edi

```

```

0x15b6f:      cmp     esi, ecx
0x15b71:      jg      0x15b7f
0x15b73:      mov     eax, esi
0x15b75:      sub     eax, edx
0x15b77:      mov     edx, 0
0x15b7c:      cmovs   eax, edx
0x15b7f:      cmp     byte ptr [rbp + 0x6d], 0
0x15b83:      mov     dword ptr [rbp + 0x2c], eax
0x15b86:      je      0x15b2a
0x15b88:      xor     eax, eax
0x15b8a:      mov     rdi, rbp
0x15b8d:      call    0x10d70
0x15b92:      cmp     byte ptr [rbp + 0x6d], 0
0x15b96:      je      0x15b2a
0x15b98:      mov     rdi, rbp
0x15b9b:      xor     eax, eax
0x15b9d:      call    0xec50
0x15ba2:      jmp     0x15b2a
0x15ba4:      call    0x49d0
0x15ba9:      nop     dword ptr [rax]
0x15bb0:      endbr64
0x15bb4:      push    r15
0x15bb6:      mov     r15, rdi
0x15bb9:      push    r14
0x15bbb:      push    r13
0x15bbd:      push    r12
0x15bbf:      mov     r12, rsi
0x15bc2:      push    rbp
0x15bc3:      push    rbx
0x15bc4:      sub     rsp, 0x48
0x15bc8:      mov     edx, dword ptr [rdi + 0x70]
0x15bcb:      mov     rax, qword ptr fs:[0x28]
0x15bd4:      mov     qword ptr [rsp + 0x38], rax
0x15bd9:      xor     eax, eax
0x15bdb:      mov     qword ptr [rsp + 0x28], 0
0x15be4:      mov     qword ptr [rsp + 0x30], 0
0x15bed:      mov     dword ptr [rdi + 0x90], 1
0x15bf7:      test    edx, edx
0x15bf9:      jne     0x15ec8
0x15bff:      movsxd  rax, dword ptr [rdi + 0x28]
0x15c03:      mov     rdx, rax
0x15c06:      lea     rax, [rax + rax*4]
0x15c0a:      shl     rax, 4
0x15c0e:      add     rax, qword ptr [rdi + 0x38]
0x15c12:      mov     rbp, qword ptr [rax]
0x15c15:      test    edx, edx
0x15c17:      jne     0x15c30
0x15c19:      mov     rax, qword ptr [rdi + 0x18]

```

```

0x15c1d:    cmp     byte ptr [rax], 0x2f
0x15c20:    jne     0x15ea0
0x15c26:    cmp     byte ptr [rax + 1], 0
0x15c2a:    jne     0x15ea0
0x15c30:    mov     rdi, rbp
0x15c33:    call    0x49b0
0x15c38:    lea     rdi, [rax + 0x11]
0x15c3c:    call    0x1d670
0x15c41:    mov     edx, 0xe
0x15c46:    mov     rsi, rbp
0x15c49:    lea     rdi, [rip + 0x15990]
0x15c50:    mov     r13, rax
0x15c53:    call    0x4880
0x15c58:    lea     rcx, [rip + 0xb7ef]
0x15c5f:    mov     esi, 1
0x15c64:    mov     rdi, r13
0x15c67:    mov     r8, rax
0x15c6a:    mov     rdx, -1
0x15c71:    xor     eax, eax
0x15c73:    call    0x4f10
0x15c78:    mov     rdi, rbp
0x15c7b:    call    0x49b0
0x15c80:    mov     rdi, qword ptr [r12 + 0x18]
0x15c85:    mov     rbx, rax
0x15c88:    call    0x49b0
0x15c8d:    lea     rdi, [rbx + rax + 2]
0x15c92:    call    0x1d670
0x15c97:    mov     r8, qword ptr [r12 + 0x18]
0x15c9c:    mov     r9, rbp
0x15c9f:    lea     rcx, [rip + 0xaa5b]
0x15ca6:    mov     rdi, rax
0x15ca9:    mov     r14, rax
0x15cac:    mov     esi, 1
0x15cb1:    xor     eax, eax
0x15cb3:    mov     rdx, -1
0x15cba:    call    0x4f10
0x15cbf:    mov     rdi, r13
0x15cc2:    lea     rsi, [rsp + 0x30]
0x15cc7:    mov     rdx, r14
0x15cca:    mov     rcx, qword ptr [rip + 0x1593f]
0x15cd1:    call    0x83e0
0x15cd6:    mov     rdi, r13
0x15cd9:    test    rax, rax
0x15cdc:    je      0x16173
0x15ce2:    call    0x18d10
0x15ce7:    movsxd  rax, dword ptr [r15 + 0x28]
0x15ceb:    lea     rax, [rax + rax*4]
0x15cef:    shl     rax, 4

```

```

0x15cf3:    add     rax, qword ptr [r15 + 0x38]
0x15cf7:    mov     eax, dword ptr [rax + 0x10]
0x15cfa:    and     eax, 0xf000
0x15cff:    cmp     eax, 0x4000
0x15d04:    je      0x160a0
0x15d0a:    mov     rdi, qword ptr [rip + 0x1479f]
0x15d11:    call    0xdb60
0x15d16:    xor     eax, eax
0x15d18:    call    0xa820
0x15d1d:    mov     rdi, qword ptr [rsp + 0x30]
0x15d22:    xor     eax, eax
0x15d24:    call    0x19940
0x15d29:    mov     rdi, qword ptr [rsp + 0x30]
0x15d2e:    mov     r13, rax
0x15d31:    call    0x18d10
0x15d36:    mov     rdi, rbp
0x15d39:    xor     eax, eax
0x15d3b:    mov     rsi, r13
0x15d3e:    mov     qword ptr [rsp + 0x30], r13
0x15d43:    call    0x11190
0x15d48:    mov     rdi, r14
0x15d4b:    mov     ebx, eax
0x15d4d:    call    0x18d10
0x15d52:    test    ebx, ebx
0x15d54:    jne     0x16150
0x15d5a:    movsxd  rax, dword ptr [r15 + 0x28]
0x15d5e:    mov     rdx, qword ptr [rsp + 0x30]
0x15d63:    mov     rsi, rbp
0x15d66:    mov     rdi, r15
0x15d69:    lea     rax, [rax + rax*4]
0x15d6d:    shl     rax, 4
0x15d71:    add     rax, qword ptr [r15 + 0x38]
0x15d75:    mov     ecx, dword ptr [rax + 0x10]
0x15d78:    xor     eax, eax
0x15d7a:    call    0x11d10
0x15d7f:    cmp     eax, 1
0x15d82:    je      0x15db6
0x15d84:    cmp     eax, 2
0x15d87:    je      0x161a0
0x15d8d:    sub     eax, 1
0x15d90:    lea     rdx, [rip + 0x14629]
0x15d97:    xor     ecx, ecx
0x15d99:    mov     rsi, rbp
0x15d9c:    cdq     rsi
0x15d9e:    mov     r8d, 0xe
0x15da4:    lea     rdi, [rip + 0xb6b0]
0x15dab:    mov     rdx, qword ptr [rdx + rax*8]
0x15daf:    xor     eax, eax

```



```

0x15db1:      call    0xf7e0
0x15db6:      mov     rdi, qword ptr [rsp + 0x30]
0x15dbb:      call    0x18d10
0x15dc0:      xor     eax, eax
0x15dc2:      call    0xe4e0
0x15dc7:      mov     rdi, r15
0x15dca:      xor     eax, eax
0x15dcc:      call    0x10240
0x15dd1:      xor     eax, eax
0x15dd3:      mov     rdi, r12
0x15dd6:      call    0x10240
0x15ddb:      xor     eax, eax
0x15ddd:      call    0xa820
0x15de2:      mov     rsi, qword ptr [r12 + 0x18]
0x15de7:      xor     eax, eax
0x15de9:      mov     edx, 1
0x15dee:      mov     rdi, r12
0x15df1:      call    0xf880
0x15df6:      test    eax, eax
0x15df8:      je      0x16190
0x15dfe:      xor     eax, eax
0x15e00:      mov     rdi, r12
0x15e03:      call    0x10ec0
0x15e08:      xor     eax, eax
0x15e0a:      mov     rdi, r12
0x15e0d:      call    0x10160
0x15e12:      cmp     byte ptr [r12 + 0x6d], 0
0x15e18:      je      0x15e24
0x15e1a:      mov     rdi, r12
0x15e1d:      xor     eax, eax
0x15e1f:      call    0xec50
0x15e24:      xor     eax, eax
0x15e26:      mov     rdi, r12
0x15e29:      call    0x10240
0x15e2e:      cmp     byte ptr [r12 + 0x6d], 0
0x15e34:      je      0x15e40
0x15e36:      mov     rdi, r12
0x15e39:      xor     eax, eax
0x15e3b:      call    0x10d70
0x15e40:      mov     rsi, qword ptr [r15 + 0x18]
0x15e44:      xor     eax, eax
0x15e46:      mov     edx, 1
0x15e4b:      mov     rdi, r15
0x15e4e:      call    0xf880
0x15e53:      test    eax, eax
0x15e55:      je      0x16180
0x15e5b:      xor     eax, eax
0x15e5d:      mov     rdi, r15

```

```

0x15e60:    call    0x10ec0
0x15e65:    xor     eax, eax
0x15e67:    mov     rdi, r15
0x15e6a:    call    0x10160
0x15e6f:    cmp     byte ptr [r15 + 0x6d], 0
0x15e74:    je      0x15e80
0x15e76:    mov     rdi, r15
0x15e79:    xor     eax, eax
0x15e7b:    call    0xec50
0x15e80:    xor     eax, eax
0x15e82:    mov     rdi, r15
0x15e85:    call    0x10240
0x15e8a:    cmp     byte ptr [r15 + 0x6d], 0
0x15e8f:    je      0x15ea0
0x15e91:    mov     rdi, r15
0x15e94:    xor     eax, eax
0x15e96:    call    0x10d70
0x15e9b:    nop     dword ptr [rax + rax]
0x15ea0:    mov     rax, qword ptr [rsp + 0x38]
0x15ea5:    sub     rax, qword ptr fs:[0x28]
0x15eae:    jne     0x161bd
0x15eb4:    add     rsp, 0x48
0x15eb8:    pop     rbx
0x15eb9:    pop     rbp
0x15eba:    pop     r12
0x15ebc:    pop     r13
0x15ebe:    pop     r14
0x15ec0:    pop     r15
0x15ec2:    ret
0x15ec3:    nop     dword ptr [rax + rax]
0x15ec8:    mov     rdx, qword ptr [rsi + 0x18]
0x15ecc:    mov     rcx, qword ptr [rip + 0x1573d]
0x15ed3:    lea     rsi, [rsp + 0x28]
0x15ed8:    lea     rdi, [rip + 0xb591]
0x15edf:    call    0x83e0
0x15ee4:    test    rax, rax
0x15ee7:    je      0x15ea0
0x15ee9:    mov     rdi, qword ptr [r15 + 0x18]
0x15eed:    mov     rsi, qword ptr [rsp + 0x28]
0x15ef2:    xor     eax, eax
0x15ef4:    call    0x11190
0x15ef9:    mov     r13d, eax
0x15efc:    test    eax, eax
0x15efe:    jne     0x160b8
0x15f04:    mov     rbp, qword ptr [rsp + 0x28]
0x15f09:    mov     rdi, rbp
0x15f0c:    call    0x49b0
0x15f11:    mov     rdi, rbp

```

```

0x15f14:    mov     rbx, rax
0x15f17:    lea     rax, [rax + 1]
0x15f1b:    lea     r14, [rbx + 2]
0x15f1f:    mov     qword ptr [rsp + 8], rax
0x15f24:    mov     rsi, r14
0x15f27:    call    0x1d690
0x15f2c:    mov     qword ptr [rsp + 0x28], rax
0x15f31:    mov     byte ptr [rax + rbx], 0x2f
0x15f35:    mov     rax, qword ptr [rsp + 0x28]
0x15f3a:    mov     byte ptr [rax + rbx + 1], 0
0x15f3f:    mov     edx, dword ptr [r15 + 0x68]
0x15f43:    test    edx, edx
0x15f45:    jle     0x15f6a
0x15f47:    mov     rax, qword ptr [r15 + 0x38]
0x15f4b:    add     rax, 0x49
0x15f4f:    jmp     0x15f65
0x15f51:    nop     dword ptr [rax]
0x15f58:    add     r13d, 1
0x15f5c:    add     rax, 0x50
0x15f60:    cmp     r13d, edx
0x15f63:    je      0x15f6a
0x15f65:    cmp     byte ptr [rax], 0
0x15f68:    je      0x15f58
0x15f6a:    mov     eax, dword ptr [r15 + 0x70]
0x15f6e:    mov     dword ptr [r15 + 0x50], 0xffffffff
0x15f76:    mov     dword ptr [r15 + 0x54], eax
0x15f7a:    nop     word ptr [rax + rax]
0x15f80:    mov     rdi, r15
0x15f83:    call    0x100e0
0x15f88:    cmp     eax, -1
0x15f8b:    je      0x16060
0x15f91:    cdqe
0x15f93:    lea     rbx, [rax + rax*4]
0x15f97:    mov     rax, qword ptr [r15 + 0x38]
0x15f9b:    shl     rbx, 4
0x15f9f:    mov     rbp, qword ptr [rax + rbx]
0x15fa3:    xor     eax, eax
0x15fa5:    call    0x18ab0
0x15faa:    mov     rdi, rbp
0x15fad:    call    0x49b0
0x15fb2:    mov     rdi, qword ptr [rsp + 0x28]
0x15fb7:    lea     rsi, [r14 + rax]
0x15fbb:    call    0x1d690
0x15fc0:    mov     rcx, qword ptr [rsp + 8]
0x15fc5:    mov     rsi, rbp
0x15fc8:    mov     qword ptr [rsp + 0x28], rax
0x15fcd:    lea     rdi, [rax + rcx]
0x15fd1:    call    0x48b0

```

```

0x15fd6:    mov     eax, dword ptr [rip + 0x15ad4]
0x15fdc:    test    eax, eax
0x15fde:    jne     0x160e0
0x15fe4:    mov     rdi, qword ptr [rip + 0x144cd]
0x15feb:    call    0xdb60
0x15ff0:    xor     eax, eax
0x15ff2:    call    0xa820
0x15ff7:    mov     rax, qword ptr [r15 + 0x38]
0x15ffb:    mov     rsi, rbp
0x15ffe:    mov     rdi, r15
0x16001:    mov     rdx, qword ptr [rsp + 0x28]
0x16006:    mov     ecx, dword ptr [rax + rbx + 0x10]
0x1600a:    xor     eax, eax
0x1600c:    call    0x11d10
0x16011:    cmp     eax, 1
0x16014:    je      0x160d0
0x1601a:    cmp     eax, 2
0x1601d:    je      0x16060
0x1601f:    cmp     eax, 3
0x16022:    je      0x15f80
0x16028:    sub     eax, 1
0x1602b:    lea     rdx, [rip + 0x1438e]
0x16032:    xor     ecx, ecx
0x16034:    mov     rsi, rbp
0x16037:    cdqe
0x16039:    mov     r8d, 0xe
0x1603f:    lea     rdi, [rip + 0xb415]
0x16046:    mov     rdx, qword ptr [rdx + rax*8]
0x1604a:    xor     eax, eax
0x1604c:    call    0xf7e0
0x16051:    test    al, al
0x16053:    jne     0x15f80
0x16059:    nop     dword ptr [rax]
0x16060:    mov     rdi, qword ptr [rsp + 0x28]
0x16065:    test    rdi, rdi
0x16068:    je      0x1606f
0x1606a:    call    0x18d10
0x1606f:    cmp     dword ptr [r15 + 0x68], r13d
0x16073:    je      0x16087
0x16075:    mov     dword ptr [r15 + 0x28], r13d
0x16079:    mov     rdi, r15
0x1607c:    xor     eax, eax
0x1607e:    call    0xf110
0x16083:    mov     dword ptr [r15 + 0x2c], eax
0x16087:    xor     eax, eax
0x16089:    call    0xe4e0
0x1608e:    xor     eax, eax
0x16090:    call    0xa820

```

```

0x16095:      jmp      0x15de2
0x1609a:      nop      word ptr [rax + rax]
0x160a0:      mov      rdi, qword ptr [rip + 0x14419]
0x160a7:      call     0xdb60
0x160ac:      jmp      0x15d16
0x160b1:      nop      dword ptr [rax]
0x160b8:      mov      edx, 0xc
0x160bd:      xor      esi, esi
0x160bf:      lea      rdi, [rip + 0x1423a]
0x160c6:      call     0x8070
0x160cb:      jmp      0x15ea0
0x160d0:      mov      rax, qword ptr [r15 + 0x38]
0x160d4:      mov      byte ptr [rax + rbx + 0x49], 0
0x160d9:      jmp      0x15f80
0x160de:      nop
0x160e0:      mov      dword ptr [rip + 0x159c6], 0
0x160ea:      xor      eax, eax
0x160ec:      call     0xc8c0
0x160f1:      mov      edx, 4
0x160f6:      lea      rsi, [rip + 0xa435]
0x160fd:      lea      rdi, [rip + 0xb0f5]
0x16104:      mov      qword ptr [rsp + 0x18], rax
0x16109:      call     0x8070
0x1610e:      mov      rdi, qword ptr [rsp + 0x18]
0x16113:      mov      byte ptr [rsp + 0x17], al
0x16117:      call     0xc970
0x1611c:      xor      eax, eax
0x1611e:      call     0xd870
0x16123:      xor      eax, eax
0x16125:      call     0xd740
0x1612a:      xor      eax, eax
0x1612c:      call     0xa820
0x16131:      movzx    eax, byte ptr [rsp + 0x17]
0x16136:      and      eax, 0xfffffdf
0x16139:      cmp      al, 0x4e
0x1613b:      je       0x15fe4
0x16141:      jmp      0x16060
0x16146:      nop      word ptr cs:[rax + rax]
0x16150:      mov      rdx, qword ptr [rsp + 0x30]
0x16155:      lea      rdi, [rip + 0xbaf4]
0x1615c:      xor      ecx, ecx
0x1615e:      xor      eax, eax
0x16160:      mov      r8d, 0x1e
0x16166:      mov      rsi, rbp
0x16169:      call     0xf7e0
0x1616e:      mov      rdi, qword ptr [rsp + 0x30]
0x16173:      call     0x18d10
0x16178:      jmp      0x15ea0

```

```

0x1617d:      nop      dword ptr [rax]
0x16180:      mov      rdi, r15
0x16183:      call     0x14d00
0x16188:      jmp      0x15e5b
0x1618d:      nop      dword ptr [rax]
0x16190:      mov      rdi, r12
0x16193:      call     0x14d00
0x16198:      jmp      0x15dfe
0x1619d:      nop      dword ptr [rax]
0x161a0:      mov      rdi, qword ptr [rsp + 0x30]
0x161a5:      call     0x18d10
0x161aa:      xor      eax, eax
0x161ac:      call     0xe4e0
0x161b1:      xor      eax, eax
0x161b3:      call     0xa820
0x161b8:      jmp      0x15ea0
0x161bd:      call     0x49d0
0x161c2:      nop      word ptr cs:[rax + rax]
0x161cd:      nop      dword ptr [rax]
0x161d0:      endbr64
0x161d4:      push     r13
0x161d6:      xor      r13d, r13d
0x161d9:      push     r12
0x161db:      lea      r12, [rip + 0xc203]
0x161e2:      push     rbp
0x161e3:      mov      rbp, rdi
0x161e6:      push     rbx
0x161e7:      mov      rbx, rsi
0x161ea:      sub      rsp, 8
0x161ee:      mov      rdx, qword ptr [rdi + 0x38]
0x161f2:      test     rdx, rdx
0x161f5:      je       0x1621a
0x161f7:      movsxd   rax, dword ptr [rdi + 0x28]
0x161fb:      lea      rax, [rax + rax*4]
0x161ff:      shl      rax, 4
0x16203:      mov      rdi, qword ptr [rdx + rax]
0x16207:      test     rdi, rdi
0x1620a:      je       0x1621a
0x1620c:      call     0x1d7a0
0x16211:      mov      r13d, 1
0x16217:      mov      r12, rax
0x1621a:      xor      edx, edx
0x1621c:      mov      rsi, qword ptr [rbp + 0x18]
0x16220:      cmp      rbx, -1
0x16224:      mov      rdi, rbp
0x16227:      sete     dl
0x1622a:      xor      eax, eax
0x1622c:      call     0xf880

```

```

0x16231:    test    eax, eax
0x16233:    je      0x16278
0x16235:    cmp     rbx, -1
0x16239:    je      0x16288
0x1623b:    mov     dword ptr [rbp + 0x28], 0
0x16242:    mov     rdi, rbp
0x16245:    xor     eax, eax
0x16247:    call    0xf110
0x1624c:    mov     dword ptr [rbp + 0x2c], eax
0x1624f:    test    r13d, r13d
0x16252:    jne     0x16260
0x16254:    add     rsp, 8
0x16258:    pop     rbx
0x16259:    pop     rbp
0x1625a:    pop     r12
0x1625c:    pop     r13
0x1625e:    ret
0x1625f:    nop
0x16260:    add     rsp, 8
0x16264:    mov     rdi, r12
0x16267:    pop     rbx
0x16268:    pop     rbp
0x16269:    pop     r12
0x1626b:    pop     r13
0x1626d:    jmp     0x18d10
0x16272:    nop     word ptr [rax + rax]
0x16278:    mov     rdi, rbp
0x1627b:    xor     eax, eax
0x1627d:    call    0x14d00
0x16282:    jmp     0x1624f
0x16284:    nop     dword ptr [rax]
0x16288:    mov     rsi, r12
0x1628b:    mov     rdi, rbp
0x1628e:    xor     eax, eax
0x16290:    call    0x122d0
0x16295:    mov     rdi, rbp
0x16298:    mov     dword ptr [rbp + 0x28], eax
0x1629b:    xor     eax, eax
0x1629d:    call    0xf110
0x162a2:    mov     dword ptr [rbp + 0x2c], eax
0x162a5:    jmp     0x1624f
0x162a7:    nop     word ptr [rax + rax]
0x162b0:    endbr64
0x162b4:    push    r15
0x162b6:    push    r14
0x162b8:    push    r13
0x162ba:    push    r12
0x162bc:    mov     r12, rdi

```

```

0x162bf:      push    rbp
0x162c0:      movsxd  rax, dword ptr [rdi + 0x28]
0x162c4:      mov     rbp, rsi
0x162c7:      lea     rax, [rax + rax*4]
0x162cb:      shl     rax, 4
0x162cf:      add     rax, qword ptr [rdi + 0x38]
0x162d3:      movzx   edx, byte ptr [rax + 0x48]
0x162d7:      mov     r14, qword ptr [rax]
0x162da:      test    dl, dl
0x162dc:      je      0x16300
0x162de:      cmp     dl, 1
0x162e1:      jne     0x162ed
0x162e3:      cmp     byte ptr [rax + 0x4a], 0
0x162e7:      jne     0x16420
0x162ed:      xor     r13d, r13d
0x162f0:      pop     rbp
0x162f1:      mov     eax, r13d
0x162f4:      pop     r12
0x162f6:      pop     r13
0x162f8:      pop     r14
0x162fa:      pop     r15
0x162fc:      ret
0x162fd:      nop     dword ptr [rax]
0x16300:      cmp     byte ptr [r14], 0x2e
0x16304:      jne     0x163c0
0x1630a:      cmp     byte ptr [r14 + 1], 0x2e
0x1630f:      jne     0x163c0
0x16315:      cmp     byte ptr [r14 + 2], 0
0x1631a:      jne     0x163c0
0x16320:      mov     rax, qword ptr [rdi + 0x18]
0x16324:      cmp     byte ptr [rax], 0x2f
0x16327:      jne     0x16330
0x16329:      cmp     byte ptr [rax + 1], 0
0x1632d:      je      0x162ed
0x1632f:      nop
0x16330:      mov     rax, qword ptr [r12 + 0x78]
0x16335:      lea     rdi, [rax + 1]
0x16339:      call    0x1d670
0x1633e:      mov     rsi, qword ptr [r12 + 0x18]
0x16343:      mov     rdi, rax
0x16346:      mov     r15, rax
0x16349:      call    0x48b0
0x1634e:      mov     edx, 1
0x16353:      mov     rsi, r14
0x16356:      mov     rdi, r12
0x16359:      xor     eax, eax
0x1635b:      call    0xf880
0x16360:      mov     r13d, eax

```



```

0x16363:      test    eax, eax
0x16365:      jne     0x165c0
0x1636b:      xor     eax, eax
0x1636d:      mov     rdi, r12
0x16370:      call    0x14d00
0x16375:      xor     eax, eax
0x16377:      mov     rdi, r12
0x1637a:      call    0x10ec0
0x1637f:      xor     eax, eax
0x16381:      mov     rdi, r12
0x16384:      call    0x10160
0x16389:      cmp     byte ptr [r12 + 0x6d], 0
0x1638f:      jne     0x165b0
0x16395:      xor     eax, eax
0x16397:      mov     rdi, r12
0x1639a:      call    0x10240
0x1639f:      cmp     byte ptr [r12 + 0x6d], 0
0x163a5:      je      0x162ed
0x163ab:      mov     rdi, r12
0x163ae:      xor     eax, eax
0x163b0:      call    0x10d70
0x163b5:      jmp     0x162f0
0x163ba:      nop     word ptr [rax + rax]
0x163c0:      mov     rax, qword ptr [r12 + 0x78]
0x163c5:      lea     rdi, [rax + 1]
0x163c9:      call    0x1d670
0x163ce:      mov     rsi, qword ptr [r12 + 0x18]
0x163d3:      mov     rdi, rax
0x163d6:      mov     r15, rax
0x163d9:      call    0x48b0
0x163de:      mov     edx, 1
0x163e3:      mov     rsi, r14
0x163e6:      mov     rdi, r12
0x163e9:      xor     eax, eax
0x163eb:      call    0xf880
0x163f0:      mov     r13d, eax
0x163f3:      test    eax, eax
0x163f5:      jne     0x164b0
0x163fb:      mov     ecx, 0x1c
0x16400:      xor     edx, edx
0x16402:      mov     rsi, r14
0x16405:      xor     eax, eax
0x16407:      lea     rdi, [rip + 0xaf94]
0x1640e:      call    0xf760
0x16413:      jmp     0x162f0
0x16418:      nop     dword ptr [rax + rax]
0x16420:      mov     rdi, r14
0x16423:      call    0x49b0

```

```

0x16428:    lea     rdi, [rax + 0x21]
0x1642c:    call   0x1d670
0x16431:    mov     r8, r14
0x16434:    mov     esi, 1
0x16439:    lea     rcx, [rip + 0xb04b]
0x16440:    mov     rdx, -1
0x16447:    mov     r13, rax
0x1644a:    mov     rdi, rax
0x1644d:    xor     eax, eax
0x1644f:    call   0x4f10
0x16454:    xor     esi, esi
0x16456:    mov     rdi, r13
0x16459:    call   0x193f0
0x1645e:    mov     rdi, r13
0x16461:    mov     r13d, 1
0x16467:    mov     dword ptr [rip + 0x1446b], 1
0x16471:    call   0x18d10
0x16476:    xor     eax, eax
0x16478:    call   0xa480
0x1647d:    mov     rdi, qword ptr [rip + 0x14444]
0x16484:    call   0xbb10
0x16489:    mov     rdi, r14
0x1648c:    mov     dword ptr [r12 + 0x14], 0xffffffff
0x16495:    mov     dword ptr [rbp + 0x14], 0xffffffff
0x1649c:    call   0xd610
0x164a1:    mov     rdi, qword ptr [r12 + 0x18]
0x164a6:    call   0xc4b0
0x164ab:    jmp     0x162f0
0x164b0:    mov     dword ptr [r12 + 0x28], 0
0x164b9:    mov     rdi, r12
0x164bc:    xor     eax, eax
0x164be:    call   0xf110
0x164c3:    mov     dword ptr [r12 + 0x2c], eax
0x164c8:    mov     rdi, r15
0x164cb:    call   0x18d10
0x164d0:    xor     eax, eax
0x164d2:    mov     rdi, r12
0x164d5:    call   0x10ec0
0x164da:    xor     eax, eax
0x164dc:    mov     rdi, r12
0x164df:    call   0x10160
0x164e4:    cmp     byte ptr [r12 + 0x6d], 0
0x164ea:    je      0x164f6
0x164ec:    mov     rdi, r12
0x164ef:    xor     eax, eax
0x164f1:    call   0xec50
0x164f6:    xor     eax, eax
0x164f8:    mov     rdi, r12

```

```

0x164fb:      call    0x10240
0x16500:      cmp     byte ptr [r12 + 0x6d], 0
0x16506:      je      0x16512
0x16508:      mov     rdi, r12
0x1650b:      xor     eax, eax
0x1650d:      call    0x10d70
0x16512:      mov     rsi, qword ptr [rbp + 0x18]
0x16516:      mov     rdi, qword ptr [r12 + 0x18]
0x1651b:      call    0x4b30
0x16520:      mov     r13d, eax
0x16523:      test    eax, eax
0x16525:      jne     0x16588
0x16527:      mov     r8d, 1
0x1652d:      mov     rdx, r12
0x16530:      mov     esi, 0x22
0x16535:      mov     rdi, rbp
0x16538:      mov     rcx, -1
0x1653f:      call    0x13e30
0x16544:      xor     eax, eax
0x16546:      mov     rdi, rbp
0x16549:      call    0x10ec0
0x1654e:      xor     eax, eax
0x16550:      mov     rdi, rbp
0x16553:      call    0x10160
0x16558:      cmp     byte ptr [rbp + 0x6d], 0
0x1655c:      jne     0x165a0
0x1655e:      xor     eax, eax
0x16560:      mov     rdi, rbp
0x16563:      call    0x10240
0x16568:      cmp     byte ptr [rbp + 0x6d], 0
0x1656c:      je      0x162ed
0x16572:      mov     rdi, rbp
0x16575:      xor     eax, eax
0x16577:      call    0x10d70
0x1657c:      jmp     0x162f0
0x16581:      nop     dword ptr [rax]
0x16588:      mov     rdi, rbp
0x1658b:      xor     eax, eax
0x1658d:      call    0x10240
0x16592:      jmp     0x162ed
0x16597:      nop     word ptr [rax + rax]
0x165a0:      mov     rdi, rbp
0x165a3:      xor     eax, eax
0x165a5:      call    0xec50
0x165aa:      jmp     0x1655e
0x165ac:      nop     dword ptr [rax]
0x165b0:      mov     rdi, r12
0x165b3:      xor     eax, eax

```

```

0x165b5:      call    0xec50
0x165ba:      jmp     0x16395
0x165bf:      nop
0x165c0:      mov     rdi, r15
0x165c3:      mov     esi, 0x2f
0x165c8:      call    0x4a20
0x165cd:      mov     rdi, r12
0x165d0:      lea     rsi, [rax + 1]
0x165d4:      xor     eax, eax
0x165d6:      call    0x122d0
0x165db:      mov     rdi, r12
0x165de:      mov     dword ptr [r12 + 0x28], eax
0x165e3:      xor     eax, eax
0x165e5:      call    0xf110
0x165ea:      jmp     0x164c3
0x165ef:      nop
0x165f0:      endbr64
0x165f4:      push    r13
0x165f6:      push    r12
0x165f8:      mov     r12, rdx
0x165fb:      push    rbp
0x165fc:      mov     rbp, rsi
0x165ff:      push    rbx
0x16600:      mov     rbx, rdi
0x16603:      mov     rdi, rdx
0x16606:      sub     rsp, 8
0x1660a:      call    0x49b0
0x1660f:      cmp     byte ptr [r12], 0x2f
0x16614:      je      0x166f0
0x1661a:      add     rax, qword ptr [rbx + 0x78]
0x1661e:      mov     rdi, qword ptr [rbx + 0x18]
0x16622:      lea     rdx, [rax + 1]
0x16626:      lea     rsi, [rax + 2]
0x1662a:      mov     qword ptr [rbx + 0x78], rdx
0x1662e:      call    0x1d690
0x16633:      mov     qword ptr [rbx + 0x18], rax
0x16637:      mov     rdi, rax
0x1663a:      mov     r13, rax
0x1663d:      call    0x49b0
0x16642:      mov     edx, 0x2f
0x16647:      mov     rsi, r12
0x1664a:      mov     word ptr [r13 + rax], dx
0x16650:      mov     rdi, qword ptr [rbx + 0x18]
0x16654:      call    0x4df0
0x16659:      mov     rdi, qword ptr [rbx + 0x18]
0x1665d:      call    0x18040
0x16662:      mov     dword ptr [rbx + 0x28], 0
0x16669:      mov     rdi, rbx

```

```

0x1666c:    xor     eax, eax
0x1666e:    mov     r8d, 1
0x16674:    call    0xf110
0x16679:    xor     ecx, ecx
0x1667b:    xor     edx, edx
0x1667d:    mov     esi, 0x22
0x16682:    mov     dword ptr [rbx + 0x2c], eax
0x16685:    mov     rdi, rbx
0x16688:    call    0x13e30
0x1668d:    xor     eax, eax
0x1668f:    mov     rdi, rbx
0x16692:    call    0x10ec0
0x16697:    xor     eax, eax
0x16699:    mov     rdi, rbx
0x1669c:    call    0x10160
0x166a1:    cmp     byte ptr [rbx + 0x6d], 0
0x166a5:    je      0x166b1
0x166a7:    mov     rdi, rbx
0x166aa:    xor     eax, eax
0x166ac:    call    0xec50
0x166b1:    xor     eax, eax
0x166b3:    mov     rdi, rbx
0x166b6:    call    0x10240
0x166bb:    cmp     byte ptr [rbx + 0x6d], 0
0x166bf:    je      0x166cb
0x166c1:    mov     rdi, rbx
0x166c4:    xor     eax, eax
0x166c6:    call    0x10d70
0x166cb:    mov     rsi, qword ptr [rbp + 0x18]
0x166cf:    mov     rdi, qword ptr [rbx + 0x18]
0x166d3:    call    0x4b30
0x166d8:    test    eax, eax
0x166da:    je      0x16718
0x166dc:    add     rsp, 8
0x166e0:    pop     rbx
0x166e1:    pop     rbp
0x166e2:    pop     r12
0x166e4:    pop     r13
0x166e6:    ret
0x166e7:    nop     word ptr [rax + rax]
0x166f0:    mov     qword ptr [rbx + 0x78], rax
0x166f4:    mov     rdi, qword ptr [rbx + 0x18]
0x166f8:    lea     rsi, [rax + 1]
0x166fc:    call    0x1d690
0x16701:    mov     rsi, r12
0x16704:    mov     qword ptr [rbx + 0x18], rax
0x16708:    mov     rdi, rax
0x1670b:    call    0x48b0

```

```

0x16710:      jmp      0x16662
0x16715:      nop      dword ptr [rax]
0x16718:      xor      edx, edx
0x1671a:      mov      r8d, 1
0x16720:      mov      esi, 0x22
0x16725:      mov      rdi, rbp
0x16728:      mov      rcx, -1
0x1672f:      call     0x13e30
0x16734:      xor      eax, eax
0x16736:      mov      rdi, rbp
0x16739:      call     0x10ec0
0x1673e:      xor      eax, eax
0x16740:      mov      rdi, rbp
0x16743:      call     0x10160
0x16748:      cmp      byte ptr [rbp + 0x6d], 0
0x1674c:      jne      0x16780
0x1674e:      xor      eax, eax
0x16750:      mov      rdi, rbp
0x16753:      call     0x10240
0x16758:      cmp      byte ptr [rbp + 0x6d], 0
0x1675c:      je       0x166dc
0x16762:      add      rsp, 8
0x16766:      mov      rdi, rbp
0x16769:      xor      eax, eax
0x1676b:      pop      rbx
0x1676c:      pop      rbp
0x1676d:      pop      r12
0x1676f:      pop      r13
0x16771:      jmp      0x10d70
0x16776:      nop      word ptr cs:[rax + rax]
0x16780:      xor      eax, eax
0x16782:      mov      rdi, rbp
0x16785:      call     0xec50
0x1678a:      xor      eax, eax
0x1678c:      mov      rdi, rbp
0x1678f:      call     0x10240
0x16794:      cmp      byte ptr [rbp + 0x6d], 0
0x16798:      je       0x166dc
0x1679e:      jmp      0x16762
0x167a0:      endbr64
0x167a4:      push     r15
0x167a6:      lea      r15, [rip + 0x13bd3]
0x167ad:      push     r14
0x167af:      lea      r14, [rip + 0xacdc]
0x167b6:      push     r13
0x167b8:      mov      r13, rsi
0x167bb:      push     r12
0x167bd:      mov      r12d, edx

```

```

0x167c0:    push    rbp
0x167c1:    mov     rbp, rdi
0x167c4:    push    rbx
0x167c5:    sub     rsp, 8
0x167c9:    mov     rdi, qword ptr [rip + 0x13d08]
0x167d0:    call    0xdb60
0x167d5:    xor     eax, eax
0x167d7:    call    0xa820
0x167dc:    mov     eax, dword ptr [rbp + 0x70]
0x167df:    mov     dword ptr [rbp + 0x50], 0xffffffff
0x167e6:    mov     dword ptr [rbp + 0x90], 1
0x167f0:    mov     dword ptr [rbp + 0x54], eax
0x167f3:    nop     dword ptr [rax + rax]
0x167f8:    mov     rdi, rbp
0x167fb:    call    0x100e0
0x16800:    movsxd  rbx, eax
0x16803:    cmp     ebx, -1
0x16806:    je      0x1686b
0x16808:    xor     eax, eax
0x1680a:    call    0x18ab0
0x1680f:    xor     eax, eax
0x16811:    mov     edx, r12d
0x16814:    mov     esi, ebx
0x16816:    mov     rdi, rbp
0x16819:    call    0x134d0
0x1681e:    cmp     eax, 1
0x16821:    je      0x168f0
0x16827:    cmp     eax, 2
0x1682a:    je      0x1686b
0x1682c:    cmp     eax, 3
0x1682f:    je      0x167f8
0x16831:    sub     eax, 1
0x16834:    xor     ecx, ecx
0x16836:    mov     rdi, r14
0x16839:    mov     r8d, 0x1e
0x1683f:    cdqe
0x16841:    mov     rdx, qword ptr [r15 + rax*8]
0x16845:    lea     rax, [rbx + rbx*4]
0x16849:    shl     rax, 4
0x1684d:    add     rax, qword ptr [rbp + 0x38]
0x16851:    mov     rsi, qword ptr [rax]
0x16854:    xor     eax, eax
0x16856:    call    0xf7e0
0x1685b:    mov     rdi, rbp
0x1685e:    call    0x100e0
0x16863:    movsxd  rbx, eax
0x16866:    cmp     ebx, -1
0x16869:    jne     0x16808

```

```

0x1686b:    xor     eax, eax
0x1686d:    call    0xe4e0
0x16872:    xor     eax, eax
0x16874:    call    0xa820
0x16879:    xor     ecx, ecx
0x1687b:    xor     edx, edx
0x1687d:    mov     r8d, 1
0x16883:    mov     esi, 0x22
0x16888:    mov     rdi, rbp
0x1688b:    call    0x13e30
0x16890:    xor     eax, eax
0x16892:    mov     rdi, rbp
0x16895:    call    0x10ec0
0x1689a:    xor     eax, eax
0x1689c:    mov     rdi, rbp
0x1689f:    call    0x10160
0x168a4:    cmp     byte ptr [rbp + 0x6d], 0
0x168a8:    je      0x168b4
0x168aa:    mov     rdi, rbp
0x168ad:    xor     eax, eax
0x168af:    call    0xec50
0x168b4:    xor     eax, eax
0x168b6:    mov     rdi, rbp
0x168b9:    call    0x10240
0x168be:    cmp     byte ptr [rbp + 0x6d], 0
0x168c2:    je      0x168ce
0x168c4:    mov     rdi, rbp
0x168c7:    xor     eax, eax
0x168c9:    call    0x10d70
0x168ce:    mov     rsi, qword ptr [r13 + 0x18]
0x168d2:    mov     rdi, qword ptr [rbp + 0x18]
0x168d6:    call    0x4b30
0x168db:    test    eax, eax
0x168dd:    je      0x16908
0x168df:    add     rsp, 8
0x168e3:    pop     rbx
0x168e4:    pop     rbp
0x168e5:    pop     r12
0x168e7:    pop     r13
0x168e9:    pop     r14
0x168eb:    pop     r15
0x168ed:    ret
0x168ee:    nop
0x168f0:    lea     rax, [rbx + rbx*4]
0x168f4:    shl     rax, 4
0x168f8:    add     rax, qword ptr [rbp + 0x38]
0x168fc:    mov     byte ptr [rax + 0x49], 0
0x16900:    jmp     0x167f8

```



```

0x16905:      nop      dword ptr [rax]
0x16908:      xor      ecx, ecx
0x1690a:      xor      edx, edx
0x1690c:      mov      r8d, 1
0x16912:      mov      esi, 0x22
0x16917:      mov      rdi, r13
0x1691a:      call     0x13e30
0x1691f:      xor      eax, eax
0x16921:      mov      rdi, r13
0x16924:      call     0x10ec0
0x16929:      xor      eax, eax
0x1692b:      mov      rdi, r13
0x1692e:      call     0x10160
0x16933:      cmp      byte ptr [r13 + 0x6d], 0
0x16938:      jne      0x16968
0x1693a:      xor      eax, eax
0x1693c:      mov      rdi, r13
0x1693f:      call     0x10240
0x16944:      cmp      byte ptr [r13 + 0x6d], 0
0x16949:      je       0x168df
0x1694b:      add      rsp, 8
0x1694f:      mov      rdi, r13
0x16952:      xor      eax, eax
0x16954:      pop      rbx
0x16955:      pop      rbp
0x16956:      pop      r12
0x16958:      pop      r13
0x1695a:      pop      r14
0x1695c:      pop      r15
0x1695e:      jmp      0x10d70
0x16963:      nop      dword ptr [rax + rax]
0x16968:      mov      rdi, r13
0x1696b:      xor      eax, eax
0x1696d:      call     0xec50
0x16972:      jmp      0x1693a
0x16974:      nop      word ptr cs:[rax + rax]
0x1697e:      nop
0x16980:      push     r15
0x16982:      push     r14
0x16984:      push     r13
0x16986:      push     r12
0x16988:      push     rbp
0x16989:      push     rbx
0x1698a:      sub      rsp, 0x8c8
0x16991:      mov      rdi, qword ptr [rip + 0x14ca8]
0x16998:      mov      r13, qword ptr [rip + 0x13cc1]
0x1699f:      mov      rax, qword ptr fs:[0x28]
0x169a8:      mov      qword ptr [rsp + 0x8b8], rax

```

```

0x169b0:    xor     eax, eax
0x169b2:    lea     rax, [rip + 0x14c8f]
0x169b9:    mov     qword ptr [rip + 0x13ca0], rax
0x169c0:    test    rdi, rdi
0x169c3:    je      0x16a78
0x169c9:    lea     rsi, [rsp + 0x20]
0x169ce:    call    0x18f40
0x169d3:    cmp     eax, -1
0x169d6:    je      0x16a78
0x169dc:    mov     rax, qword ptr [rsp + 0x68]
0x169e1:    mov     rdi, qword ptr [rip + 0x14c58]
0x169e8:    lea     rsi, [rip + 0xb5da]
0x169ef:    mov     qword ptr [rsp + 0x10], rax
0x169f4:    mov     rax, qword ptr [rsp + 0x78]
0x169f9:    mov     qword ptr [rsp + 0x18], rax
0x169fe:    call    0x4da0
0x16a03:    mov     r14, rax
0x16a06:    test    rax, rax
0x16a09:    je      0x16a78
0x16a0b:    mov     dword ptr [rsp + 4], 0
0x16a13:    xor     ebp, ebp
0x16a15:    xor     ebx, ebx
0x16a17:    xor     r12d, r12d
0x16a1a:    lea     r15, [rsp + 0xb0]
0x16a22:    nop     word ptr [rax + rax]
0x16a28:    mov     rdx, r14
0x16a2b:    mov     esi, 0x800
0x16a30:    mov     rdi, r15
0x16a33:    call    0x4b00
0x16a38:    test    rax, rax
0x16a3b:    je      0x16ad0
0x16a41:    cmp     byte ptr [rsp + 0xb0], 0
0x16a49:    je      0x16aa8
0x16a4b:    cmp     dword ptr [r15], 0x6d6f7246
0x16a52:    je      0x16ab0
0x16a54:    test    ebp, ebp
0x16a56:    jne     0x16a28
0x16a58:    test    ebx, ebx
0x16a5a:    je      0x16a28
0x16a5c:    cmp     dword ptr [r15], 0x74617453
0x16a63:    je      0x16b60
0x16a69:    xor     ebp, ebp
0x16a6b:    mov     ebx, 1
0x16a70:    jmp     0x16a28
0x16a72:    nop     word ptr [rax + rax]
0x16a78:    xor     eax, eax
0x16a7a:    mov     rdx, qword ptr [rsp + 0x8b8]
0x16a82:    sub     rdx, qword ptr fs:[0x28]

```

```

0x16a8b:    jne     0x16c02
0x16a91:    add     rsp, 0x8c8
0x16a98:    pop     rbx
0x16a99:    pop     rbp
0x16a9a:    pop     r12
0x16a9c:    pop     r13
0x16a9e:    pop     r14
0x16aa0:    pop     r15
0x16aa2:    ret
0x16aa3:    nop     dword ptr [rax + rax]
0x16aa8:    xor     ebx, ebx
0x16aaa:    jmp     0x16a28
0x16aaf:    nop
0x16ab0:    cmp     byte ptr [r15 + 4], 0x20
0x16ab5:    jne     0x16a54
0x16ab7:    add     r12d, 1
0x16abb:    xor     ebp, ebp
0x16abd:    mov     ebx, 1
0x16ac2:    jmp     0x16a28
0x16ac7:    nop     word ptr [rax + rax]
0x16ad0:    mov     rdi, r14
0x16ad3:    call    0x4940
0x16ad8:    mov     rdi, qword ptr [rip + 0x14b61]
0x16adf:    lea     rsi, [rsp + 0x10]
0x16ae4:    call    0x47f0
0x16ae9:    mov     rbp, qword ptr [rip + 0x13b70]
0x16af0:    test    r12d, r12d
0x16af3:    je      0x16b0a
0x16af5:    cmp     r12d, dword ptr [rsp + 4]
0x16afa:    je      0x16b48
0x16afc:    lea     rbp, [rip + 0x13b1d]
0x16b03:    mov     qword ptr [rip + 0x13b56], rbp
0x16b0a:    mov     rdi, rbp
0x16b0d:    call    0x49b0
0x16b12:    mov     rsi, r13
0x16b15:    mov     rdi, rbp
0x16b18:    mov     r8, rax
0x16b1b:    mov     eax, dword ptr [rip + 0x14b2f]
0x16b21:    add     eax, dword ptr [rip + 0x14b2d]
0x16b27:    lea     eax, [rax + r8 + 0x15]
0x16b2c:    mov     dword ptr [rip + 0x14b1a], eax
0x16b32:    call    0x4b30
0x16b37:    test    eax, eax
0x16b39:    setne   al
0x16b3c:    movzx   eax, al
0x16b3f:    jmp     0x16a7a
0x16b44:    nop     dword ptr [rax]
0x16b48:    lea     rbp, [rip + 0x13adc]

```

```

0x16b4f:      mov     qword ptr [rip + 0x13b0a], rbp
0x16b56:      jmp     0x16b0a
0x16b58:      nop     dword ptr [rax + rax]
0x16b60:      cmp     word ptr [r15 + 4], 0x7375
0x16b67:      jne     0x16a69
0x16b6d:      cmp     byte ptr [r15 + 6], 0x3a
0x16b72:      jne     0x16a69
0x16b78:      mov     rdi, r15
0x16b7b:      mov     esi, 0x3a
0x16b80:      call    0x4a00
0x16b85:      movsx   rbx, byte ptr [rax + 1]
0x16b8a:      lea     rdi, [rax + 1]
0x16b8e:      test    bl, bl
0x16b90:      je      0x16bf8
0x16b92:      mov     qword ptr [rsp + 8], rdi
0x16b97:      call    0x4f00
0x16b9c:      mov     rdi, qword ptr [rsp + 8]
0x16ba1:      mov     rax, qword ptr [rax]
0x16ba4:      jmp     0x16bbd
0x16ba6:      nop     word ptr cs:[rax + rax]
0x16bb0:      movsx   rbx, byte ptr [rdi + 1]
0x16bb5:      add     rdi, 1
0x16bb9:      test    bl, bl
0x16bbb:      je      0x16bf8
0x16bbd:      test    byte ptr [rax + rbx*2 + 1], 0x20
0x16bc2:      jne     0x16bb0
0x16bc4:      mov     esi, 0x52
0x16bc9:      mov     ebp, 1
0x16bce:      mov     ebx, 1
0x16bd3:      call    0x4a00
0x16bd8:      test    rax, rax
0x16bdb:      je      0x16a28
0x16be1:      add     dword ptr [rsp + 4], 1
0x16be6:      mov     ebp, 1
0x16beb:      mov     ebx, 1
0x16bf0:      jmp     0x16a28
0x16bf5:      nop     dword ptr [rax]
0x16bf8:      mov     ebx, 1
0x16bfd:      jmp     0x16a28
0x16c02:      call    0x49d0
0x16c07:      nop     word ptr [rax + rax]
0x16c10:      endbr64
0x16c14:      sub     rsp, 0xa8
0x16c1b:      mov     edi, 0x1d
0x16c20:      mov     rax, qword ptr fs:[0x28]
0x16c29:      mov     qword ptr [rsp + 0x98], rax
0x16c31:      xor     eax, eax
0x16c33:      call    0x1d670

```

```

0x16c38:    lea     r9, [rip + 0x9950]
0x16c3f:    lea     r8, [rip + 0x991f]
0x16c46:    mov     rdx, -1
0x16c4d:    lea     rcx, [rip + 0xb219]
0x16c54:    mov     esi, 1
0x16c59:    mov     rdi, rax
0x16c5c:    mov     qword ptr [rip + 0x149fd], rax
0x16c63:    xor     eax, eax
0x16c65:    call    0x4f10
0x16c6a:    mov     rdi, qword ptr [rip + 0x149ef]
0x16c71:    call    0x49b0
0x16c76:    mov     rdi, qword ptr [rip + 0x14e1b]
0x16c7d:    mov     dword ptr [rip + 0x149d5], eax
0x16c83:    call    0x49b0
0x16c88:    mov     rdi, qword ptr [rip + 0x13d01]
0x16c8f:    mov     dword ptr [rip + 0x149bf], eax
0x16c95:    call    0x49b0
0x16c9a:    lea     rdi, [rip + 0x9999]
0x16ca1:    mov     dword ptr [rip + 0x149a9], eax
0x16ca7:    call    0x47e0
0x16cac:    mov     qword ptr [rip + 0x1498d], rax
0x16cb3:    test    rax, rax
0x16cb6:    je      0x16cc3
0x16cb8:    mov     rdi, rax
0x16cbb:    mov     rsi, rsp
0x16cbe:    call    0x18f40
0x16cc3:    xor     eax, eax
0x16cc5:    call    0x16980
0x16cca:    mov     rdi, qword ptr [rip + 0x1398f]
0x16cd1:    call    0x49b0
0x16cd6:    mov     r8, rax
0x16cd9:    mov     eax, dword ptr [rip + 0x14971]
0x16cdf:    add     eax, dword ptr [rip + 0x1496f]
0x16ce5:    lea     eax, [rax + r8 + 0x15]
0x16cea:    mov     dword ptr [rip + 0x1495c], eax
0x16cf0:    xor     eax, eax
0x16cf2:    call    0xc590
0x16cf7:    mov     qword ptr [rip + 0x1493a], rax
0x16cfe:    mov     rax, qword ptr [rsp + 0x98]
0x16d06:    sub     rax, qword ptr fs:[0x28]
0x16d0f:    jne     0x16d19
0x16d11:    add     rsp, 0xa8
0x16d18:    ret
0x16d19:    call    0x49d0
0x16d1e:    nop
0x16d20:    endbr64
0x16d24:    mov     rdi, qword ptr [rip + 0x1490d]
0x16d2b:    jmp     0xc5c0

```

```

0x16d30:      endbr64
0x16d34:      mov     r8d, edi
0x16d37:      mov     rdi, qword ptr [rip + 0x148fa]
0x16d3e:      mov     edx, esi
0x16d40:      mov     ecx, 1
0x16d45:      xor     esi, esi
0x16d47:      jmp     0xc5e0
0x16d4c:      nop     dword ptr [rax]
0x16d50:      endbr64
0x16d54:      push    r14
0x16d56:      push    r13
0x16d58:      push    r12
0x16d5a:      push    rbp
0x16d5b:      push    rbx
0x16d5c:      sub     rsp, 0x10
0x16d60:      mov     rax, qword ptr fs:[0x28]
0x16d69:      mov     qword ptr [rsp + 8], rax
0x16d6e:      xor     eax, eax
0x16d70:      lea     rbp, [rsp + 7]
0x16d75:      mov     rdi, rbp
0x16d78:      call    0xa6a0
0x16d7d:      mov     edx, dword ptr [rip + 0x138c1]
0x16d83:      mov     esi, dword ptr [rip + 0x138b7]
0x16d89:      mov     edi, dword ptr [rip + 0x138b9]
0x16d8f:      call    0xa570
0x16d94:      mov     rdi, qword ptr [rip + 0x1489d]
0x16d9b:      xor     edx, edx
0x16d9d:      xor     esi, esi
0x16d9f:      call    0xc680
0x16da4:      mov     edx, dword ptr [rip + 0x148ae]
0x16daa:      mov     rsi, qword ptr [rip + 0x148af]
0x16db1:      mov     rdi, qword ptr [rip + 0x14880]
0x16db8:      call    0xc600
0x16dbd:      mov     rax, qword ptr [rip + 0x14874]
0x16dc4:      mov     edi, dword ptr [rax + 0xc]
0x16dc7:      add     edi, 1
0x16dca:      movsxd  rdi, edi
0x16dcd:      call    0x1d670
0x16dd2:      mov     r13, rax
0x16dd5:      xor     eax, eax
0x16dd7:      call    0x16980
0x16ddc:      mov     r14, qword ptr [rip + 0x14855]
0x16de3:      mov     eax, dword ptr [rip + 0x1486f]
0x16de9:      mov     ecx, dword ptr [rip + 0x1485d]
0x16def:      mov     edx, dword ptr [r14 + 0xc]
0x16df3:      lea     esi, [rax + rcx + 2]
0x16df7:      cmp     esi, edx
0x16df9:      jl      0x16e58

```

```

0x16dfb:      cmp     eax, edx
0x16dfd:      jl      0x16e30
0x16dff:      mov     rdi, r13
0x16e02:      call    0x18d10
0x16e07:      mov     rdi, rbp
0x16e0a:      call    0xa6b0
0x16e0f:      mov     rax, qword ptr [rsp + 8]
0x16e14:      sub     rax, qword ptr fs:[0x28]
0x16e1d:      jne     0x16fc7
0x16e23:      add     rsp, 0x10
0x16e27:      pop     rbx
0x16e28:      pop     rbp
0x16e29:      pop     r12
0x16e2b:      pop     r13
0x16e2d:      pop     r14
0x16e2f:      ret
0x16e30:      sub     edx, eax
0x16e32:      mov     rdi, r13
0x16e35:      mov     esi, 0x20
0x16e3a:      mov     r12d, edx
0x16e3d:      movsxd  rdx, edx
0x16e40:      call    0x4a50
0x16e45:      mov     edx, r12d
0x16e48:      mov     rsi, r13
0x16e4b:      mov     rdi, r14
0x16e4e:      call    0xc600
0x16e53:      jmp     0x16dff
0x16e55:      nop     dword ptr [rax]
0x16e58:      sub     edx, eax
0x16e5a:      mov     rdi, r13
0x16e5d:      mov     esi, 0x20
0x16e62:      mov     ebx, edx
0x16e64:      sub     ebx, ecx
0x16e66:      movsxd  rdx, ebx
0x16e69:      call    0x4a50
0x16e6e:      mov     edx, ebx
0x16e70:      mov     rsi, r13
0x16e73:      mov     rdi, r14
0x16e76:      call    0xc600
0x16e7b:      add     ebx, dword ptr [rip + 0x147d7]
0x16e81:      mov     rdi, qword ptr [rip + 0x147b0]
0x16e88:      xor     esi, esi
0x16e8a:      mov     edx, ebx
0x16e8c:      call    0xc680
0x16e91:      mov     rdi, qword ptr [rip + 0x147a0]
0x16e98:      mov     edx, 5
0x16e9d:      lea     rsi, [rip + 0x13793]
0x16ea4:      call    0xc600

```

```

0x16ea9:    mov     rdi, qword ptr [rip + 0x14788]
0x16eb0:    mov     esi, 0x20
0x16eb5:    call    0xc650
0x16eba:    mov     edi, dword ptr [rip + 0x1378c]
0x16ec0:    call    0xa4f0
0x16ec5:    mov     edx, dword ptr [rip + 0x14789]
0x16ecb:    mov     rsi, qword ptr [rip + 0x14bc6]
0x16ed2:    mov     rdi, qword ptr [rip + 0x1475f]
0x16ed9:    call    0xc600
0x16ede:    mov     rdi, qword ptr [rip + 0x14753]
0x16ee5:    mov     esi, 0x20
0x16eea:    call    0xc650
0x16eef:    mov     r12, qword ptr [rip + 0x1376a]
0x16ef6:    mov     rdi, r12
0x16ef9:    call    0x49b0
0x16efe:    mov     rdi, qword ptr [rip + 0x14733]
0x16f05:    mov     rsi, r12
0x16f08:    mov     edx, eax
0x16f0a:    call    0xc600
0x16f0f:    mov     rdi, qword ptr [rip + 0x14722]
0x16f16:    mov     esi, 0x20
0x16f1b:    call    0xc650
0x16f20:    mov     edi, dword ptr [rip + 0x1371a]
0x16f26:    call    0xa4f0
0x16f2b:    mov     rdi, qword ptr [rip + 0x14706]
0x16f32:    mov     edx, 4
0x16f37:    lea     rsi, [rip + 0x136f4]
0x16f3e:    call    0xc600
0x16f43:    mov     rdi, qword ptr [rip + 0x146ee]
0x16f4a:    mov     esi, 0x20
0x16f4f:    call    0xc650
0x16f54:    mov     edi, dword ptr [rip + 0x136f6]
0x16f5a:    call    0xa4f0
0x16f5f:    mov     edx, dword ptr [rip + 0x146eb]
0x16f65:    mov     rsi, qword ptr [rip + 0x13a24]
0x16f6c:    mov     rdi, qword ptr [rip + 0x146c5]
0x16f73:    call    0xc600
0x16f78:    mov     edi, dword ptr [rip + 0x136c2]
0x16f7e:    call    0xa4f0
0x16f83:    mov     rdi, qword ptr [rip + 0x146ae]
0x16f8a:    mov     esi, 0x20
0x16f8f:    call    0xc650
0x16f94:    xor     edi, edi
0x16f96:    xor     eax, eax
0x16f98:    call    0x16fd0
0x16f9d:    mov     rdi, qword ptr [rip + 0x14694]
0x16fa4:    xor     esi, esi
0x16fa6:    mov     eax, dword ptr [rdi + 0xc]

```



```

0x16fa9:    lea     edx, [rax - 1]
0x16fac:    call    0xc680
0x16fb1:    mov     rdi, qword ptr [rip + 0x14680]
0x16fb8:    mov     esi, 0x20
0x16fbd:    call    0xc650
0x16fc2:    jmp     0x16dff
0x16fc7:    call    0x49d0
0x16fcc:    nop     dword ptr [rax]
0x16fd0:    endbr64
0x16fd4:    push    r12
0x16fd6:    push    rbp
0x16fd7:    push    rbx
0x16fd8:    mov     ebx, edi
0x16fda:    sub     rsp, 0x30
0x16fde:    mov     rax, qword ptr fs:[0x28]
0x16fe7:    mov     qword ptr [rsp + 0x28], rax
0x16fec:    xor     eax, eax
0x16fee:    call    0x7870
0x16ff3:    test    eax, eax
0x16ff5:    jne     0x17114
0x16ffb:    mov     eax, dword ptr [rip + 0x1464b]
0x17001:    mov     edx, dword ptr [rip + 0x14651]
0x17007:    lea     edx, [rdx + rax + 2]
0x1700b:    mov     rax, qword ptr [rip + 0x14626]
0x17012:    cmp     edx, dword ptr [rax + 0xc]
0x17015:    jge     0x17114
0x1701b:    test    ebx, ebx
0x1701d:    jne     0x17140
0x17023:    lea     r12, [rsp + 7]
0x17028:    xor     eax, eax
0x1702a:    call    0x18330
0x1702f:    mov     rdi, r12
0x17032:    mov     rbp, rax
0x17035:    call    0xa6a0
0x1703a:    lea     rsi, [rsp + 0xc]
0x1703f:    lea     rdi, [rsp + 8]
0x17044:    call    0xa4d0
0x17049:    xor     edi, edi
0x1704b:    call    0xa660
0x17050:    movsxd  r8, dword ptr [rbp + 8]
0x17054:    mov     r9d, dword ptr [rbp + 4]
0x17058:    lea     rbp, [rsp + 0x10]
0x1705d:    lea     rcx, [rip + 0xa0d4]
0x17064:    mov     esi, 1
0x17069:    mov     rdi, rbp
0x1706c:    mov     rax, r8
0x1706f:    imul    r8, r8, 0x2aaaaaab
0x17076:    cdq

```

```

0x17077:      sar      r8, 0x21
0x1707b:      sub      r8d, edx
0x1707e:      lea      edx, [r8 + r8*2]
0x17082:      mov      r8d, eax
0x17085:      shl      edx, 2
0x17088:      sub      r8d, edx
0x1708b:      mov      edx, 0xc
0x17090:      cmov     r8d, edx
0x17094:      cmp      eax, 0xc
0x17097:      mov      edx, 0x70
0x1709c:      mov      eax, 0x61
0x170a1:      cmovge   eax, edx
0x170a4:      sub      rsp, 8
0x170a8:      mov      edx, 0x10
0x170ad:      push     rax
0x170ae:      xor      eax, eax
0x170b0:      call     0x4f10
0x170b5:      mov      rdi, qword ptr [rip + 0x1457c]
0x170bc:      xor      esi, esi
0x170be:      mov      eax, dword ptr [rdi + 0xc]
0x170c1:      lea      edx, [rax - 7]
0x170c4:      call     0xc680
0x170c9:      mov      edx, dword ptr [rip + 0x13589]
0x170cf:      mov      esi, dword ptr [rip + 0x1357f]
0x170d5:      mov      edi, dword ptr [rip + 0x13581]
0x170db:      call     0xa570
0x170e0:      mov      rdi, rbp
0x170e3:      call     0x49b0
0x170e8:      mov      rdi, qword ptr [rip + 0x14549]
0x170ef:      mov      rsi, rbp
0x170f2:      mov      edx, eax
0x170f4:      call     0xc600
0x170f9:      mov      esi, dword ptr [rsp + 0x1c]
0x170fd:      mov      edi, dword ptr [rsp + 0x18]
0x17101:      call     0xa4b0
0x17106:      mov      rdi, r12
0x17109:      call     0xa6b0
0x1710e:      pop      rax
0x1710f:      pop      rdx
0x17110:      test     ebx, ebx
0x17112:      jne      0x17130
0x17114:      mov      rax, qword ptr [rsp + 0x28]
0x17119:      sub      rax, qword ptr fs:[0x28]
0x17122:      jne      0x1715b
0x17124:      add      rsp, 0x30
0x17128:      pop      rbx
0x17129:      pop      rbp
0x1712a:      pop      r12

```

```

0x1712c:    ret
0x1712d:    nop     dword ptr [rax]
0x17130:    xor     eax, eax
0x17132:    call    0xa820
0x17137:    jmp     0x17114
0x17139:    nop     dword ptr [rax]
0x17140:    xor     eax, eax
0x17142:    call    0x16980
0x17147:    test    eax, eax
0x17149:    je      0x17023
0x1714f:    xor     eax, eax
0x17151:    call    0x16d50
0x17156:    jmp     0x17023
0x1715b:    call    0x49d0
0x17160:    push    rbp
0x17161:    mov     esi, 0x400
0x17166:    push    rbx
0x17167:    lea     rbx, [rip + 0x14512]
0x1716e:    mov     rdi, rbx
0x17171:    sub     rsp, 8
0x17175:    mov     rdx, qword ptr [rip + 0x1490c]
0x1717c:    call    0x4b00
0x17181:    test    rax, rax
0x17184:    je      0x171d8
0x17186:    mov     rdi, rbx
0x17189:    call    0x49b0
0x1718e:    mov     rbp, rax
0x17191:    cmp     rax, 0x3ff
0x17197:    je      0x171e8
0x17199:    mov     esi, 0x23
0x1719e:    mov     rdi, rbx
0x171a1:    call    0x4a00
0x171a6:    test    rax, rax
0x171a9:    je      0x171c0
0x171ab:    mov     byte ptr [rax], 0
0x171ae:    mov     eax, 1
0x171b3:    add     rsp, 8
0x171b7:    pop     rbx
0x171b8:    pop     rbp
0x171b9:    ret
0x171ba:    nop     word ptr [rax + rax]
0x171c0:    cmp     byte ptr [rbx + rbp - 1], 0xa
0x171c5:    mov     eax, 1
0x171ca:    jne     0x171b3
0x171cc:    mov     byte ptr [rbx + rbp - 1], 0
0x171d1:    jmp     0x171b3
0x171d3:    nop     dword ptr [rax + rax]
0x171d8:    add     rsp, 8

```

```

0x171dc:    xor     eax, eax
0x171de:    pop     rbx
0x171df:    pop     rbp
0x171e0:    ret
0x171e1:    nop     dword ptr [rax]
0x171e8:    mov     rcx, qword ptr [rip + 0x136f9]
0x171ef:    mov     rdi, qword ptr [rip + 0x1368a]
0x171f6:    lea     rdx, [rip + 0xaceb]
0x171fd:    xor     eax, eax
0x171ff:    mov     esi, 1
0x17204:    call    0x4e60
0x17209:    nop     dword ptr [rax]
0x17210:    mov     rdi, qword ptr [rip + 0x14871]
0x17217:    call    0x4a90
0x1721c:    cmp     eax, 0xa
0x1721f:    je      0x17199
0x17225:    cmp     eax, -1
0x17228:    jne     0x17210
0x1722a:    jmp     0x17199
0x1722f:    nop
0x17230:    endbr64
0x17234:    test    rdi, rdi
0x17237:    je      0x17270
0x17239:    sub     rsp, 8
0x1723d:    lea     rsi, [rip + 0xad85]
0x17244:    call    0x4da0
0x17249:    mov     qword ptr [rip + 0x14838], rax
0x17250:    test    rax, rax
0x17253:    je      0x17269
0x17255:    mov     dword ptr [rip + 0x14821], 0xffffffff
0x1725f:    mov     eax, 1
0x17264:    add     rsp, 8
0x17268:    ret
0x17269:    xor     eax, eax
0x1726b:    add     rsp, 8
0x1726f:    ret
0x17270:    xor     eax, eax
0x17272:    ret
0x17273:    nop     word ptr cs:[rax + rax]
0x1727e:    nop
0x17280:    endbr64
0x17284:    mov     rdi, qword ptr [rip + 0x147fd]
0x1728b:    test    rdi, rdi
0x1728e:    je      0x17298
0x17290:    jmp     0x4940
0x17295:    nop     dword ptr [rax]
0x17298:    ret
0x17299:    nop     dword ptr [rax]

```

```

0x172a0:      endbr64
0x172a4:      push    rbp
0x172a5:      xor     edx, edx
0x172a7:      xor     esi, esi
0x172a9:      lea     rbp, [rip + 0x143d0]
0x172b0:      push    rbx
0x172b1:      mov     rbx, rdi
0x172b4:      sub     rsp, 8
0x172b8:      mov     rdi, qword ptr [rip + 0x147c9]
0x172bf:      call    0x4c80
0x172c4:      jmp     0x172df
0x172c6:      nop     word ptr cs:[rax + rax]
0x172d0:      mov     rsi, rbp
0x172d3:      mov     rdi, rbx
0x172d6:      call    0x4b30
0x172db:      test    eax, eax
0x172dd:      je      0x17300
0x172df:      xor     eax, eax
0x172e1:      call    0x17160
0x172e6:      test    eax, eax
0x172e8:      jne     0x172d0
0x172ea:      mov     eax, 0xffffffff
0x172ef:      mov     dword ptr [rip + 0x1478b], eax
0x172f5:      add     rsp, 8
0x172f9:      pop     rbx
0x172fa:      pop     rbp
0x172fb:      ret
0x172fc:      nop     dword ptr [rax]
0x17300:      mov     rdi, qword ptr [rip + 0x14781]
0x17307:      call    0x4b80
0x1730c:      mov     dword ptr [rip + 0x1476e], eax
0x17312:      add     rsp, 8
0x17316:      pop     rbx
0x17317:      pop     rbp
0x17318:      ret
0x17319:      nop     dword ptr [rax]
0x17320:      endbr64
0x17324:      push    r15
0x17326:      push    r14
0x17328:      push    r13
0x1732a:      mov     r13, rdi
0x1732d:      push    r12
0x1732f:      push    rbp
0x17330:      push    rbx
0x17331:      mov     ebx, ecx
0x17333:      sub     rsp, 0x438
0x1733a:      mov     qword ptr [rsp + 0x10], rsi
0x1733f:      mov     dword ptr [rsp + 0xc], edx

```

```

0x17343:    mov     rax, qword ptr fs:[0x28]
0x1734c:    mov     qword ptr [rsp + 0x428], rax
0x17354:    xor     eax, eax
0x17356:    test    ecx, ecx
0x17358:    jne     0x17618
0x1735e:    cmp     dword ptr [rsp + 0xc], 1
0x17363:    jne     0x175e0
0x17369:    mov     rax, qword ptr [rsp + 0x10]
0x1736e:    mov     qword ptr [rax], 0
0x17375:    lea     rbp, [rip + 0x14304]
0x1737c:    lea     r12, [rsp + 0x20]
0x17381:    lea     r14, [rip + 0xa092]
0x17388:    nop     dword ptr [rax + rax]
0x17390:    xor     eax, eax
0x17392:    call    0x17160
0x17397:    test    eax, eax
0x17399:    je      0x17600
0x1739f:    cmp     byte ptr [rbp], 0
0x173a3:    je      0x17600
0x173a9:    mov     rdx, r12
0x173ac:    mov     rsi, r14
0x173af:    mov     rdi, rbp
0x173b2:    xor     eax, eax
0x173b4:    mov     byte ptr [rsp + 0x20], 0
0x173b9:    call    0x4c60
0x173be:    call    0x4f00
0x173c3:    mov     r15, rax
0x173c6:    test    ebx, ebx
0x173c8:    jne     0x17410
0x173ca:    movsx    rcx, byte ptr [rsp + 0x20]
0x173d0:    mov     rax, qword ptr [rax]
0x173d3:    mov     byte ptr [rsp + 0x40], 0
0x173d8:    test    byte ptr [rax + rcx*2 + 1], 0x40
0x173dd:    jne     0x17433
0x173df:    mov     rax, qword ptr [rsp + 0x428]
0x173e7:    sub     rax, qword ptr fs:[0x28]
0x173f0:    jne     0x17657
0x173f6:    add     rsp, 0x438
0x173fd:    pop     rbx
0x173fe:    pop     rbp
0x173ff:    pop     r12
0x17401:    pop     r13
0x17403:    pop     r14
0x17405:    pop     r15
0x17407:    ret
0x17408:    nop     dword ptr [rax + rax]
0x17410:    movsx    rcx, byte ptr [rsp + 0x20]
0x17416:    mov     rax, qword ptr [rax]

```

```

0x17419:    test    byte ptr [rax + rcx*2 + 1], 0x40
0x1741e:    je      0x173df
0x17420:    mov     rsi, r12
0x17423:    mov     rdi, r13
0x17426:    call    0x4b30
0x1742b:    test    eax, eax
0x1742d:    jne     0x17390
0x17433:    mov     esi, 0x3d
0x17438:    mov     rdi, rbp
0x1743b:    call    0x4a00
0x17440:    test    rax, rax
0x17443:    je      0x174e3
0x17449:    cmp     byte ptr [rax + 1], 0
0x1744d:    je      0x174e3
0x17453:    mov     rdx, qword ptr [rsp + 0x10]
0x17458:    lea     rdi, [rax + 1]
0x1745c:    mov     qword ptr [rdx], rdi
0x1745f:    movzx   ecx, byte ptr [rax + 1]
0x17463:    test    cl, cl
0x17465:    je      0x1763f
0x1746b:    cmp     dword ptr [rsp + 0xc], 1
0x17470:    jle     0x1763f
0x17476:    mov     r8d, 1
0x1747c:    jmp     0x1748e
0x1747e:    nop
0x17480:    mov     ecx, eax
0x17482:    cmp     r8d, dword ptr [rsp + 0xc]
0x17487:    jge     0x174d1
0x17489:    test    sil, sil
0x1748c:    je      0x174d1
0x1748e:    movzx   eax, byte ptr [rdi + 1]
0x17492:    test    al, al
0x17494:    setne   sil
0x17498:    add     rdi, 1
0x1749c:    cmp     cl, 0x3b
0x1749f:    jne     0x17480
0x174a1:    mov     byte ptr [rdi - 1], 0
0x174a5:    cmp     al, 0x3b
0x174a7:    je      0x17638
0x174ad:    test    sil, sil
0x174b0:    je      0x17638
0x174b6:    mov     rdx, qword ptr [rsp + 0x10]
0x174bb:    movsxd  rax, r8d
0x174be:    mov     qword ptr [rdx + rax*8], rdi
0x174c2:    movzx   ecx, byte ptr [rdi]
0x174c5:    test    cl, cl
0x174c7:    setne   sil
0x174cb:    add     r8d, 1

```

```

0x174cf:      jmp      0x17482
0x174d1:      mov      esi, 0x3b
0x174d6:      call     0x4a00
0x174db:      test     rax, rax
0x174de:      je        0x174ef
0x174e0:      mov      byte ptr [rax], 0
0x174e3:      mov      eax, dword ptr [rsp + 0xc]
0x174e7:      test     eax, eax
0x174e9:      jle       0x175c0
0x174ef:      mov      rsi, qword ptr [rsp + 0x10]
0x174f4:      xor      r8d, r8d
0x174f7:      jmp      0x17513
0x174f9:      nop       dword ptr [rax]
0x17500:      add      r8d, 1
0x17504:      add      rsi, 8
0x17508:      cmp      dword ptr [rsp + 0xc], r8d
0x1750d:      jle       0x175c0
0x17513:      mov      rbp, qword ptr [rsi]
0x17516:      test     rbp, rbp
0x17519:      je        0x17500
0x1751b:      mov      r14, qword ptr [r15]
0x1751e:      movsx    rax, byte ptr [rbp]
0x17523:      test     byte ptr [r14 + rax*2 + 1], 0x20
0x17529:      je        0x17547
0x1752b:      lea      rax, [rbp + 1]
0x1752f:      nop
0x17530:      mov      rbp, rax
0x17533:      mov      qword ptr [rsi], rax
0x17536:      add      rax, 1
0x1753a:      movsx    rcx, byte ptr [rbp]
0x1753f:      test     byte ptr [r14 + rcx*2 + 1], 0x20
0x17545:      jne       0x17530
0x17547:      mov      rdi, rbp
0x1754a:      mov      qword ptr [rsp + 0x18], rsi
0x1754f:      mov      dword ptr [rsp + 0x10], r8d
0x17554:      call     0x49b0
0x17559:      mov      r8d, dword ptr [rsp + 0x10]
0x1755e:      mov      rsi, qword ptr [rsp + 0x18]
0x17563:      lea      rax, [rbp + rax - 1]
0x17568:      movsx    rcx, byte ptr [rax]
0x1756c:      test     byte ptr [r14 + rcx*2 + 1], 0x20
0x17572:      jne       0x17591
0x17574:      jmp      0x17596
0x17576:      nop       word ptr cs:[rax + rax]
0x17580:      movsx    rcx, byte ptr [rax - 1]
0x17585:      sub      rax, 1
0x17589:      test     byte ptr [r14 + rcx*2 + 1], 0x20
0x1758f:      je        0x17596

```



```

0x17591:      cmp     rax, rbp
0x17594:      jae     0x17580
0x17596:      mov     byte ptr [rax + 1], 0
0x1759a:      mov     rax, qword ptr [rsi]
0x1759d:      cmp     byte ptr [rax], 0
0x175a0:      jne     0x17500
0x175a6:      mov     qword ptr [rsi], 0
0x175ad:      add     r8d, 1
0x175b1:      add     rsi, 8
0x175b5:      cmp     dword ptr [rsp + 0xc], r8d
0x175ba:      jg       0x17513
0x175c0:      test    ebx, ebx
0x175c2:      jne     0x173df
0x175c8:      mov     rsi, r12
0x175cb:      mov     rdi, r13
0x175ce:      call    0x48b0
0x175d3:      jmp     0x173df
0x175d8:      nop     dword ptr [rax + rax]
0x175e0:      movsxd  rdx, dword ptr [rsp + 0xc]
0x175e5:      mov     rdi, qword ptr [rsp + 0x10]
0x175ea:      xor     esi, esi
0x175ec:      shl     rdx, 3
0x175f0:      call    0x4a50
0x175f5:      jmp     0x17375
0x175fa:      nop     word ptr [rax + rax]
0x17600:      test    ebx, ebx
0x17602:      jne     0x173df
0x17608:      mov     byte ptr [r13], 0
0x1760d:      jmp     0x173df
0x17612:      nop     word ptr [rax + rax]
0x17618:      movsxd  rsi, dword ptr [rip + 0x14461]
0x1761f:      mov     rdi, qword ptr [rip + 0x14462]
0x17626:      xor     edx, edx
0x17628:      call    0x4c80
0x1762d:      jmp     0x1735e
0x17632:      nop     word ptr [rax + rax]
0x17638:      mov     ecx, eax
0x1763a:      jmp     0x174cb
0x1763f:      mov     esi, 0x3b
0x17644:      call    0x4a00
0x17649:      test    rax, rax
0x1764c:      jne     0x174e0
0x17652:      jmp     0x174e3
0x17657:      call    0x49d0
0x1765c:      nop     dword ptr [rax]
0x17660:      endbr64
0x17664:      push    r12
0x17666:      mov     edi, 0xf

```

```

0x1766b:    lea     r12, [rip + 0xa8b4]
0x17672:    call   0x1d670
0x17677:    mov     rcx, r12
0x1767a:    lea     r9, [rip + 0xa895]
0x17681:    lea     r8, [rip + 0xa893]
0x17688:    mov     rdi, rax
0x1768b:    mov     rdx, -1
0x17692:    mov     esi, 1
0x17697:    mov     qword ptr [rip + 0x14402], rax
0x1769e:    xor     eax, eax
0x176a0:    call   0x4f10
0x176a5:    mov     edi, 0x17
0x176aa:    call   0x1d670
0x176af:    mov     rcx, r12
0x176b2:    pop     r12
0x176b4:    lea     r9, [rip + 0xa870]
0x176bb:    mov     qword ptr [rip + 0x143e6], rax
0x176c2:    mov     rdi, rax
0x176c5:    xor     eax, eax
0x176c7:    lea     r8, [rip + 0xa84d]
0x176ce:    mov     rdx, -1
0x176d5:    mov     esi, 1
0x176da:    jmp     0x4f10
0x176df:    nop
0x176e0:    endbr64
0x176e4:    push    rbp
0x176e5:    mov     rbp, rdi
0x176e8:    xor     edi, edi
0x176ea:    push    rbx
0x176eb:    sub     rsp, 0x18
0x176ef:    mov     rax, qword ptr fs:[0x28]
0x176f8:    mov     qword ptr [rsp + 8], rax
0x176fd:    xor     eax, eax
0x176ff:    call   0x4be0
0x17704:    mov     rdi, rsp
0x17707:    mov     qword ptr [rsp], rax
0x1770b:    call   0x4840
0x17710:    mov     rbx, rax
0x17713:    call   0x4920
0x17718:    movsxd  rcx, dword ptr [rbx + 0x10]
0x1771c:    sub     rsp, 8
0x17720:    lea     rdx, [rip + 0x12fd9]
0x17727:    mov     r9d, dword ptr [rbx + 8]
0x1772b:    mov     r8d, dword ptr [rbx + 0xc]
0x1772f:    mov     esi, 1
0x17734:    mov     rcx, qword ptr [rdx + rcx*8]
0x17738:    push    rbp
0x17739:    lea     rdx, [rip + 0xa8d0]

```

```

0x17740:    push    rax
0x17741:    mov     rdi, qword ptr [rip + 0x13138]
0x17748:    push    qword ptr [rip + 0x1319a]
0x1774e:    mov     eax, dword ptr [rbx]
0x17750:    push    rax
0x17751:    mov     eax, dword ptr [rbx + 4]
0x17754:    push    rax
0x17755:    xor     eax, eax
0x17757:    call    0x4e60
0x1775c:    add     rsp, 0x30
0x17760:    mov     rax, qword ptr [rsp + 8]
0x17765:    sub     rax, qword ptr fs:[0x28]
0x1776e:    jne     0x17777
0x17770:    add     rsp, 0x18
0x17774:    pop     rbx
0x17775:    pop     rbp
0x17776:    ret
0x17777:    call    0x49d0
0x1777c:    nop     dword ptr [rax]
0x17780:    endbr64
0x17784:    push    rbp
0x17785:    xor     eax, eax
0x17787:    mov     ebp, edi
0x17789:    call    0x8ad0
0x1778e:    cmp     ebp, 0xb
0x17791:    je      0x177d3
0x17793:    jg      0x177c0
0x17795:    lea     eax, [rbp - 1]
0x17798:    cmp     eax, 1
0x1779b:    ja      0x17802
0x1779d:    cmp     ebp, 1
0x177a0:    lea     rdi, [rip + 0xa791]
0x177a7:    lea     rax, [rip + 0xa78e]
0x177ae:    cmovne  rdi, rax
0x177b2:    call    0x176e0
0x177b7:    mov     edi, ebp
0x177b9:    call    0x4e20
0x177be:    nop
0x177c0:    cmp     ebp, 0xf
0x177c3:    jne     0x17802
0x177c5:    lea     rdi, [rip + 0x8dbe]
0x177cc:    call    0x176e0
0x177d1:    jmp     0x177b7
0x177d3:    lea     rdi, [rip + 0xa766]
0x177da:    call    0x176e0
0x177df:    mov     rcx, qword ptr [rip + 0x13102]
0x177e6:    mov     rdi, qword ptr [rip + 0x13093]
0x177ed:    lea     rdx, [rip + 0xa88c]

```

```

0x177f4:      xor     eax, eax
0x177f6:      mov     esi, 1
0x177fb:      call    0x4e60
0x17800:      jmp     0x177b7
0x17802:      mov     rcx, qword ptr [rip + 0x130df]
0x17809:      mov     rdi, qword ptr [rip + 0x13070]
0x17810:      mov     r8d, ebp
0x17813:      xor     eax, eax
0x17815:      lea     rdx, [rip + 0xa82c]
0x1781c:      mov     esi, 1
0x17821:      call    0x4e60
0x17826:      jmp     0x177df
0x17828:      nop     dword ptr [rax + rax]
0x17830:      endbr64
0x17834:      push    rbx
0x17835:      mov     rcx, qword ptr [rip + 0x130ac]
0x1783c:      mov     rbx, rdi
0x1783f:      mov     esi, 1
0x17844:      mov     rdi, qword ptr [rip + 0x13035]
0x1784b:      lea     rdx, [rip + 0xa6f3]
0x17852:      xor     eax, eax
0x17854:      call    0x4e60
0x17859:      mov     r8, rbx
0x1785c:      mov     rcx, qword ptr [rip + 0x13085]
0x17863:      pop     rbx
0x17864:      mov     rdi, qword ptr [rip + 0x13015]
0x1786b:      lea     rdx, [rip + 0xa836]
0x17872:      mov     esi, 1
0x17877:      xor     eax, eax
0x17879:      jmp     0x4e60
0x1787e:      nop
0x17880:      endbr64
0x17884:      sub     rsp, 8
0x17888:      mov     r8, rdi
0x1788b:      mov     rcx, qword ptr [rip + 0x13056]
0x17892:      xor     eax, eax
0x17894:      mov     rdi, qword ptr [rip + 0x12fe5]
0x1789b:      lea     rdx, [rip + 0xa836]
0x178a2:      mov     esi, 1
0x178a7:      call    0x4e60
0x178ac:      mov     rcx, qword ptr [rip + 0x13035]
0x178b3:      mov     esi, 1
0x178b8:      xor     eax, eax
0x178ba:      mov     rdi, qword ptr [rip + 0x12fbf]
0x178c1:      lea     rdx, [rip + 0xa840]
0x178c8:      call    0x4e60
0x178cd:      mov     rcx, qword ptr [rip + 0x13014]
0x178d4:      mov     esi, 1

```

```

0x178d9:    xor     eax, eax
0x178db:    mov     rdi, qword ptr [rip + 0x12f9e]
0x178e2:    lea     rdx, [rip + 0xa84f]
0x178e9:    call    0x4e60
0x178ee:    mov     rcx, qword ptr [rip + 0x12ff3]
0x178f5:    mov     esi, 1
0x178fa:    xor     eax, eax
0x178fc:    mov     rdi, qword ptr [rip + 0x12f7d]
0x17903:    lea     rdx, [rip + 0xa86e]
0x1790a:    call    0x4e60
0x1790f:    mov     rcx, qword ptr [rip + 0x12fd2]
0x17916:    mov     esi, 1
0x1791b:    xor     eax, eax
0x1791d:    mov     rdi, qword ptr [rip + 0x12f5c]
0x17924:    lea     rdx, [rip + 0xa885]
0x1792b:    call    0x4e60
0x17930:    mov     rcx, qword ptr [rip + 0x12fb1]
0x17937:    mov     esi, 1
0x1793c:    xor     eax, eax
0x1793e:    mov     rdi, qword ptr [rip + 0x12f3b]
0x17945:    lea     rdx, [rip + 0xa8a4]
0x1794c:    call    0x4e60
0x17951:    mov     rcx, qword ptr [rip + 0x12f90]
0x17958:    mov     esi, 1
0x1795d:    xor     eax, eax
0x1795f:    mov     rdi, qword ptr [rip + 0x12f1a]
0x17966:    lea     rdx, [rip + 0xa8c3]
0x1796d:    call    0x4e60
0x17972:    mov     rcx, qword ptr [rip + 0x12f6f]
0x17979:    mov     esi, 1
0x1797e:    xor     eax, eax
0x17980:    mov     rdi, qword ptr [rip + 0x12ef9]
0x17987:    lea     rdx, [rip + 0xa8ea]
0x1798e:    add     rsp, 8
0x17992:    jmp     0x4e60
0x17997:    nop     word ptr [rax + rax]
0x179a0:    endbr64
0x179a4:    push    r12
0x179a6:    lea     r12, [rip + 0x12d33]
0x179ad:    push    rbp
0x179ae:    sub     rsp, 8
0x179b2:    mov     rdi, qword ptr [rip + 0x140ef]
0x179b9:    call    0x49b0
0x179be:    mov     rdi, r12
0x179c1:    mov     rbp, rax
0x179c4:    call    0x49b0
0x179c9:    lea     rdi, [rbp + rax + 9]
0x179ce:    call    0x1d670

```

```

0x179d3:    mov     rsi, qword ptr [rip + 0x140ce]
0x179da:    mov     rdi, rax
0x179dd:    mov     rbp, rax
0x179e0:    call    0x4980
0x179e5:    mov     rsi, r12
0x179e8:    mov     rdi, rax
0x179eb:    call    0x4980
0x179f0:    mov     edx, 0x6e6f
0x179f5:    mov     rdi, rbp
0x179f8:    mov     dword ptr [rax], 0x6d6d6f63
0x179fe:    mov     word ptr [rax + 4], dx
0x17a02:    mov     byte ptr [rax + 6], 0
0x17a06:    call    0x17230
0x17a0b:    test    eax, eax
0x17a0d:    je      0x17a17
0x17a0f:    add     rsp, 8
0x17a13:    pop     rbp
0x17a14:    pop     r12
0x17a16:    ret
0x17a17:    mov     rdi, rbp
0x17a1a:    call    0x17830
0x17a1f:    mov     edi, 1
0x17a24:    call    0x4e20
0x17a29:    nop     dword ptr [rax]
0x17a30:    endbr64
0x17a34:    push    r13
0x17a36:    push    r12
0x17a38:    lea     r12, [rip + 0x12c91]
0x17a3f:    push    rbp
0x17a40:    push    rbx
0x17a41:    sub     rsp, 8
0x17a45:    mov     rdi, qword ptr [rip + 0x12ea4]
0x17a4c:    call    0x49b0
0x17a51:    mov     rdi, r12
0x17a54:    mov     rbx, rax
0x17a57:    call    0x49b0
0x17a5c:    mov     rdi, qword ptr [rip + 0x12eed]
0x17a63:    mov     rbp, rax
0x17a66:    call    0x49b0
0x17a6b:    lea     rdi, [rbx + rbp + 2]
0x17a70:    add     rdi, rax
0x17a73:    call    0x1d670
0x17a78:    mov     rsi, qword ptr [rip + 0x12e71]
0x17a7f:    mov     rdi, rax
0x17a82:    mov     rbp, rax
0x17a85:    call    0x4980
0x17a8a:    mov     rsi, r12
0x17a8d:    mov     rdi, rax

```

```

0x17a90:    call    0x4980
0x17a95:    mov     rsi, qword ptr [rip + 0x12eb4]
0x17a9c:    mov     rdi, rax
0x17a9f:    call    0x48b0
0x17aa4:    mov     rdi, rbp
0x17aa7:    call    0x17230
0x17aac:    test    eax, eax
0x17aae:    jne     0x17b2a
0x17ab0:    mov     rdi, rbp
0x17ab3:    lea     r12, [rip + 0x12c06]
0x17aba:    call    0x18d10
0x17abf:    mov     rdi, qword ptr [rip + 0x12e2a]
0x17ac6:    call    0x49b0
0x17acb:    mov     rdi, r12
0x17ace:    mov     rbp, rax
0x17ad1:    call    0x49b0
0x17ad6:    mov     rdi, qword ptr [rip + 0x12e73]
0x17add:    mov     rbx, rax
0x17ae0:    call    0x49b0
0x17ae5:    lea     rdi, [rbp + rbx + 2]
0x17aea:    add     rdi, rax
0x17aed:    call    0x1d670
0x17af2:    mov     rsi, qword ptr [rip + 0x12df7]
0x17af9:    mov     rdi, rax
0x17afc:    mov     rbp, rax
0x17aff:    call    0x4980
0x17b04:    mov     rsi, r12
0x17b07:    mov     rdi, rax
0x17b0a:    call    0x4980
0x17b0f:    mov     rsi, qword ptr [rip + 0x12e3a]
0x17b16:    mov     rdi, rax
0x17b19:    call    0x48b0
0x17b1e:    mov     rdi, rbp
0x17b21:    call    0x17230
0x17b26:    test    eax, eax
0x17b28:    je      0x17b50
0x17b2a:    mov     rdi, rbp
0x17b2d:    mov     r12d, 1
0x17b33:    call    0x18d10
0x17b38:    add     rsp, 8
0x17b3c:    mov     eax, r12d
0x17b3f:    pop     rbx
0x17b40:    pop     rbp
0x17b41:    pop     r12
0x17b43:    pop     r13
0x17b45:    ret
0x17b46:    nop     word ptr cs:[rax + rax]
0x17b50:    mov     rdi, rbp

```

```

0x17b53:    lea     r13, [rip + 0x12b86]
0x17b5a:    call   0x18d10
0x17b5f:    mov     rdi, qword ptr [rip + 0x13f42]
0x17b66:    call   0x49b0
0x17b6b:    mov     rdi, r13
0x17b6e:    mov     rbp, rax
0x17b71:    call   0x49b0
0x17b76:    mov     rdi, qword ptr [rip + 0x12dd3]
0x17b7d:    mov     rbx, rax
0x17b80:    call   0x49b0
0x17b85:    lea     rdi, [rbp + rbx + 2]
0x17b8a:    add     rdi, rax
0x17b8d:    call   0x1d670
0x17b92:    mov     rsi, qword ptr [rip + 0x13f0f]
0x17b99:    mov     rdi, rax
0x17b9c:    mov     rbp, rax
0x17b9f:    call   0x4980
0x17ba4:    mov     rsi, r13
0x17ba7:    mov     rdi, rax
0x17baa:    call   0x4980
0x17baf:    mov     rsi, qword ptr [rip + 0x12d9a]
0x17bb6:    mov     rdi, rax
0x17bb9:    call   0x48b0
0x17bbe:    mov     rdi, rbp
0x17bc1:    call   0x17230
0x17bc6:    mov     r12d, eax
0x17bc9:    test    eax, eax
0x17bcb:    jne     0x17b2a
0x17bd1:    xor     eax, eax
0x17bd3:    mov     rdi, rbp
0x17bd6:    call   0x17880
0x17bdb:    mov     rdi, rbp
0x17bde:    call   0x18d10
0x17be3:    mov     rdi, qword ptr [rip + 0x13ebe]
0x17bea:    call   0x49b0
0x17bef:    mov     rdi, r13
0x17bf2:    mov     rbx, rax
0x17bf5:    call   0x49b0
0x17bfa:    lea     rdi, [rbx + rax + 0xa]
0x17bff:    call   0x1d670
0x17c04:    mov     rsi, qword ptr [rip + 0x13e9d]
0x17c0b:    mov     rdi, rax
0x17c0e:    mov     rbp, rax
0x17c11:    call   0x4980
0x17c16:    mov     rsi, r13
0x17c19:    mov     rdi, rax
0x17c1c:    call   0x4980
0x17c21:    mov     rdi, rbp

```



```

0x17c24:      movabs  rdx, 0x636972656e6567
0x17c2e:      mov     qword ptr [rax], rdx
0x17c31:      call   0x17230
0x17c36:      test   eax, eax
0x17c38:      jne     0x17b38
0x17c3e:      mov     rdi, rbp
0x17c41:      call   0x17830
0x17c46:      mov     edi, 1
0x17c4b:      call   0x4e20
0x17c50:      endbr64
0x17c54:      push    rbp
0x17c55:      push    rbx
0x17c56:      mov     rbx, rdi
0x17c59:      sub     rsp, 8
0x17c5d:      call   0x172a0
0x17c62:      cmp     eax, -1
0x17c65:      je      0x17c70
0x17c67:      add     rsp, 8
0x17c6b:      pop     rbx
0x17c6c:      pop     rbp
0x17c6d:      ret
0x17c6e:      nop
0x17c70:      mov     edx, 7
0x17c75:      lea     rsi, [rip + 0xa2e5]
0x17c7c:      mov     rdi, rbx
0x17c7f:      call   0x4890
0x17c84:      test   eax, eax
0x17c86:      jne     0x17cd4
0x17c88:      lea     rdi, [rbx + 5]
0x17c8c:      call   0x49b0
0x17c91:      lea     rdi, [rax + 6]
0x17c95:      call   0x1d670
0x17c9a:      lea     rsi, [rbx + 7]
0x17c9e:      mov     dword ptr [rax], 0x5449475b
0x17ca4:      mov     rbp, rax
0x17ca7:      lea     rdi, [rax + 5]
0x17cab:      mov     byte ptr [rax + 4], 0x2d
0x17caf:      call   0x48b0
0x17cb4:      mov     rdi, rbp
0x17cb7:      call   0x172a0
0x17cbc:      mov     rdi, rbp
0x17cbf:      cmp     eax, -1
0x17cc2:      je      0x17ccf
0x17cc4:      add     rsp, 8
0x17cc8:      pop     rbx
0x17cc9:      pop     rbp
0x17cca:      jmp     0x18d10
0x17ccf:      call   0x18d10

```

```

0x17cd4:    mov     rdi, qword ptr [rip + 0x12ba5]
0x17cdb:    mov     rcx, qword ptr [rip + 0x12c06]
0x17ce2:    mov     r8, rbx
0x17ce5:    xor     eax, eax
0x17ce7:    lea     rdx, [rip + 0xa5ba]
0x17cee:    mov     esi, 1
0x17cf3:    call    0x4e60
0x17cf8:    mov     edi, 1
0x17cfd:    call    0x4e20
0x17d02:    nop     word ptr cs:[rax + rax]
0x17d0d:    nop     dword ptr [rax]
0x17d10:    endbr64
0x17d14:    push    rbx
0x17d15:    mov     ecx, 1
0x17d1a:    mov     ebx, esi
0x17d1c:    mov     edx, 1
0x17d21:    sub     rsp, 0x10
0x17d25:    mov     rax, qword ptr fs:[0x28]
0x17d2e:    mov     qword ptr [rsp + 8], rax
0x17d33:    xor     eax, eax
0x17d35:    mov     rsi, rsp
0x17d38:    call    0x17320
0x17d3d:    mov     rdi, qword ptr [rsp]
0x17d41:    mov     eax, ebx
0x17d43:    test    rdi, rdi
0x17d46:    je      0x17d54
0x17d48:    mov     edx, 0xa
0x17d4d:    xor     esi, esi
0x17d4f:    call    0x4ba0
0x17d54:    mov     rdx, qword ptr [rsp + 8]
0x17d59:    sub     rdx, qword ptr fs:[0x28]
0x17d62:    jne     0x17d6a
0x17d64:    add     rsp, 0x10
0x17d68:    pop     rbx
0x17d69:    ret
0x17d6a:    call    0x49d0
0x17d6f:    nop
0x17d70:    endbr64
0x17d74:    push    r15
0x17d76:    push    r14
0x17d78:    mov     r14, rdi
0x17d7b:    push    r13
0x17d7d:    mov     r13d, ecx
0x17d80:    mov     ecx, 1
0x17d85:    push    r12
0x17d87:    mov     r12d, r13d
0x17d8a:    push    rbp
0x17d8b:    mov     ebp, edx

```

```

0x17d8d:    mov     edx, 1
0x17d92:    push    rbx
0x17d93:    mov     rbx, rsi
0x17d96:    sub     rsp, 0x18
0x17d9a:    mov     rax, qword ptr fs:[0x28]
0x17da3:    mov     qword ptr [rsp + 8], rax
0x17da8:    xor     eax, eax
0x17daa:    mov     rsi, rsp
0x17dad:    call    0x17320
0x17db2:    mov     r15, qword ptr [rsp]
0x17db6:    test    r15, r15
0x17db9:    je      0x17de4
0x17dbb:    test    ebp, ebp
0x17dbd:    jle     0x17e3c
0x17dbf:    xor     r12d, r12d
0x17dc2:    jmp     0x17dd5
0x17dc4:    nop     dword ptr [rax]
0x17dc8:    add     r12d, 1
0x17dcc:    add     rbx, 8
0x17dd0:    cmp     ebp, r12d
0x17dd3:    je      0x17e10
0x17dd5:    mov     rsi, qword ptr [rbx]
0x17dd8:    mov     rdi, r15
0x17ddb:    call    0x4b30
0x17de0:    test    eax, eax
0x17de2:    jne     0x17dc8
0x17de4:    mov     rax, qword ptr [rsp + 8]
0x17de9:    sub     rax, qword ptr fs:[0x28]
0x17df2:    jne     0x17e43
0x17df4:    add     rsp, 0x18
0x17df8:    mov     eax, r12d
0x17dfb:    pop     rbx
0x17dfc:    pop     rbp
0x17dfd:    pop     r12
0x17dff:    pop     r13
0x17e01:    pop     r14
0x17e03:    pop     r15
0x17e05:    ret
0x17e06:    nop     word ptr cs:[rax + rax]
0x17e10:    mov     rcx, qword ptr [rip + 0x12ad1]
0x17e17:    mov     r9, r15
0x17e1a:    mov     r8, r14
0x17e1d:    xor     eax, eax
0x17e1f:    mov     rdi, qword ptr [rip + 0x12a5a]
0x17e26:    lea     rdx, [rip + 0xa13c]
0x17e2d:    mov     esi, 1
0x17e32:    mov     r12d, r13d
0x17e35:    call    0x4e60

```

```

0x17e3a:    jmp     0x17de4
0x17e3c:    je      0x17e10
0x17e3e:    xor     r12d, r12d
0x17e41:    jmp     0x17de4
0x17e43:    call    0x49d0
0x17e48:    nop     dword ptr [rax + rax]
0x17e50:    endbr64
0x17e54:    push    r12
0x17e56:    mov     ecx, 1
0x17e5b:    mov     edx, 1
0x17e60:    mov     r12, rdi
0x17e63:    push    rbp
0x17e64:    push    rbx
0x17e65:    mov     ebx, esi
0x17e67:    sub     rsp, 0x10
0x17e6b:    mov     rax, qword ptr fs:[0x28]
0x17e74:    mov     qword ptr [rsp + 8], rax
0x17e79:    xor     eax, eax
0x17e7b:    mov     rsi, rsp
0x17e7e:    call    0x17320
0x17e83:    mov     rbp, qword ptr [rsp]
0x17e87:    mov     eax, ebx
0x17e89:    test    rbp, rbp
0x17e8c:    je      0x17ea7
0x17e8e:    cmp     byte ptr [rbp], 0x4f
0x17e92:    je      0x17ec0
0x17e94:    lea     rsi, [rip + 0x8e4b]
0x17e9b:    mov     rdi, rbp
0x17e9e:    call    0x4b30
0x17ea3:    test    eax, eax
0x17ea5:    jne     0x17ed8
0x17ea7:    mov     rdx, qword ptr [rsp + 8]
0x17eac:    sub     rdx, qword ptr fs:[0x28]
0x17eb5:    jne     0x17f03
0x17eb7:    add     rsp, 0x10
0x17ebb:    pop     rbx
0x17ebc:    pop     rbp
0x17ebd:    pop     r12
0x17ebf:    ret
0x17ec0:    cmp     byte ptr [rbp + 1], 0x4e
0x17ec4:    jne     0x17e94
0x17ec6:    cmp     byte ptr [rbp + 2], 0
0x17eca:    mov     eax, 1
0x17ecf:    je      0x17ea7
0x17ed1:    jmp     0x17e94
0x17ed3:    nop     dword ptr [rax + rax]
0x17ed8:    mov     rcx, qword ptr [rip + 0x12a09]
0x17edf:    mov     r9, rbp

```

```

0x17ee2:    mov     r8, r12
0x17ee5:    xor     eax, eax
0x17ee7:    mov     rdi, qword ptr [rip + 0x12992]
0x17eee:    lea     rdx, [rip + 0xa074]
0x17ef5:    mov     esi, 1
0x17efa:    call    0x4e60
0x17eff:    mov     eax, ebx
0x17f01:    jmp     0x17ea7
0x17f03:    call    0x49d0
0x17f08:    nop     dword ptr [rax + rax]
0x17f10:    endbr64
0x17f14:    push    r12
0x17f16:    mov     ecx, 1
0x17f1b:    mov     r12, rsi
0x17f1e:    mov     edx, 1
0x17f23:    sub     rsp, 0x10
0x17f27:    mov     rax, qword ptr fs:[0x28]
0x17f30:    mov     qword ptr [rsp + 8], rax
0x17f35:    xor     eax, eax
0x17f37:    mov     rsi, rsp
0x17f3a:    call    0x17320
0x17f3f:    mov     rdi, qword ptr [rsp]
0x17f43:    test    rdi, rdi
0x17f46:    je      0x17f50
0x17f48:    call    0x1d7a0
0x17f4d:    mov     r12, rax
0x17f50:    mov     rax, qword ptr [rsp + 8]
0x17f55:    sub     rax, qword ptr fs:[0x28]
0x17f5e:    jne     0x17f6a
0x17f60:    add     rsp, 0x10
0x17f64:    mov     rax, r12
0x17f67:    pop     r12
0x17f69:    ret
0x17f6a:    call    0x49d0
0x17f6f:    nop
0x17f70:    endbr64
0x17f74:    push    r14
0x17f76:    push    r13
0x17f78:    push    r12
0x17f7a:    push    rbp
0x17f7b:    push    rbx
0x17f7c:    sub     rsp, 0x10
0x17f80:    mov     rax, qword ptr fs:[0x28]
0x17f89:    mov     qword ptr [rsp + 8], rax
0x17f8e:    xor     eax, eax
0x17f90:    test    edx, edx
0x17f92:    jle     0x17fe8
0x17f94:    lea     eax, [rdx - 1]

```

```

0x17f97:      mov     rbx, rsi
0x17f9a:      mov     rbp, rdi
0x17f9d:      mov     r12, rsp
0x17fa0:      lea     r13, [rsi + rax*8 + 8]
0x17fa5:      lea     r14, [rip + 0x9fbd]
0x17fac:      nop     dword ptr [rax]
0x17fb0:      mov     rdi, qword ptr [rbx]
0x17fb3:      mov     ecx, 1
0x17fb8:      mov     edx, 1
0x17fbd:      mov     rsi, r12
0x17fc0:      call    0x17320
0x17fc5:      mov     rdi, qword ptr [rsp]
0x17fc9:      test    rdi, rdi
0x17fcc:      je      0x17fdb
0x17fce:      call    0xbd00
0x17fd3:      cmp     eax, -1
0x17fd6:      je      0x18008
0x17fd8:      mov     dword ptr [rbp], eax
0x17fdb:      add     rbx, 8
0x17fdf:      add     rbp, 4
0x17fe3:      cmp     rbx, r13
0x17fe6:      jne     0x17fb0
0x17fe8:      mov     rax, qword ptr [rsp + 8]
0x17fed:      sub     rax, qword ptr fs:[0x28]
0x17ff6:      jne     0x1802e
0x17ff8:      add     rsp, 0x10
0x17ffc:      pop     rbx
0x17ffd:      pop     rbp
0x17ffe:      pop     r12
0x18000:      pop     r13
0x18002:      pop     r14
0x18004:      ret
0x18005:      nop     dword ptr [rax]
0x18008:      mov     r9, qword ptr [rsp]
0x1800c:      mov     r8, qword ptr [rbx]
0x1800f:      mov     rdx, r14
0x18012:      mov     esi, 1
0x18017:      mov     rcx, qword ptr [rip + 0x128ca]
0x1801e:      mov     rdi, qword ptr [rip + 0x1285b]
0x18025:      xor     eax, eax
0x18027:      call    0x4e60
0x1802c:      jmp     0x17fdb
0x1802e:      call    0x49d0
0x18033:      nop     word ptr cs:[rax + rax]
0x1803e:      nop
0x18040:      endbr64
0x18044:      push    r12
0x18046:      cmp     byte ptr [rdi], 0x2f

```

```

0x18049:      mov     r12, rdi
0x1804c:      lea     rax, [rdi + 1]
0x18050:      je      0x18061
0x18052:      lea     rdi, [rip + 0x9f26]
0x18059:      call    0x8b00
0x1805e:      mov     rax, r12
0x18061:      movzx   edx, byte ptr [rax]
0x18064:      mov     rsi, rax
0x18067:      test    dl, dl
0x18069:      jne     0x1808c
0x1806b:      jmp     0x180f0
0x18070:      cmp     dl, 0x2e
0x18073:      jne     0x18118
0x18079:      movzx   ecx, byte ptr [rax + 1]
0x1807d:      cmp     cl, 0x2f
0x18080:      je      0x18086
0x18082:      test    cl, cl
0x18084:      jne     0x180a0
0x18086:      add     rax, 1
0x1808a:      mov     edx, ecx
0x1808c:      cmp     dl, 0x2f
0x1808f:      jne     0x18070
0x18091:      movzx   ecx, byte ptr [rax + 1]
0x18095:      jmp     0x18086
0x18097:      nop     word ptr [rax + rax]
0x180a0:      cmp     cl, 0x2e
0x180a3:      jne     0x180c0
0x180a5:      movzx   edx, byte ptr [rax + 2]
0x180a9:      cmp     dl, 0x2f
0x180ac:      je      0x18120
0x180ae:      test    dl, dl
0x180b0:      je      0x18120
0x180b2:      mov     edx, 0x2e
0x180b7:      nop     word ptr [rax + rax]
0x180c0:      mov     rdi, rax
0x180c3:      mov     rcx, rsi
0x180c6:      add     rax, 1
0x180ca:      add     rsi, 1
0x180ce:      mov     byte ptr [rsi - 1], dl
0x180d1:      movzx   edx, byte ptr [rax]
0x180d4:      test    dl, dl
0x180d6:      je      0x180f0
0x180d8:      cmp     dl, 0x2f
0x180db:      jne     0x180c0
0x180dd:      test    dl, dl
0x180df:      je      0x180f0
0x180e1:      mov     byte ptr [rsi], 0x2f
0x180e4:      movzx   edx, byte ptr [rdi + 1]

```

```

0x180e8:    lea     rsi, [rcx + 2]
0x180ec:    test    dl, dl
0x180ee:    jne     0x1808c
0x180f0:    cmp     byte ptr [rsi - 1], 0x2f
0x180f4:    jne     0x18109
0x180f6:    mov     rax, rsi
0x180f9:    sub     rax, r12
0x180fc:    cmp     rax, 1
0x18100:    setg    al
0x18103:    movzx   eax, al
0x18106:    sub     rsi, rax
0x18109:    mov     rax, r12
0x1810c:    mov     byte ptr [rsi], 0
0x1810f:    pop     r12
0x18111:    ret
0x18112:    nop     word ptr [rax + rax]
0x18118:    test    dl, dl
0x1811a:    jne     0x180c0
0x1811c:    jmp     0x180f0
0x1811e:    nop
0x18120:    lea     rdi, [rsi - 1]
0x18124:    cmp     rdi, r12
0x18127:    jne     0x1812f
0x18129:    add     rax, 2
0x1812d:    jmp     0x180ec
0x1812f:    cmp     byte ptr [rsi - 2], 0x2f
0x18133:    lea     rcx, [rsi - 2]
0x18137:    je      0x1814e
0x18139:    nop     dword ptr [rax]
0x18140:    mov     rsi, rcx
0x18143:    sub     rcx, 1
0x18147:    cmp     byte ptr [rcx], 0x2f
0x1814a:    jne     0x18140
0x1814c:    jmp     0x18129
0x1814e:    mov     rsi, rdi
0x18151:    jmp     0x18129
0x18153:    nop     word ptr cs:[rax + rax]
0x1815e:    nop
0x18160:    endbr64
0x18164:    push    rbx
0x18165:    call    0x4a60
0x1816a:    mov     edi, eax
0x1816c:    mov     ebx, eax
0x1816e:    call    0x4970
0x18173:    test    rax, rax
0x18176:    je      0x18189
0x18178:    mov     rdi, qword ptr [rax]
0x1817b:    call    0x1d7a0

```



```

0x18180:    pop     rbx
0x18181:    mov     qword ptr [rip + 0x13910], rax
0x18188:    ret
0x18189:    mov     rcx, qword ptr [rip + 0x12758]
0x18190:    mov     r8d, ebx
0x18193:    mov     rdi, qword ptr [rip + 0x126e6]
0x1819a:    lea     rdx, [rip + 0xa13f]
0x181a1:    mov     esi, 1
0x181a6:    call    0x4e60
0x181ab:    mov     rcx, qword ptr [rip + 0x12736]
0x181b2:    xor     eax, eax
0x181b4:    mov     rdi, qword ptr [rip + 0x126c5]
0x181bb:    lea     r8, [rip + 0x9dd7]
0x181c2:    lea     rdx, [rip + 0xa14f]
0x181c9:    mov     esi, 1
0x181ce:    call    0x4e60
0x181d3:    mov     rdi, qword ptr [rip + 0x126a6]
0x181da:    mov     esi, 1
0x181df:    xor     eax, eax
0x181e1:    mov     rcx, qword ptr [rip + 0x12700]
0x181e8:    lea     rdx, [rip + 0xa161]
0x181ef:    call    0x4e60
0x181f4:    mov     edi, 1
0x181f9:    call    0x4e20
0x181fe:    nop
0x18200:    endbr64
0x18204:    push    r12
0x18206:    push    rbp
0x18207:    mov     rbp, rsi
0x1820a:    push    rbx
0x1820b:    movsxd  rbx, edx
0x1820e:    cmp     ebx, 2
0x18211:    je      0x18250
0x18213:    mov     r12, rdi
0x18216:    jg      0x18230
0x18218:    test    ebx, ebx
0x1821a:    je      0x18224
0x1821c:    cmp     ebx, 1
0x1821f:    jne     0x18260
0x18221:    mov     byte ptr [rsi], 0x20
0x18224:    pop     rbx
0x18225:    pop     rbp
0x18226:    pop     r12
0x18228:    ret
0x18229:    nop     dword ptr [rax]
0x18230:    cmp     ebx, 3
0x18233:    jne     0x18260
0x18235:    mov     edx, 0x2020

```

```

0x1823a:    mov     byte ptr [rsi + 2], 0x20
0x1823e:    mov     word ptr [rsi], dx
0x18241:    pop     rbx
0x18242:    pop     rbp
0x18243:    pop     r12
0x18245:    ret
0x18246:    nop     word ptr cs:[rax + rax]
0x18250:    mov     ecx, 0x2020
0x18255:    mov     word ptr [rsi], cx
0x18258:    pop     rbx
0x18259:    pop     rbp
0x1825a:    pop     r12
0x1825c:    ret
0x1825d:    nop     dword ptr [rax]
0x18260:    mov     rdi, r12
0x18263:    call    0x49b0
0x18268:    movsxd  rdx, eax
0x1826b:    cmp     ebx, eax
0x1826d:    jge     0x182a0
0x1826f:    mov     eax, 0x2e2e
0x18274:    lea     r8d, [rbx - 3]
0x18278:    mov     byte ptr [rbp + 2], 0x2e
0x1827c:    sub     rdx, rbx
0x1827f:    mov     word ptr [rbp], ax
0x18283:    movsxd  r8, r8d
0x18286:    lea     rsi, [r12 + rdx + 3]
0x1828b:    pop     rbx
0x1828c:    lea     rdi, [rbp + 3]
0x18290:    mov     rdx, r8
0x18293:    pop     rbp
0x18294:    pop     r12
0x18296:    jmp     0x4bb0
0x1829b:    nop     dword ptr [rax + rax]
0x182a0:    pop     rbx
0x182a1:    mov     rsi, r12
0x182a4:    mov     rdi, rbp
0x182a7:    pop     rbp
0x182a8:    pop     r12
0x182aa:    jmp     0x4bb0
0x182af:    nop
0x182b0:    endbr64
0x182b4:    push    r12
0x182b6:    mov     r12, rsi
0x182b9:    push    rbp
0x182ba:    mov     rbp, rdi
0x182bd:    push    rbx
0x182be:    lea     ebx, [rdx - 1]
0x182c1:    mov     edx, ebx

```

```

0x182c3:      call    0x18200
0x182c8:      mov     rdi, rbp
0x182cb:      call    0x49b0
0x182d0:      movsxd  rcx, ebx
0x182d3:      cmp     ebx, eax
0x182d5:      movsxd  rdx, eax
0x182d8:      mov     rax, rcx
0x182db:      pop     rbx
0x182dc:      cmovg   rax, rdx
0x182e0:      pop     rbp
0x182e1:      mov     byte ptr [r12 + rax], 0
0x182e6:      mov     rax, r12
0x182e9:      pop     r12
0x182eb:      ret
0x182ec:      nop     dword ptr [rax]
0x182f0:      endbr64
0x182f4:      push    r13
0x182f6:      mov     edx, 1
0x182fb:      xor     esi, esi
0x182fd:      push    r12
0x182ff:      push    rbp
0x18300:      mov     ebp, edi
0x18302:      call    0x4e50
0x18307:      mov     edi, ebp
0x18309:      mov     edx, 2
0x1830e:      xor     esi, esi
0x18310:      mov     r13, rax
0x18313:      call    0x4e50
0x18318:      mov     rsi, r13
0x1831b:      mov     edi, ebp
0x1831d:      xor     edx, edx
0x1831f:      mov     r12, rax
0x18322:      call    0x4e50
0x18327:      mov     rax, r12
0x1832a:      pop     rbp
0x1832b:      pop     r12
0x1832d:      pop     r13
0x1832f:      ret
0x18330:      endbr64
0x18334:      sub     rsp, 0x18
0x18338:      xor     edi, edi
0x1833a:      mov     rax, qword ptr fs:[0x28]
0x18343:      mov     qword ptr [rsp + 8], rax
0x18348:      xor     eax, eax
0x1834a:      call    0x4be0
0x1834f:      mov     rdi, rsp
0x18352:      mov     qword ptr [rsp], rax
0x18356:      call    0x4840

```

```

0x1835b:    mov     rdx, qword ptr [rsp + 8]
0x18360:    sub     rdx, qword ptr fs:[0x28]
0x18369:    jne     0x18370
0x1836b:    add     rsp, 0x18
0x1836f:    ret
0x18370:    call    0x49d0
0x18375:    nop     word ptr cs:[rax + rax]
0x18380:    endbr64
0x18384:    push    r12
0x18386:    push    rbp
0x18387:    mov     rbp, rsi
0x1838a:    push    rbx
0x1838b:    mov     rbx, rdi
0x1838e:    call    0x49b0
0x18393:    mov     rdi, rbp
0x18396:    mov     r12, rax
0x18399:    call    0x49b0
0x1839e:    lea     rdi, [r12 + rax + 2]
0x183a3:    call    0x1d670
0x183a8:    mov     r9, rbp
0x183ab:    mov     r8, rbx
0x183ae:    mov     esi, 1
0x183b3:    mov     r12, rax
0x183b6:    mov     rdi, rax
0x183b9:    xor     eax, eax
0x183bb:    mov     rdx, -1
0x183c2:    lea     rcx, [rip + 0x9be5]
0x183c9:    call    0x4f10
0x183ce:    mov     rdi, r12
0x183d1:    call    0x4b20
0x183d6:    mov     r12d, eax
0x183d9:    cmp     eax, -1
0x183dc:    je      0x183f0
0x183de:    mov     eax, r12d
0x183e1:    pop     rbx
0x183e2:    pop     rbp
0x183e3:    pop     r12
0x183e5:    ret
0x183e6:    nop     word ptr cs:[rax + rax]
0x183f0:    mov     rcx, qword ptr [rip + 0x124f1]
0x183f7:    mov     rdi, qword ptr [rip + 0x12482]
0x183fe:    mov     r8, rbx
0x18401:    xor     eax, eax
0x18403:    lea     rdx, [rip + 0x9f6e]
0x1840a:    mov     esi, 1
0x1840f:    call    0x4e60
0x18414:    mov     eax, r12d
0x18417:    pop     rbx

```

```

0x18418:    pop     rbp
0x18419:    pop     r12
0x1841b:    ret
0x1841c:    nop     dword ptr [rax]
0x18420:    endbr64
0x18424:    push    r13
0x18426:    push    r12
0x18428:    lea     r12, [rip + 0x9b85]
0x1842f:    push    rbp
0x18430:    mov     rdi, r12
0x18433:    push    rbx
0x18434:    sub     rsp, 8
0x18438:    call    0x47e0
0x1843d:    mov     r13, qword ptr [rip + 0x1365c]
0x18444:    test    rax, rax
0x18447:    je      0x184b0
0x18449:    mov     rdi, rax
0x1844c:    mov     rbx, rax
0x1844f:    call    0x49b0
0x18454:    mov     rdi, r13
0x18457:    mov     rbp, rax
0x1845a:    call    0x49b0
0x1845f:    lea     rdi, [rbp + rax + 2]
0x18464:    call    0x1d670
0x18469:    mov     r9, qword ptr [rip + 0x13630]
0x18470:    mov     r8, rbx
0x18473:    lea     rcx, [rip + 0x9b3f]
0x1847a:    mov     rbp, rax
0x1847d:    mov     rdi, rax
0x18480:    mov     esi, 1
0x18485:    xor     eax, eax
0x18487:    mov     rdx, -1
0x1848e:    call    0x4f10
0x18493:    mov     rdi, r12
0x18496:    mov     rsi, rbp
0x18499:    call    0x18380
0x1849e:    add     rsp, 8
0x184a2:    mov     rdi, rbp
0x184a5:    pop     rbx
0x184a6:    pop     rbp
0x184a7:    pop     r12
0x184a9:    pop     r13
0x184ab:    jmp     0x18d10
0x184b0:    add     rsp, 8
0x184b4:    mov     rsi, r13
0x184b7:    mov     rdi, r12
0x184ba:    pop     rbx
0x184bb:    pop     rbp

```

```

0x184bc:    pop     r12
0x184be:    pop     r13
0x184c0:    jmp     0x18380
0x184c5:    nop     word ptr cs:[rax + rax]
0x184d0:    endbr64
0x184d4:    test    rsi, rsi
0x184d7:    je      0x18520
0x184d9:    push    rbp
0x184da:    mov     rbp, rsi
0x184dd:    push    rbx
0x184de:    mov     rbx, rdi
0x184e1:    sub     rsp, 8
0x184e5:    call    0x4f00
0x184ea:    mov     rdi, rbx
0x184ed:    lea     rsi, [rbx + rbp]
0x184f1:    nop     dword ptr [rax]
0x184f8:    movsx   rcx, byte ptr [rdi]
0x184fc:    mov     rdx, qword ptr [rax]
0x184ff:    test    byte ptr [rdx + rcx*2 + 1], 0x40
0x18504:    jne     0x18509
0x18506:    mov     byte ptr [rdi], 0x3f
0x18509:    add     rdi, 1
0x1850d:    cmp     rsi, rdi
0x18510:    jne     0x184f8
0x18512:    add     rsp, 8
0x18516:    pop     rbx
0x18517:    pop     rbp
0x18518:    ret
0x18519:    nop     dword ptr [rax]
0x18520:    ret
0x18521:    nop     word ptr cs:[rax + rax]
0x1852c:    nop     dword ptr [rax]
0x18530:    endbr64
0x18534:    test    rsi, rsi
0x18537:    je      0x185c2
0x1853d:    push    rbp
0x1853e:    mov     rbp, rsi
0x18541:    push    rbx
0x18542:    mov     rbx, rdi
0x18545:    sub     rsp, 8
0x18549:    call    0x4f00
0x1854e:    mov     rdi, rbx
0x18551:    lea     rsi, [rbx + rbp]
0x18555:    movabs  r9, 0x68000000c0000001
0x1855f:    movabs  r8, 0x2400f82000000000
0x18569:    mov     rcx, qword ptr [rax]
0x1856c:    nop     dword ptr [rax]
0x18570:    movsx   rdx, byte ptr [rdi]

```

```

0x18574:      mov     rax, rdx
0x18577:      test    byte ptr [rcx + rdx*2], 8
0x1857b:      jne     0x185b0
0x1857d:      cmp     dl, 0x3d
0x18580:      jg      0x185a0
0x18582:      cmp     dl, 0x24
0x18585:      jle     0x1858d
0x18587:      bt      r8, rdx
0x1858b:      jb      0x185b0
0x1858d:      add     rsp, 8
0x18591:      mov     eax, 1
0x18596:      pop     rbx
0x18597:      pop     rbp
0x18598:      ret
0x18599:      nop     dword ptr [rax]
0x185a0:      sub     eax, 0x40
0x185a3:      cmp     al, 0x3e
0x185a5:      ja      0x1858d
0x185a7:      bt      r9, rax
0x185ab:      jae     0x1858d
0x185ad:      nop     dword ptr [rax]
0x185b0:      add     rdi, 1
0x185b4:      cmp     rsi, rdi
0x185b7:      jne     0x18570
0x185b9:      add     rsp, 8
0x185bd:      xor     eax, eax
0x185bf:      pop     rbx
0x185c0:      pop     rbp
0x185c1:      ret
0x185c2:      xor     eax, eax
0x185c4:      ret
0x185c5:      nop     word ptr cs:[rax + rax]
0x185d0:      endbr64
0x185d4:      push    rbx
0x185d5:      mov     rbx, rdi
0x185d8:      call    0x49b0
0x185dd:      sub     eax, 1
0x185e0:      js      0x18618
0x185e2:      cdqe
0x185e4:      jmp     0x18602
0x185e6:      nop     word ptr cs:[rax + rax]
0x185f0:      cmp     dl, 0x20
0x185f3:      je      0x185fa
0x185f5:      cmp     dl, 9
0x185f8:      jne     0x18618
0x185fa:      sub     rax, 1
0x185fe:      test    eax, eax
0x18600:      js      0x18618

```

```

0x18602:    movzx    edx, byte ptr [rbx + rax]
0x18606:    cmp      dl, 0x26
0x18609:    jne      0x185f0
0x1860b:    mov      eax, 1
0x18610:    pop      rbx
0x18611:    ret
0x18612:    nop      word ptr [rax + rax]
0x18618:    xor      eax, eax
0x1861a:    pop      rbx
0x1861b:    ret
0x1861c:    nop      dword ptr [rax]
0x18620:    endbr64
0x18624:    jmp      0x1863c
0x18626:    nop      word ptr cs:[rax + rax]
0x18630:    cmp      al, 0x20
0x18632:    je       0x18638
0x18634:    cmp      al, 0x3b
0x18636:    jne      0x18650
0x18638:    add      rdi, 1
0x1863c:    movzx    eax, byte ptr [rdi]
0x1863f:    test     al, al
0x18641:    jne      0x18630
0x18643:    mov      eax, 1
0x18648:    ret
0x18649:    nop      dword ptr [rax]
0x18650:    xor      eax, eax
0x18652:    ret
0x18653:    nop      word ptr cs:[rax + rax]
0x1865e:    nop
0x18660:    endbr64
0x18664:    push     r15
0x18666:    push     r14
0x18668:    push     r13
0x1866a:    push     r12
0x1866c:    push     rbp
0x1866d:    push     rbx
0x1866e:    sub      rsp, 0x98
0x18675:    mov      rax, qword ptr fs:[0x28]
0x1867e:    mov      qword ptr [rsp + 0x88], rax
0x18686:    xor      eax, eax
0x18688:    lea      r15, [rsp + 0x10]
0x1868d:    lea      r13, [rsp + 0x30]
0x18692:    mov      qword ptr [rsp + 8], 0
0x1869b:    mov      qword ptr [rsp], 0
0x186a3:    jmp      0x186d3
0x186a5:    nop      dword ptr [rax]
0x186a8:    mov      rax, qword ptr [rsp]
0x186ac:    mov      qword ptr [rax + 0x18], r14

```



```

0x186b0:    mov     rdi, r13
0x186b3:    call   0x1d7a0
0x186b8:    mov     dword ptr [r14 + 8], r12d
0x186bc:    mov     qword ptr [r14], rax
0x186bf:    mov     dword ptr [r14 + 0xc], ebp
0x186c3:    mov     dword ptr [r14 + 0x10], ebx
0x186c7:    mov     qword ptr [r14 + 0x18], 0
0x186cf:    mov     qword ptr [rsp], r14
0x186d3:    xor     ecx, ecx
0x186d5:    mov     edx, 3
0x186da:    mov     rsi, r15
0x186dd:    mov     rdi, r13
0x186e0:    call   0x17320
0x186e5:    cmp     byte ptr [rsp + 0x30], 0
0x186ea:    je      0x18758
0x186ec:    mov     rdi, qword ptr [rsp + 0x10]
0x186f1:    mov     r12d, 0xffffffff
0x186f7:    test    rdi, rdi
0x186fa:    je      0x18704
0x186fc:    call   0xbd00
0x18701:    mov     r12d, eax
0x18704:    mov     rdi, qword ptr [rsp + 0x18]
0x18709:    mov     ebp, 0xffffffff
0x1870e:    test    rdi, rdi
0x18711:    je      0x1871a
0x18713:    call   0xbd00
0x18718:    mov     ebp, eax
0x1871a:    mov     rdi, qword ptr [rsp + 0x20]
0x1871f:    mov     ebx, 0xffffffff
0x18724:    test    rdi, rdi
0x18727:    je      0x18730
0x18729:    call   0xbd00
0x1872e:    mov     ebx, eax
0x18730:    mov     edi, 0x20
0x18735:    call   0x1d670
0x1873a:    cmp     qword ptr [rsp + 8], 0
0x18740:    mov     r14, rax
0x18743:    jne     0x186a8
0x18749:    mov     qword ptr [rsp + 8], rax
0x1874e:    jmp     0x186b0
0x18753:    nop     dword ptr [rax + rax]
0x18758:    mov     rdx, qword ptr [rsp + 8]
0x1875d:    test    rdx, rdx
0x18760:    je      0x1877d
0x18762:    mov     rax, qword ptr [rip + 0x13327]
0x18769:    test    rax, rax
0x1876c:    je      0x187a8
0x1876e:    mov     rcx, qword ptr [rsp]

```

```

0x18772:    mov     qword ptr [rip + 0x13317], rdx
0x18779:    mov     qword ptr [rcx + 0x18], rax
0x1877d:    mov     rax, qword ptr [rsp + 0x88]
0x18785:    sub     rax, qword ptr fs:[0x28]
0x1878e:    jne     0x187b6
0x18790:    add     rsp, 0x98
0x18797:    pop     rbx
0x18798:    pop     rbp
0x18799:    pop     r12
0x1879b:    pop     r13
0x1879d:    pop     r14
0x1879f:    pop     r15
0x187a1:    ret
0x187a2:    nop     word ptr [rax + rax]
0x187a8:    mov     rax, qword ptr [rsp + 8]
0x187ad:    mov     qword ptr [rip + 0x132dc], rax
0x187b4:    jmp     0x1877d
0x187b6:    call    0x49d0
0x187bb:    nop     dword ptr [rax + rax]
0x187c0:    endbr64
0x187c4:    sub     rsp, 8
0x187c8:    lea     rsi, [rip - 0xf]
0x187cf:    call    0x4b50
0x187d4:    mov     dword ptr [rip + 0x132d2], 1
0x187de:    add     rsp, 8
0x187e2:    ret
0x187e3:    nop     word ptr cs:[rax + rax]
0x187ee:    nop
0x187f0:    cmp     edi, 0x14
0x187f3:    je      0x18850
0x187f5:    jg      0x18818
0x187f7:    cmp     edi, 0xe
0x187fa:    je      0x18838
0x187fc:    cmp     edi, 0x12
0x187ff:    jne     0x18861
0x18801:    lea     rsi, [rip + 0x148]
0x18808:    mov     edi, 0x12
0x1880d:    jmp     0x4b50
0x18812:    nop     word ptr [rax + rax]
0x18818:    cmp     edi, 0x1c
0x1881b:    jne     0x18830
0x1881d:    lea     rsi, [rip + 0x1cc]
0x18824:    mov     edi, 0x1c
0x18829:    jmp     0x4b50
0x1882e:    nop
0x18830:    ret
0x18831:    nop     dword ptr [rax]
0x18838:    lea     rsi, [rip + 0x31]

```

```

0x1883f:      mov     edi, 0xe
0x18844:      jmp     0x4b50
0x18849:      nop     dword ptr [rax]
0x18850:      lea     rsi, [rip + 0x89]
0x18857:      mov     edi, 0x14
0x1885c:      jmp     0x4b50
0x18861:      ret
0x18862:      nop     word ptr cs:[rax + rax]
0x1886d:      nop     dword ptr [rax]
0x18870:      endbr64
0x18874:      mov     edx, dword ptr [rip + 0x1323a]
0x1887a:      push    rbp
0x1887b:      mov     eax, 1
0x18880:      mov     ebp, edi
0x18882:      test    edx, edx
0x18884:      jne     0x188b0
0x18886:      mov     edi, ebp
0x18888:      mov     dword ptr [rip + 0x1322e], eax
0x1888e:      call    0x187f0
0x18893:      xor     eax, eax
0x18895:      call    0x18330
0x1889a:      mov     edi, 0x3c
0x1889f:      pop     rbp
0x188a0:      sub     edi, dword ptr [rax]
0x188a2:      jmp     0x4aa0
0x188a7:      nop     word ptr [rax + rax]
0x188b0:      xor     eax, eax
0x188b2:      call    0x18330
0x188b7:      mov     eax, dword ptr [rax]
0x188b9:      test    eax, eax
0x188bb:      je      0x188d0
0x188bd:      xor     eax, eax
0x188bf:      call    0xb4e0
0x188c4:      xor     eax, eax
0x188c6:      jmp     0x18886
0x188c8:      nop     dword ptr [rax + rax]
0x188d0:      mov     edi, ebp
0x188d2:      call    0x16fd0
0x188d7:      jmp     0x188bd
0x188d9:      nop     dword ptr [rax]
0x188e0:      endbr64
0x188e4:      mov     eax, dword ptr [rip + 0x131ca]
0x188ea:      push    rbp
0x188eb:      mov     ebp, edi
0x188ed:      test    eax, eax
0x188ef:      jne     0x18908
0x188f1:      mov     eax, 1
0x188f6:      mov     edi, ebp

```

```

0x188f8:      pop     rbp
0x188f9:      mov     dword ptr [rip + 0x131c5], eax
0x188ff:      jmp     0x187f0
0x18904:      nop     dword ptr [rax]
0x18908:      xor     eax, eax
0x1890a:      call    0x9c90
0x1890f:      cmp     eax, 1
0x18912:      sete    al
0x18915:      movzx   eax, al
0x18918:      mov     dword ptr [rip + 0x1319a], eax
0x1891e:      xor     eax, eax
0x18920:      call    0x9610
0x18925:      call    0x4920
0x1892a:      mov     esi, 0x13
0x1892f:      mov     edi, eax
0x18931:      call    0x4bd0
0x18936:      xor     eax, eax
0x18938:      mov     edi, ebp
0x1893a:      pop     rbp
0x1893b:      mov     dword ptr [rip + 0x13183], eax
0x18941:      jmp     0x187f0
0x18946:      nop     word ptr cs:[rax + rax]
0x18950:      endbr64
0x18954:      mov     eax, dword ptr [rip + 0x1315e]
0x1895a:      push    rbx
0x1895b:      mov     ebx, edi
0x1895d:      test    eax, eax
0x1895f:      jne     0x189a0
0x18961:      cmp     ebx, 0x14
0x18964:      je      0x189b4
0x18966:      jg      0x18988
0x18968:      cmp     ebx, 0xe
0x1896b:      je      0x189d8
0x1896d:      cmp     ebx, 0x12
0x18970:      jne     0x189d0
0x18972:      lea     rsi, [rip - 0x29]
0x18979:      mov     edi, 0x12
0x1897e:      pop     rbx
0x1897f:      jmp     0x4b50
0x18984:      nop     dword ptr [rax]
0x18988:      cmp     ebx, 0x1c
0x1898b:      jne     0x189d0
0x1898d:      lea     rsi, [rip + 0x5c]
0x18994:      mov     edi, 0x1c
0x18999:      pop     rbx
0x1899a:      jmp     0x4b50
0x1899f:      nop
0x189a0:      call    0x7bd0

```

```

0x189a5:    mov     dword ptr [rip + 0x13111], 0
0x189af:    cmp     ebx, 0x14
0x189b2:    jne     0x18966
0x189b4:    lea     rsi, [rip - 0xdb]
0x189bb:    mov     edi, 0x14
0x189c0:    pop     rbx
0x189c1:    jmp     0x4b50
0x189c6:    nop     word ptr cs:[rax + rax]
0x189d0:    pop     rbx
0x189d1:    ret
0x189d2:    nop     word ptr [rax + rax]
0x189d8:    lea     rsi, [rip - 0x16f]
0x189df:    mov     edi, 0xe
0x189e4:    pop     rbx
0x189e5:    jmp     0x4b50
0x189ea:    nop     word ptr [rax + rax]
0x189f0:    endbr64
0x189f4:    mov     eax, dword ptr [rip + 0x130ba]
0x189fa:    push    rbx
0x189fb:    mov     ebx, edi
0x189fd:    test    eax, eax
0x189ff:    jne     0x18a50
0x18a01:    mov     eax, 1
0x18a06:    mov     dword ptr [rip + 0x130b4], eax
0x18a0c:    cmp     ebx, 0x14
0x18a0f:    je      0x18a72
0x18a11:    jg      0x18a30
0x18a13:    cmp     ebx, 0xe
0x18a16:    je      0x18a90
0x18a18:    cmp     ebx, 0x12
0x18a1b:    jne     0x18a88
0x18a1d:    lea     rsi, [rip - 0xd4]
0x18a24:    mov     edi, 0x12
0x18a29:    pop     rbx
0x18a2a:    jmp     0x4b50
0x18a2f:    nop
0x18a30:    cmp     ebx, 0x1c
0x18a33:    jne     0x18a88
0x18a35:    lea     rsi, [rip - 0x4c]
0x18a3c:    mov     edi, 0x1c
0x18a41:    pop     rbx
0x18a42:    jmp     0x4b50
0x18a47:    nop     word ptr [rax + rax]
0x18a50:    xor     eax, eax
0x18a52:    call    0xa6c0
0x18a57:    xor     eax, eax
0x18a59:    call    0x9d80
0x18a5e:    mov     edi, ebx

```

```

0x18a60:    call    0x7bd0
0x18a65:    xor     eax, eax
0x18a67:    mov     dword ptr [rip + 0x13053], eax
0x18a6d:    cmp     ebx, 0x14
0x18a70:    jne     0x18a11
0x18a72:    lea     rsi, [rip - 0x199]
0x18a79:    mov     edi, 0x14
0x18a7e:    pop     rbx
0x18a7f:    jmp     0x4b50
0x18a84:    nop     dword ptr [rax]
0x18a88:    pop     rbx
0x18a89:    ret
0x18a8a:    nop     word ptr [rax + rax]
0x18a90:    lea     rsi, [rip - 0x227]
0x18a97:    mov     edi, 0xe
0x18a9c:    pop     rbx
0x18a9d:    jmp     0x4b50
0x18aa2:    nop     word ptr cs:[rax + rax]
0x18aad:    nop     dword ptr [rax]
0x18ab0:    endbr64
0x18ab4:    sub     rsp, 8
0x18ab8:    mov     ecx, dword ptr [rip + 0x13006]
0x18abe:    test    ecx, ecx
0x18ac0:    jne     0x18af0
0x18ac2:    mov     edx, dword ptr [rip + 0x12ff8]
0x18ac8:    test    edx, edx
0x18aca:    jne     0x18b30
0x18acc:    mov     eax, dword ptr [rip + 0x12fea]
0x18ad2:    test    eax, eax
0x18ad4:    jne     0x18ae0
0x18ad6:    add     rsp, 8
0x18ada:    ret
0x18adb:    nop     dword ptr [rax + rax]
0x18ae0:    mov     dword ptr [rip + 0x12fd2], 0
0x18aea:    add     rsp, 8
0x18aee:    ret
0x18aef:    nop
0x18af0:    xor     eax, eax
0x18af2:    call    0x9610
0x18af7:    call    0x4920
0x18afc:    mov     esi, 0x13
0x18b01:    mov     edi, eax
0x18b03:    call    0x4bd0
0x18b08:    mov     dword ptr [rip + 0x12fb2], 0
0x18b12:    mov     dword ptr [rip + 0x12fa4], 0
0x18b1c:    mov     dword ptr [rip + 0x12f96], 0
0x18b26:    add     rsp, 8
0x18b2a:    ret

```

```

0x18b2b:    nop     dword ptr [rax + rax]
0x18b30:    xor     eax, eax
0x18b32:    call    0xa6c0
0x18b37:    xor     eax, eax
0x18b39:    call    0x9d80
0x18b3e:    mov     edi, 0x1c
0x18b43:    call    0x7bd0
0x18b48:    mov     dword ptr [rip + 0x12f6e], 0
0x18b52:    mov     dword ptr [rip + 0x12f60], 0
0x18b5c:    add     rsp, 8
0x18b60:    ret
0x18b61:    nop     word ptr cs:[rax + rax]
0x18b6c:    nop     dword ptr [rax]
0x18b70:    endbr64
0x18b74:    mov     dword ptr [rip + 0x12f3a], edi
0x18b7a:    test    edi, edi
0x18b7c:    jne     0x18b80
0x18b7e:    ret
0x18b7f:    nop
0x18b80:    xor     eax, eax
0x18b82:    jmp     0x18ab0
0x18b87:    nop     word ptr [rax + rax]
0x18b90:    endbr64
0x18b94:    sub     rsp, 8
0x18b98:    mov     eax, dword ptr [rip + 0x11bfa]
0x18b9e:    cmp     edi, 1
0x18ba1:    je      0x18bf0
0x18ba3:    test    eax, eax
0x18ba5:    jne     0x18bd0
0x18ba7:    mov     esi, 1
0x18bac:    mov     edi, 0x1c
0x18bb1:    call    0x4b50
0x18bb6:    mov     esi, 1
0x18bbb:    mov     edi, 0xe
0x18bc0:    add     rsp, 8
0x18bc4:    jmp     0x4b50
0x18bc9:    nop     dword ptr [rax]
0x18bd0:    mov     esi, 1
0x18bd5:    mov     edi, 0x14
0x18bda:    call    0x4b50
0x18bdf:    mov     esi, 1
0x18be4:    mov     edi, 0x12
0x18be9:    call    0x4b50
0x18bee:    jmp     0x18ba7
0x18bf0:    test    eax, eax
0x18bf2:    jne     0x18c20
0x18bf4:    lea     rsi, [rip - 0x20b]
0x18bfb:    mov     edi, 0x1c

```

```

0x18c00:      call    0x4b50
0x18c05:      lea     rsi, [rip - 0x39c]
0x18c0c:      mov     edi, 0xe
0x18c11:      add     rsp, 8
0x18c15:      jmp     0x4b50
0x18c1a:      nop     word ptr [rax + rax]
0x18c20:      lea     rsi, [rip - 0x347]
0x18c27:      mov     edi, 0x14
0x18c2c:      call    0x4b50
0x18c31:      lea     rsi, [rip - 0x2e8]
0x18c38:      mov     edi, 0x12
0x18c3d:      call    0x4b50
0x18c42:      jmp     0x18bf4
0x18c44:      nop     word ptr cs:[rax + rax]
0x18c4f:      nop
0x18c50:      endbr64
0x18c54:      push    rbp
0x18c55:      mov     esi, 1
0x18c5a:      mov     edi, 0x14
0x18c5f:      call    0x4b50
0x18c64:      mov     esi, 1
0x18c69:      mov     edi, 0x12
0x18c6e:      cmp     rax, 1
0x18c72:      setne   al
0x18c75:      movzx   eax, al
0x18c78:      mov     dword ptr [rip + 0x11b1a], eax
0x18c7e:      call    0x4b50
0x18c83:      mov     esi, 1
0x18c88:      mov     edi, 0x1c
0x18c8d:      call    0x4b50
0x18c92:      lea     rbp, [rip - 0x1519]
0x18c99:      mov     edi, 0xb
0x18c9e:      mov     rsi, rbp
0x18ca1:      call    0x4b50
0x18ca6:      mov     rsi, rbp
0x18ca9:      mov     edi, 1
0x18cae:      call    0x4b50
0x18cb3:      mov     rsi, rbp
0x18cb6:      mov     edi, 0xf
0x18cbb:      call    0x4b50
0x18cc0:      mov     esi, 1
0x18cc5:      mov     edi, 3
0x18cca:      call    0x4b50
0x18ccf:      lea     rsi, [rip - 0x516]
0x18cd6:      mov     edi, 2
0x18cdb:      pop     rbp
0x18cdc:      jmp     0x4b50
0x18ce1:      nop     word ptr cs:[rax + rax]

```



```

0x18ceb:      nop      dword ptr [rax + rax]
0x18cf0:      endbr64
0x18cf4:      push     rax
0x18cf5:      pop      rax
0x18cf6:      lea      rdi, [rip + 0x96a8]
0x18cfd:      sub      rsp, 8
0x18d01:      call     0x8b00
0x18d06:      call     0x4850
0x18d0b:      nop      dword ptr [rax + rax]
0x18d10:      endbr64
0x18d14:      test     rdi, rdi
0x18d17:      je       0x18d20
0x18d19:      jmp      0x4800
0x18d1e:      nop
0x18d20:      lea      rdi, [rip + 0x9697]
0x18d27:      jmp      0x8b00
0x18d2c:      nop      dword ptr [rax]
0x18d30:      endbr64
0x18d34:      xor      r8d, r8d
0x18d37:      test     rdx, rdx
0x18d3a:      je       0x18d90
0x18d3c:      push     r14
0x18d3e:      push     r13
0x18d40:      push     r12
0x18d42:      mov      r12, rsi
0x18d45:      push     rbp
0x18d46:      mov      ebp, edi
0x18d48:      push     rbx
0x18d49:      mov      rbx, rdx
0x18d4c:      call     0x4860
0x18d51:      mov      r13d, dword ptr [rax]
0x18d54:      mov      r14, rax
0x18d57:      jmp      0x18d66
0x18d59:      nop      dword ptr [rax]
0x18d60:      cmp      dword ptr [r14], 4
0x18d64:      jne      0x18d7d
0x18d66:      mov      dword ptr [r14], r13d
0x18d69:      mov      rdx, rbx
0x18d6c:      mov      rsi, r12
0x18d6f:      mov      edi, ebp
0x18d71:      call     0x47a0
0x18d76:      mov      r8d, eax
0x18d79:      test     eax, eax
0x18d7b:      js       0x18d60
0x18d7d:      pop      rbx
0x18d7e:      mov      eax, r8d
0x18d81:      pop      rbp
0x18d82:      pop      r12

```

```

0x18d84:    pop     r13
0x18d86:    pop     r14
0x18d88:    ret
0x18d89:    nop     dword ptr [rax]
0x18d90:    mov     eax, r8d
0x18d93:    ret
0x18d94:    nop     word ptr cs:[rax + rax]
0x18d9f:    nop
0x18da0:    endbr64
0x18da4:    xor     r8d, r8d
0x18da7:    test    rdx, rdx
0x18daa:    je      0x18e00
0x18dac:    push    r14
0x18dae:    push    r13
0x18db0:    push    r12
0x18db2:    mov     r12, rsi
0x18db5:    push    rbp
0x18db6:    mov     ebp, edi
0x18db8:    push    rbx
0x18db9:    mov     rbx, rdx
0x18dbc:    call    0x4860
0x18dc1:    mov     r13d, dword ptr [rax]
0x18dc4:    mov     r14, rax
0x18dc7:    jmp     0x18dd6
0x18dc9:    nop     dword ptr [rax]
0x18dd0:    cmp     dword ptr [r14], 4
0x18dd4:    jne     0x18ded
0x18dd6:    mov     dword ptr [r14], r13d
0x18dd9:    mov     rdx, rbx
0x18ddc:    mov     rsi, r12
0x18ddf:    mov     edi, ebp
0x18de1:    call    0x4790
0x18de6:    mov     r8d, eax
0x18de9:    test    eax, eax
0x18deb:    js      0x18dd0
0x18ded:    pop     rbx
0x18dee:    mov     eax, r8d
0x18df1:    pop     rbp
0x18df2:    pop     r12
0x18df4:    pop     r13
0x18df6:    pop     r14
0x18df8:    ret
0x18df9:    nop     dword ptr [rax]
0x18e00:    mov     eax, r8d
0x18e03:    ret
0x18e04:    nop     word ptr cs:[rax + rax]
0x18e0f:    nop
0x18e10:    endbr64

```

```

0x18e14:    xor     r8d, r8d
0x18e17:    test    rdx, rdx
0x18e1a:    je      0x18e70
0x18e1c:    push    r14
0x18e1e:    push    r13
0x18e20:    push    r12
0x18e22:    mov     r12, rsi
0x18e25:    push    rbp
0x18e26:    mov     rbp, rdi
0x18e29:    push    rbx
0x18e2a:    mov     rbx, rdx
0x18e2d:    call    0x4860
0x18e32:    mov     r13d, dword ptr [rax]
0x18e35:    mov     r14, rax
0x18e38:    jmp     0x18e46
0x18e3a:    nop     word ptr [rax + rax]
0x18e40:    cmp     dword ptr [r14], 4
0x18e44:    jne     0x18e5e
0x18e46:    mov     dword ptr [r14], r13d
0x18e49:    mov     rdx, rbx
0x18e4c:    mov     rsi, r12
0x18e4f:    mov     rdi, rbp
0x18e52:    call    0x4910
0x18e57:    mov     r8d, eax
0x18e5a:    test    eax, eax
0x18e5c:    js      0x18e40
0x18e5e:    pop     rbx
0x18e5f:    mov     eax, r8d
0x18e62:    pop     rbp
0x18e63:    pop     r12
0x18e65:    pop     r13
0x18e67:    pop     r14
0x18e69:    ret
0x18e6a:    nop     word ptr [rax + rax]
0x18e70:    mov     eax, r8d
0x18e73:    ret
0x18e74:    nop     word ptr cs:[rax + rax]
0x18e7f:    nop
0x18e80:    endbr64
0x18e84:    push    r13
0x18e86:    mov     r13, rdi
0x18e89:    push    r12
0x18e8b:    push    rbp
0x18e8c:    push    rbx
0x18e8d:    mov     ebx, 0x64
0x18e92:    sub     rsp, 0x18
0x18e96:    jmp     0x18eaa
0x18e98:    nop     dword ptr [rax + rax]

```

```

0x18ea0:    mov     rdi, rbp
0x18ea3:    add     ebx, ebx
0x18ea5:    call    0x18d10
0x18eaa:    movsxd  r12, ebx
0x18ead:    mov     rdi, r12
0x18eb0:    call    0x1d670
0x18eb5:    mov     rdx, r12
0x18eb8:    mov     rdi, r13
0x18ebb:    mov     rbp, rax
0x18ebe:    mov     rsi, rax
0x18ec1:    xor     eax, eax
0x18ec3:    call    0x18e10
0x18ec8:    cmp     ebx, eax
0x18eca:    jle     0x18ea0
0x18ecc:    mov     rdi, rbp
0x18ecf:    mov     dword ptr [rsp + 0xc], eax
0x18ed3:    call    0x18d10
0x18ed8:    mov     eax, dword ptr [rsp + 0xc]
0x18edc:    add     rsp, 0x18
0x18ee0:    pop     rbx
0x18ee1:    pop     rbp
0x18ee2:    pop     r12
0x18ee4:    pop     r13
0x18ee6:    ret
0x18ee7:    nop     word ptr [rax + rax]
0x18ef0:    endbr64
0x18ef4:    push    r13
0x18ef6:    push    r12
0x18ef8:    mov     r12d, edi
0x18efb:    push    rbp
0x18efc:    mov     rbp, rsi
0x18eff:    push    rbx
0x18f00:    sub     rsp, 8
0x18f04:    call    0x4860
0x18f09:    mov     r13d, dword ptr [rax]
0x18f0c:    mov     rbx, rax
0x18f0f:    jmp     0x18f1d
0x18f11:    nop     dword ptr [rax]
0x18f18:    cmp     dword ptr [rbx], 4
0x18f1b:    jne     0x18f2f
0x18f1d:    mov     dword ptr [rbx], r13d
0x18f20:    mov     rsi, rbp
0x18f23:    mov     edi, r12d
0x18f26:    call    0x4eb0
0x18f2b:    test    eax, eax
0x18f2d:    js      0x18f18
0x18f2f:    add     rsp, 8
0x18f33:    pop     rbx

```

```

0x18f34:    pop     rbp
0x18f35:    pop     r12
0x18f37:    pop     r13
0x18f39:    ret
0x18f3a:    nop     word ptr [rax + rax]
0x18f40:    endbr64
0x18f44:    push    r13
0x18f46:    push    r12
0x18f48:    mov     r12, rdi
0x18f4b:    push    rbp
0x18f4c:    mov     rbp, rsi
0x18f4f:    push    rbx
0x18f50:    sub     rsp, 8
0x18f54:    call    0x4860
0x18f59:    mov     r13d, dword ptr [rax]
0x18f5c:    mov     rbx, rax
0x18f5f:    jmp     0x18f6d
0x18f61:    nop     dword ptr [rax]
0x18f68:    cmp     dword ptr [rbx], 4
0x18f6b:    jne     0x18f7f
0x18f6d:    mov     dword ptr [rbx], r13d
0x18f70:    mov     rsi, rbp
0x18f73:    mov     rdi, r12
0x18f76:    call    0x4b90
0x18f7b:    test    eax, eax
0x18f7d:    js      0x18f68
0x18f7f:    add     rsp, 8
0x18f83:    pop     rbx
0x18f84:    pop     rbp
0x18f85:    pop     r12
0x18f87:    pop     r13
0x18f89:    ret
0x18f8a:    nop     word ptr [rax + rax]
0x18f90:    endbr64
0x18f94:    push    r13
0x18f96:    push    r12
0x18f98:    mov     r12, rdi
0x18f9b:    push    rbp
0x18f9c:    mov     rbp, rsi
0x18f9f:    push    rbx
0x18fa0:    sub     rsp, 8
0x18fa4:    call    0x4860
0x18fa9:    mov     r13d, dword ptr [rax]
0x18fac:    mov     rbx, rax
0x18faf:    jmp     0x18fbd
0x18fb1:    nop     dword ptr [rax]
0x18fb8:    cmp     dword ptr [rbx], 4
0x18fbb:    jne     0x18fcf

```

```

0x18fbd:    mov     dword ptr [rbx], r13d
0x18fc0:    mov     rsi, rbp
0x18fc3:    mov     rdi, r12
0x18fc6:    call    0x4ad0
0x18fcb:    test    eax, eax
0x18fcd:    js      0x18fb8
0x18fcf:    add     rsp, 8
0x18fd3:    pop     rbx
0x18fd4:    pop     rbp
0x18fd5:    pop     r12
0x18fd7:    pop     r13
0x18fd9:    ret
0x18fda:    nop     word ptr [rax + rax]
0x18fe0:    endbr64
0x18fe4:    push    r12
0x18fe6:    push    rbp
0x18fe7:    push    rbx
0x18fe8:    mov     ebx, 0x40
0x18fed:    call    0x4860
0x18ff2:    mov     edi, 0x40
0x18ff7:    mov     dword ptr [rax], 0
0x18ffd:    mov     r12, rax
0x19000:    call    0x1d670
0x19005:    mov     rbp, rax
0x19008:    jmp     0x19031
0x1900a:    nop     word ptr [rax + rax]
0x19010:    cmp     dword ptr [r12], 0x22
0x19015:    jne     0x19050
0x19017:    add     rbx, 0x40
0x1901b:    mov     rdi, rbp
0x1901e:    mov     rsi, rbx
0x19021:    call    0x1d690
0x19026:    mov     dword ptr [r12], 0
0x1902e:    mov     rbp, rax
0x19031:    mov     rsi, rbx
0x19034:    mov     rdi, rbp
0x19037:    call    0x4a80
0x1903c:    test    rax, rax
0x1903f:    je      0x19010
0x19041:    mov     rax, rbp
0x19044:    pop     rbx
0x19045:    pop     rbp
0x19046:    pop     r12
0x19048:    ret
0x19049:    nop     dword ptr [rax]
0x19050:    mov     rdi, rbp
0x19053:    xor     ebp, ebp
0x19055:    call    0x18d10

```

```

0x1905a:    mov     rax, rbp
0x1905d:    pop     rbx
0x1905e:    pop     rbp
0x1905f:    pop     r12
0x19061:    ret
0x19062:    nop     word ptr cs:[rax + rax]
0x1906d:    nop     dword ptr [rax]
0x19070:    endbr64
0x19074:    push    rbp
0x19075:    mov     esi, 0x2f
0x1907a:    mov     rbp, rdi
0x1907d:    push    rbx
0x1907e:    sub     rsp, 8
0x19082:    call    0x4a20
0x19087:    test    rax, rax
0x1908a:    je      0x190a8
0x1908c:    mov     rbx, rax
0x1908f:    mov     rdi, rbp
0x19092:    call    0x1d7a0
0x19097:    sub     rbx, rbp
0x1909a:    mov     byte ptr [rax + rbx], 0x30
0x1909e:    add     rsp, 8
0x190a2:    pop     rbx
0x190a3:    pop     rbp
0x190a4:    ret
0x190a5:    nop     dword ptr [rax]
0x190a8:    add     rsp, 8
0x190ac:    lea     rdi, [rip + 0x862b]
0x190b3:    pop     rbx
0x190b4:    pop     rbp
0x190b5:    jmp     0x1d7a0
0x190ba:    nop     word ptr [rax + rax]
0x190c0:    endbr64
0x190c4:    push    rbx
0x190c5:    mov     rbx, rdi
0x190c8:    call    0x49b0
0x190cd:    lea     rax, [rbx + rax - 1]
0x190d2:    cmp     byte ptr [rax], 0x2f
0x190d5:    jne     0x190da
0x190d7:    mov     byte ptr [rax], 0
0x190da:    mov     rdi, rbx
0x190dd:    mov     esi, 0x2f
0x190e2:    call    0x4a20
0x190e7:    lea     rdx, [rax + 1]
0x190eb:    test    rax, rax
0x190ee:    mov     rax, rbx
0x190f1:    pop     rbx
0x190f2:    cmovne  rax, rdx

```

```

0x190f6:    ret
0x190f7:    nop      word ptr [rax + rax]
0x19100:    endbr64
0x19104:    push     rbx
0x19105:    mov      esi, 0x10
0x1910a:    mov      ebx, edi
0x1910c:    mov      edi, 1
0x19111:    call     0x1d740
0x19116:    mov      dword ptr [rax + 8], ebx
0x19119:    pop      rbx
0x1911a:    ret
0x1911b:    nop      dword ptr [rax + rax]
0x19120:    endbr64
0x19124:    test     rdi, rdi
0x19127:    je       0x19148
0x19129:    push     rbp
0x1912a:    mov      rbp, rdi
0x1912d:    mov      rdi, qword ptr [rdi]
0x19130:    test     rdi, rdi
0x19133:    je       0x1913a
0x19135:    call     0x18d10
0x1913a:    mov      rdi, rbp
0x1913d:    pop      rbp
0x1913e:    jmp      0x18d10
0x19143:    nop      dword ptr [rax + rax]
0x19148:    ret
0x19149:    nop      dword ptr [rax]
0x19150:    endbr64
0x19154:    push     rbp
0x19155:    mov      rbp, rsi
0x19158:    push     rbx
0x19159:    mov      rbx, rdi
0x1915c:    sub      rsp, 8
0x19160:    mov      eax, dword ptr [rdi + 0xc]
0x19163:    lea      esi, [rax + 1]
0x19166:    mov      dword ptr [rdi + 0xc], esi
0x19169:    imul     esi, dword ptr [rdi + 8]
0x1916d:    mov      rdi, qword ptr [rdi]
0x19170:    movsxd   rsi, esi
0x19173:    call     0x1d690
0x19178:    mov      ecx, dword ptr [rbx + 0xc]
0x1917b:    movsxd   rdx, dword ptr [rbx + 8]
0x1917f:    mov      rsi, rbp
0x19182:    mov      qword ptr [rbx], rax
0x19185:    add      rsp, 8
0x19189:    sub      ecx, 1
0x1918c:    pop      rbx
0x1918d:    pop      rbp

```



```

0x1918e:      imul     ecx, edx
0x19191:      movsxd   rcx, ecx
0x19194:      lea      rdi, [rax + rcx]
0x19198:      jmp      0x4bb0
0x1919d:      nop      dword ptr [rax]
0x191a0:      endbr64
0x191a4:      mov      eax, dword ptr [rdi + 0xc]
0x191a7:      test     eax, eax
0x191a9:      je       0x191f8
0x191ab:      push     rbp
0x191ac:      sub      eax, 1
0x191af:      mov      rbp, rsi
0x191b2:      push     rbx
0x191b3:      mov      rbx, rdi
0x191b6:      sub      rsp, 8
0x191ba:      movsxd   rdx, dword ptr [rdi + 8]
0x191be:      mov      dword ptr [rdi + 0xc], eax
0x191c1:      mov      rsi, rdx
0x191c4:      imul     esi, eax
0x191c7:      movsxd   rsi, esi
0x191ca:      add      rsi, qword ptr [rdi]
0x191cd:      mov      rdi, rbp
0x191d0:      call     0x4bb0
0x191d5:      mov      esi, dword ptr [rbx + 0xc]
0x191d8:      imul     esi, dword ptr [rbx + 8]
0x191dc:      mov      rdi, qword ptr [rbx]
0x191df:      movsxd   rsi, esi
0x191e2:      call     0x1d690
0x191e7:      mov      qword ptr [rbx], rax
0x191ea:      add      rsp, 8
0x191ee:      mov      rax, rbp
0x191f1:      pop      rbx
0x191f2:      pop      rbp
0x191f3:      ret
0x191f4:      nop      dword ptr [rax]
0x191f8:      xor      eax, eax
0x191fa:      ret
0x191fb:      nop      dword ptr [rax + rax]
0x19200:      endbr64
0x19204:      mov      ecx, dword ptr [rdi + 0xc]
0x19207:      cmp      ecx, edx
0x19209:      jl       0x19240
0x1920b:      test     ecx, ecx
0x1920d:      je       0x19240
0x1920f:      sub      rsp, 8
0x19213:      movsxd   r9, dword ptr [rdi + 8]
0x19217:      sub      ecx, edx
0x19219:      mov      r8, rsi

```

```

0x1921c:    imul    ecx, r9d
0x19220:    mov     rdx, r9
0x19223:    movsxd  rsi, ecx
0x19226:    add     rsi, qword ptr [rdi]
0x19229:    mov     rdi, r8
0x1922c:    call    0x4bb0
0x19231:    add     rsp, 8
0x19235:    ret
0x19236:    nop     word ptr cs:[rax + rax]
0x19240:    xor     eax, eax
0x19242:    ret
0x19243:    nop     word ptr cs:[rax + rax]
0x1924e:    nop
0x19250:    endbr64
0x19254:    push    rbx
0x19255:    mov     rbx, rdi
0x19258:    mov     dword ptr [rdi + 0xc], esi
0x1925b:    imul    esi, dword ptr [rdi + 8]
0x1925f:    mov     rdi, qword ptr [rdi]
0x19262:    movsxd  rsi, esi
0x19265:    call    0x1d690
0x1926a:    mov     qword ptr [rbx], rax
0x1926d:    pop     rbx
0x1926e:    ret
0x1926f:    nop
0x19270:    endbr64
0x19274:    mov     eax, dword ptr [rdi + 0xc]
0x19277:    ret
0x19278:    nop     dword ptr [rax + rax]
0x19280:    push    r12
0x19282:    mov     r12, rdi
0x19285:    mov     edi, 2
0x1928a:    push    rbp
0x1928b:    push    rbx
0x1928c:    mov     ebx, esi
0x1928e:    xor     esi, esi
0x19290:    call    0x4b50
0x19295:    xor     esi, esi
0x19297:    mov     edi, 3
0x1929c:    call    0x4b50
0x192a1:    test    ebx, ebx
0x192a3:    jne     0x192e0
0x192a5:    mov     r9, qword ptr [rip + 0x115b4]
0x192ac:    xor     r8d, r8d
0x192af:    mov     rcx, r12
0x192b2:    xor     eax, eax
0x192b4:    lea     rdi, [rip + 0x7363]
0x192bb:    lea     rdx, [rip + 0x911e]

```

```

0x192c2:    lea     rsi, [rip + 0x735a]
0x192c9:    call    0x4e30
0x192ce:    mov     edi, 0x7f
0x192d3:    call    0x48a0
0x192d8:    nop     dword ptr [rax + rax]
0x192e0:    mov     edi, 1
0x192e5:    lea     rbp, [rip + 0x90ed]
0x192ec:    call    0x4ab0
0x192f1:    mov     edi, 2
0x192f6:    call    0x4ab0
0x192fb:    mov     rdi, qword ptr [rip + 0x127ce]
0x19302:    mov     rsi, rbp
0x19305:    call    0x4da0
0x1930a:    mov     rdi, qword ptr [rip + 0x127b7]
0x19311:    mov     rsi, rbp
0x19314:    call    0x4da0
0x19319:    mov     edx, 4
0x1931e:    lea     rsi, [rip + 0x90b6]
0x19325:    mov     rdi, r12
0x19328:    call    0x4890
0x1932d:    test    eax, eax
0x1932f:    je      0x192a5
0x19335:    xor     edi, edi
0x19337:    call    0x4ab0
0x1933c:    jmp     0x192a5
0x19341:    nop     word ptr cs:[rax + rax]
0x1934c:    nop     dword ptr [rax]
0x19350:    endbr64
0x19354:    push    r12
0x19356:    push    rbp
0x19357:    push    rbx
0x19358:    sub     rsp, 0x10
0x1935c:    mov     rax, qword ptr fs:[0x28]
0x19365:    mov     qword ptr [rsp + 8], rax
0x1936a:    xor     eax, eax
0x1936c:    test    rdi, rdi
0x1936f:    je      0x193d0
0x19371:    mov     rbp, rdi
0x19374:    mov     r12d, esi
0x19377:    call    0x4ed0
0x1937c:    mov     ebx, eax
0x1937e:    test    eax, eax
0x19380:    js      0x193c0
0x19382:    je      0x193dc
0x19384:    lea     rbp, [rsp + 4]
0x19389:    nop     dword ptr [rax]
0x19390:    mov     rdi, rbp
0x19393:    call    0x4ec0

```

```

0x19398:      cmp     eax, ebx
0x1939a:      jne     0x19390
0x1939c:      mov     eax, dword ptr [rsp + 4]
0x193a0:      mov     rdx, qword ptr [rsp + 8]
0x193a5:      sub     rdx, qword ptr fs:[0x28]
0x193ae:      jne     0x193d7
0x193b0:      add     rsp, 0x10
0x193b4:      pop     rbx
0x193b5:      pop     rbp
0x193b6:      pop     r12
0x193b8:      ret
0x193b9:      nop     dword ptr [rax]
0x193c0:      mov     eax, 0xffffffff
0x193c5:      jmp     0x193a0
0x193c7:      nop     word ptr [rax + rax]
0x193d0:      mov     eax, 1
0x193d5:      jmp     0x193a0
0x193d7:      call    0x49d0
0x193dc:      mov     esi, r12d
0x193df:      mov     rdi, rbp
0x193e2:      xor     eax, eax
0x193e4:      call    0x19280
0x193e9:      nop     dword ptr [rax]
0x193f0:      endbr64
0x193f4:      push    r13
0x193f6:      mov     r13, rdi
0x193f9:      push    r12
0x193fb:      push    rbp
0x193fc:      push    rbx
0x193fd:      sub     rsp, 0x18
0x19401:      mov     rax, qword ptr fs:[0x28]
0x1940a:      mov     qword ptr [rsp + 8], rax
0x1940f:      xor     eax, eax
0x19411:      test    esi, esi
0x19413:      je      0x19488
0x19415:      mov     edi, 1
0x1941a:      mov     ebp, esi
0x1941c:      call    0x18b70
0x19421:      test    r13, r13
0x19424:      je      0x19520
0x1942a:      call    0x4ed0
0x1942f:      mov     r12d, eax
0x19432:      test    eax, eax
0x19434:      js      0x19550
0x1943a:      je      0x19560
0x19440:      lea     rbx, [rsp + 4]
0x19445:      nop     dword ptr [rax]
0x19448:      mov     rdi, rbx

```

```

0x1944b:    call    0x4ec0
0x19450:    cmp     r12d, eax
0x19453:    jne     0x19448
0x19455:    mov     r12d, dword ptr [rsp + 4]
0x1945a:    xor     edi, edi
0x1945c:    call    0x18b70
0x19461:    mov     rax, qword ptr [rsp + 8]
0x19466:    sub     rax, qword ptr fs:[0x28]
0x1946f:    jne     0x1955b
0x19475:    add     rsp, 0x18
0x19479:    mov     eax, r12d
0x1947c:    pop     rbx
0x1947d:    pop     rbp
0x1947e:    pop     r12
0x19480:    pop     r13
0x19482:    ret
0x19483:    nop     dword ptr [rax + rax]
0x19488:    xor     edi, edi
0x1948a:    call    0x9b00
0x1948f:    xor     eax, eax
0x19491:    call    0xa6c0
0x19496:    mov     rdi, qword ptr [rip + 0x1142b]
0x1949d:    call    0xbb40
0x194a2:    xor     edi, edi
0x194a4:    call    0x18b90
0x194a9:    test    r13, r13
0x194ac:    je      0x19530
0x194b2:    call    0x4ed0
0x194b7:    mov     ebp, eax
0x194b9:    test    eax, eax
0x194bb:    js      0x19540
0x194c1:    je      0x1956c
0x194c7:    lea     rbx, [rsp + 4]
0x194cc:    nop     dword ptr [rax]
0x194d0:    mov     rdi, rbx
0x194d3:    call    0x4ec0
0x194d8:    cmp     ebp, eax
0x194da:    jne     0x194d0
0x194dc:    mov     r12d, dword ptr [rsp + 4]
0x194e1:    mov     edi, 1
0x194e6:    call    0x18b90
0x194eb:    mov     edx, 2
0x194f0:    mov     edi, 1
0x194f5:    lea     rsi, [rip + 0x8ee7]
0x194fc:    call    0x18da0
0x19501:    mov     edi, 1
0x19506:    call    0x9b00
0x1950b:    xor     eax, eax

```

```

0x1950d:    call    0xa6c0
0x19512:    xor     edi, edi
0x19514:    call    0x7890
0x19519:    jmp     0x19461
0x1951e:    nop
0x19520:    mov     r12d, 1
0x19526:    jmp     0x1945a
0x1952b:    nop     dword ptr [rax + rax]
0x19530:    mov     r12d, 1
0x19536:    jmp     0x194e1
0x19538:    nop     dword ptr [rax + rax]
0x19540:    mov     r12d, 0xffffffff
0x19546:    jmp     0x194e1
0x19548:    nop     dword ptr [rax + rax]
0x19550:    mov     r12d, 0xffffffff
0x19556:    jmp     0x1945a
0x1955b:    call    0x49d0
0x19560:    mov     esi, ebp
0x19562:    mov     rdi, r13
0x19565:    xor     eax, eax
0x19567:    call    0x19280
0x1956c:    xor     esi, esi
0x1956e:    mov     rdi, r13
0x19571:    xor     eax, eax
0x19573:    call    0x19280
0x19578:    nop     dword ptr [rax + rax]
0x19580:    endbr64
0x19584:    mov     rdi, qword ptr [rip + 0x12545]
0x1958b:    test    rdi, rdi
0x1958e:    je      0x195b8
0x19590:    sub     rsp, 8
0x19594:    call    0x4870
0x19599:    mov     rdi, qword ptr [rip + 0x12528]
0x195a0:    test    rdi, rdi
0x195a3:    je      0x195b0
0x195a5:    add     rsp, 8
0x195a9:    jmp     0x4870
0x195ae:    nop
0x195b0:    add     rsp, 8
0x195b4:    ret
0x195b5:    nop     dword ptr [rax]
0x195b8:    mov     rdi, qword ptr [rip + 0x12509]
0x195bf:    test    rdi, rdi
0x195c2:    je      0x195d0
0x195c4:    jmp     0x4870
0x195c9:    nop     dword ptr [rax]
0x195d0:    ret
0x195d1:    nop     word ptr cs:[rax + rax]

```

```

0x195dc:      nop      dword ptr [rax]
0x195e0:      endbr64
0x195e4:      push     r12
0x195e6:      lea      rsi, [rip + 0x89dc]
0x195ed:      push     rbp
0x195ee:      mov      rbp, rdi
0x195f1:      sub      rsp, 8
0x195f5:      mov      rdi, qword ptr [rip + 0x124cc]
0x195fc:      call     0x4da0
0x19601:      test     rax, rax
0x19604:      je       0x19676
0x19606:      mov      edi, 0x801
0x1960b:      mov      r12, rax
0x1960e:      call     0x1d670
0x19613:      mov      rbp, rax
0x19616:      jmp      0x1964a
0x19618:      nop      dword ptr [rax + rax]
0x19620:      mov      rdi, rbp
0x19623:      call     0x49b0
0x19628:      cdqe
0x1962a:      lea      rax, [rbp + rax - 1]
0x1962f:      cmp      byte ptr [rax], 0xa
0x19632:      jne      0x19637
0x19634:      mov      byte ptr [rax], 0
0x19637:      xor      esi, esi
0x19639:      mov      edx, 0xa
0x1963e:      mov      rdi, rbp
0x19641:      call     0x8070
0x19646:      test     al, al
0x19648:      je       0x1965f
0x1964a:      mov      rdx, r12
0x1964d:      mov      esi, 0x801
0x19652:      mov      rdi, rbp
0x19655:      call     0x4b00
0x1965a:      test     rax, rax
0x1965d:      jne      0x19620
0x1965f:      mov      rdi, rbp
0x19662:      call     0x18d10
0x19667:      add      rsp, 8
0x1966b:      mov      rdi, r12
0x1966e:      pop      rbp
0x1966f:      pop      r12
0x19671:      jmp      0x4940
0x19676:      mov      rdi, rbp
0x19679:      call     0x49b0
0x1967e:      mov      rdi, qword ptr [rip + 0x12443]
0x19685:      mov      r12, rax
0x19688:      call     0x49b0

```

```

0x1968d:    lea     rdi, [r12 + rax + 0x20]
0x19692:    call   0x1d670
0x19697:    mov     r9, qword ptr [rip + 0x1242a]
0x1969e:    mov     r8, rbp
0x196a1:    or      rdx, 0xffffffffffffffff
0x196a5:    mov     r12, rax
0x196a8:    mov     rdi, rax
0x196ab:    mov     esi, 1
0x196b0:    xor     eax, eax
0x196b2:    lea     rcx, [rip + 0x8d2d]
0x196b9:    call   0x4f10
0x196be:    mov     rdi, r12
0x196c1:    mov     edx, 0x1e
0x196c6:    xor     esi, esi
0x196c8:    call   0x8070
0x196cd:    pop     rax
0x196ce:    mov     rdi, r12
0x196d1:    pop     rbp
0x196d2:    pop     r12
0x196d4:    jmp     0x18d10
0x196d9:    nop     dword ptr [rax]
0x196e0:    endbr64
0x196e4:    push    r15
0x196e6:    push    r14
0x196e8:    push    r13
0x196ea:    push    r12
0x196ec:    push    rbp
0x196ed:    push    rbx
0x196ee:    sub     rsp, 8
0x196f2:    test    rdi, rdi
0x196f5:    je      0x19878
0x196fb:    mov     rbp, rdi
0x196fe:    call   0x49b0
0x19703:    lea     rdi, [rax + 1]
0x19707:    call   0x1d670
0x1970c:    mov     rsi, rbp
0x1970f:    mov     rdi, rax
0x19712:    call   0x48b0
0x19717:    cmp     byte ptr [rax], 0x7e
0x1971a:    mov     rbp, rax
0x1971d:    mov     r12, rax
0x19720:    jne     0x19800
0x19726:    movzx   eax, byte ptr [rax + 1]
0x1972a:    lea     r13, [rbp + 1]
0x1972e:    test    al, al
0x19730:    je      0x19818
0x19736:    cmp     al, 0x2f
0x19738:    je      0x19818

```



```

0x1973e:      mov     rdi, rbp
0x19741:      call    0x49b0
0x19746:      mov     rdi, rax
0x19749:      call    0x1d670
0x1974e:      movzx   ecx, byte ptr [rbp + 1]
0x19752:      mov     r14, rax
0x19755:      cmp     cl, 0x2f
0x19758:      je      0x19938
0x1975e:      test    cl, cl
0x19760:      je      0x19938
0x19766:      mov     edx, 2
0x1976b:      nop     dword ptr [rax + rax]
0x19770:      mov     byte ptr [r14 + rdx - 2], cl
0x19775:      movzx   ecx, byte ptr [rbp + rdx]
0x1977a:      lea     r13, [rbp + rdx]
0x1977f:      mov     rax, rdx
0x19782:      add     rdx, 1
0x19786:      test    cl, cl
0x19788:      je      0x1978f
0x1978a:      cmp     cl, 0x2f
0x1978d:      jne     0x19770
0x1978f:      lea     rax, [r14 + rax - 1]
0x19794:      mov     byte ptr [rax], 0
0x19797:      mov     rdi, r14
0x1979a:      call    0x4b60
0x1979f:      mov     rbx, rax
0x197a2:      test    rax, rax
0x197a5:      je      0x19880
0x197ab:      mov     rdi, qword ptr [rax + 0x20]
0x197af:      call    0x49b0
0x197b4:      mov     rdi, r13
0x197b7:      mov     r15, rax
0x197ba:      call    0x49b0
0x197bf:      lea     edi, [r15 + 1]
0x197c3:      movsxd  rdi, edi
0x197c6:      add     rdi, rax
0x197c9:      call    0x1d670
0x197ce:      mov     rsi, qword ptr [rbx + 0x20]
0x197d2:      mov     rdi, rax
0x197d5:      mov     r12, rax
0x197d8:      call    0x48b0
0x197dd:      movsxd  rdi, r15d
0x197e0:      mov     rsi, r13
0x197e3:      add     rdi, r12
0x197e6:      call    0x48b0
0x197eb:      mov     rdi, rbp
0x197ee:      call    0x18d10
0x197f3:      call    0x4c70

```

```

0x197f8:      mov     rdi, r14
0x197fb:      call    0x18d10
0x19800:      add     rsp, 8
0x19804:      mov     rax, r12
0x19807:      pop     rbx
0x19808:      pop     rbp
0x19809:      pop     r12
0x1980b:      pop     r13
0x1980d:      pop     r14
0x1980f:      pop     r15
0x19811:      ret
0x19812:      nop     word ptr [rax + rax]
0x19818:      lea     rdi, [rip + 0x6d41]
0x1981f:      call    0x47e0
0x19824:      mov     r14, rax
0x19827:      test    rax, rax
0x1982a:      je      0x198f8
0x19830:      mov     rdi, r14
0x19833:      call    0x49b0
0x19838:      mov     rdi, r13
0x1983b:      mov     rbx, rax
0x1983e:      call    0x49b0
0x19843:      movsxd  rbx, ebx
0x19846:      lea     rdi, [rbx + rax + 1]
0x1984b:      call    0x1d670
0x19850:      mov     rsi, r14
0x19853:      mov     r12, rax
0x19856:      mov     rdi, rax
0x19859:      call    0x48b0
0x1985e:      lea     rdi, [r12 + rbx]
0x19862:      mov     rsi, r13
0x19865:      call    0x48b0
0x1986a:      mov     rdi, rbp
0x1986d:      call    0x18d10
0x19872:      jmp     0x19800
0x19874:      nop     dword ptr [rax]
0x19878:      xor     r12d, r12d
0x1987b:      jmp     0x19800
0x1987d:      nop     dword ptr [rax]
0x19880:      mov     rdx, qword ptr [rip + 0x12251]
0x19887:      test    rdx, rdx
0x1988a:      je      0x197f3
0x19890:      mov     rdi, r14
0x19893:      xor     eax, eax
0x19895:      call    rdx
0x19897:      mov     r15, rax
0x1989a:      test    rax, rax
0x1989d:      je      0x197f3

```

```

0x198a3:    mov     rdi, rax
0x198a6:    call    0x49b0
0x198ab:    mov     rdi, r13
0x198ae:    mov     rbx, rax
0x198b1:    call    0x49b0
0x198b6:    lea     edi, [rbx + 1]
0x198b9:    movsxd  rdi, edi
0x198bc:    add     rdi, rax
0x198bf:    call    0x1d670
0x198c4:    mov     rsi, r15
0x198c7:    mov     rdi, rax
0x198ca:    mov     r12, rax
0x198cd:    call    0x48b0
0x198d2:    movsxd  rdi, ebx
0x198d5:    mov     rsi, r13
0x198d8:    add     rdi, r12
0x198db:    call    0x48b0
0x198e0:    mov     rdi, r15
0x198e3:    call    0x18d10
0x198e8:    mov     rdi, rbp
0x198eb:    call    0x18d10
0x198f0:    jmp     0x197f3
0x198f5:    nop     dword ptr [rax]
0x198f8:    call    0x49e0
0x198fd:    mov     edi, eax
0x198ff:    call    0x4970
0x19904:    test    rax, rax
0x19907:    je      0x19916
0x19909:    mov     r14, qword ptr [rax + 0x20]
0x1990d:    test    r14, r14
0x19910:    jne     0x19830
0x19916:    mov     rdi, r13
0x19919:    call    0x49b0
0x1991e:    lea     rdi, [rax + 1]
0x19922:    call    0x1d670
0x19927:    mov     r12, rax
0x1992a:    mov     rdi, rax
0x1992d:    jmp     0x19862
0x19932:    nop     word ptr [rax + rax]
0x19938:    mov     rax, r14
0x1993b:    jmp     0x19794
0x19940:    endbr64
0x19944:    push    r15
0x19946:    push    r14
0x19948:    mov     r14, rdi
0x1994b:    push    r13
0x1994d:    push    r12
0x1994f:    push    rbp

```

```

0x19950:    push    rbx
0x19951:    xor     ebx, ebx
0x19953:    sub     rsp, 0x48
0x19957:    mov     dword ptr [rsp + 0x1c], 0
0x1995f:    mov     qword ptr [rsp + 0x10], 0
0x19968:    nop     dword ptr [rax + rax]
0x19970:    mov     r15, qword ptr [rip + 0x10e31]
0x19977:    mov     rdi, r14
0x1997a:    mov     qword ptr [rsp + 8], r15
0x1997f:    call    0x49b0
0x19984:    mov     qword ptr [rsp + 0x20], rax
0x19989:    mov     rcx, rax
0x1998c:    movzx   eax, byte ptr [r14]
0x19990:    test    al, al
0x19992:    je      0x19bd0
0x19998:    cmp     al, 0x7e
0x1999a:    je      0x19bd0
0x199a0:    mov     dword ptr [rsp + 0x18], ecx
0x199a4:    mov     rax, rcx
0x199a7:    test    r15, r15
0x199aa:    je      0x19c50
0x199b0:    mov     edx, dword ptr [rsp + 0x20]
0x199b4:    test    edx, edx
0x199b6:    jle     0x19c4a
0x199bc:    mov     rax, qword ptr [rsp + 8]
0x199c1:    mov     dword ptr [rsp + 0x3c], ebx
0x199c5:    xor     r12d, r12d
0x199c8:    mov     rax, qword ptr [rax]
0x199cb:    mov     qword ptr [rsp + 0x28], rax
0x199d0:    mov     rax, qword ptr [rsp + 0x20]
0x199d5:    sub     eax, 1
0x199d8:    mov     qword ptr [rsp + 0x30], rax
0x199dd:    mov     rax, qword ptr [rsp + 8]
0x199e2:    mov     dword ptr [rsp + 0x38], r12d
0x199e7:    lea     rbp, [r14 + r12]
0x199eb:    lea     rbx, [rax + 8]
0x199ef:    mov     rax, qword ptr [rsp + 0x28]
0x199f4:    mov     r13, rax
0x199f7:    test    rax, rax
0x199fa:    jne     0x19a18
0x199fc:    jmp     0x19bb0
0x19a01:    nop     dword ptr [rax]
0x19a08:    mov     r13, qword ptr [rbx]
0x19a0b:    add     rbx, 8
0x19a0f:    test    r13, r13
0x19a12:    je      0x19bb0
0x19a18:    mov     rdi, r13
0x19a1b:    call    0x49b0

```

```

0x19a20:    mov     rsi, r13
0x19a23:    mov     rdi, rbp
0x19a26:    mov     rdx, rax
0x19a29:    mov     r15, rax
0x19a2c:    call    0x4890
0x19a31:    test    eax, eax
0x19a33:    jne     0x19a08
0x19a35:    mov     eax, dword ptr [rsp + 0x38]
0x19a39:    mov     ebx, dword ptr [rsp + 0x3c]
0x19a3d:    lea     eax, [rax + r15 - 1]
0x19a42:    lea     ecx, [rbx + rax]
0x19a45:    mov     dword ptr [rsp + 0x18], eax
0x19a49:    movsxd  r13, eax
0x19a4c:    mov     dword ptr [rsp + 8], ecx
0x19a50:    lea     r15, [r14 + r13]
0x19a54:    mov     ecx, dword ptr [rsp + 8]
0x19a58:    cmp     dword ptr [rsp + 0x1c], ecx
0x19a5c:    jle     0x19bf0
0x19a62:    movsxd  rbx, ebx
0x19a65:    add     rbx, qword ptr [rsp + 0x10]
0x19a6a:    mov     rdx, r13
0x19a6d:    mov     rsi, r14
0x19a70:    mov     rdi, rbx
0x19a73:    mov     qword ptr [rsp + 0x28], rbx
0x19a78:    call    0x4880
0x19a7d:    mov     rdi, r15
0x19a80:    mov     r13, qword ptr [rip + 0x10d19]
0x19a87:    call    0x49b0
0x19a8c:    mov     dword ptr [rsp + 0x20], eax
0x19a90:    test    eax, eax
0x19a92:    jle     0x19c64
0x19a98:    mov     rbx, r15
0x19a9b:    xor     ebp, ebp
0x19a9d:    movzx   eax, byte ptr [rbx]
0x19aa0:    cmp     al, 0x2f
0x19aa2:    je      0x19b80
0x19aa8:    test    al, al
0x19aaa:    je      0x19b80
0x19ab0:    test    r13, r13
0x19ab3:    je      0x19aea
0x19ab5:    mov     r14, r13
0x19ab8:    jmp     0x19ae2
0x19aba:    nop     word ptr [rax + rax]
0x19ac0:    mov     rdi, r12
0x19ac3:    add     r14, 8
0x19ac7:    call    0x49b0
0x19acc:    mov     rsi, r12
0x19acf:    mov     rdi, rbx

```

```

0x19ad2:    mov     rdx, rax
0x19ad5:    call    0x4890
0x19ada:    test    eax, eax
0x19adc:    je      0x19b80
0x19ae2:    mov     r12, qword ptr [r14]
0x19ae5:    test    r12, r12
0x19ae8:    jne     0x19ac0
0x19aea:    add     ebp, 1
0x19aed:    add     rbx, 1
0x19af1:    cmp     dword ptr [rsp + 0x20], ebp
0x19af5:    jne     0x19a9d
0x19af7:    nop     word ptr [rax + rax]
0x19b00:    lea     edi, [rbp + 1]
0x19b03:    movsxd  rbp, ebp
0x19b06:    movsxd  rdi, edi
0x19b09:    lea     r14, [r15 + rbp]
0x19b0d:    call    0x1d670
0x19b12:    mov     rdx, rbp
0x19b15:    mov     rsi, r15
0x19b18:    mov     r13, rax
0x19b1b:    mov     rdi, rax
0x19b1e:    call    0x4880
0x19b23:    mov     byte ptr [r13 + rbp], 0
0x19b29:    mov     rdi, r13
0x19b2c:    xor     eax, eax
0x19b2e:    call    0x196e0
0x19b33:    mov     rdi, r13
0x19b36:    mov     rbp, rax
0x19b39:    call    0x18d10
0x19b3e:    mov     rdi, rbp
0x19b41:    call    0x49b0
0x19b46:    mov     ecx, dword ptr [rsp + 8]
0x19b4a:    lea     ebx, [rcx + rax]
0x19b4d:    cmp     ebx, dword ptr [rsp + 0x1c]
0x19b51:    jge     0x19c20
0x19b57:    movsxd  rdi, dword ptr [rsp + 8]
0x19b5c:    mov     rsi, rbp
0x19b5f:    add     rdi, qword ptr [rsp + 0x10]
0x19b64:    call    0x48b0
0x19b69:    mov     rdi, rbp
0x19b6c:    call    0x18d10
0x19b71:    jmp     0x19970
0x19b76:    nop     word ptr cs:[rax + rax]
0x19b80:    or      dword ptr [rsp + 0x18], ebp
0x19b84:    mov     eax, dword ptr [rsp + 0x18]
0x19b88:    test    eax, eax
0x19b8a:    jne     0x19b00
0x19b90:    mov     rax, qword ptr [rsp + 0x28]

```

```

0x19b95:    mov     byte ptr [rax], 0
0x19b98:    mov     rax, qword ptr [rsp + 0x10]
0x19b9d:    add     rsp, 0x48
0x19ba1:    pop     rbx
0x19ba2:    pop     rbp
0x19ba3:    pop     r12
0x19ba5:    pop     r13
0x19ba7:    pop     r14
0x19ba9:    pop     r15
0x19bab:    ret
0x19bac:    nop     dword ptr [rax]
0x19bb0:    lea     rax, [r12 + 1]
0x19bb5:    cmp     qword ptr [rsp + 0x30], r12
0x19bba:    je      0x19c46
0x19bc0:    mov     r12, rax
0x19bc3:    jmp     0x199dd
0x19bc8:    nop     dword ptr [rax + rax]
0x19bd0:    mov     dword ptr [rsp + 8], ebx
0x19bd4:    mov     r15, r14
0x19bd7:    mov     ecx, dword ptr [rsp + 8]
0x19bdb:    xor     r13d, r13d
0x19bde:    mov     dword ptr [rsp + 0x18], 0
0x19be6:    cmp     dword ptr [rsp + 0x1c], ecx
0x19bea:    jg      0x19a62
0x19bf0:    mov     eax, dword ptr [rsp + 0x1c]
0x19bf4:    mov     ecx, dword ptr [rsp + 0x18]
0x19bf8:    mov     rdi, qword ptr [rsp + 0x10]
0x19bfd:    lea     eax, [rax + rcx + 0x14]
0x19c01:    lea     esi, [rax + 1]
0x19c04:    mov     dword ptr [rsp + 0x1c], eax
0x19c08:    movsxd  rsi, esi
0x19c0b:    call    0x1d690
0x19c10:    mov     qword ptr [rsp + 0x10], rax
0x19c15:    jmp     0x19a62
0x19c1a:    nop     word ptr [rax + rax]
0x19c20:    mov     ecx, dword ptr [rsp + 0x1c]
0x19c24:    mov     rdi, qword ptr [rsp + 0x10]
0x19c29:    lea     eax, [rcx + rax + 0x14]
0x19c2d:    lea     esi, [rax + 1]
0x19c30:    mov     dword ptr [rsp + 0x1c], eax
0x19c34:    movsxd  rsi, esi
0x19c37:    call    0x1d690
0x19c3c:    mov     qword ptr [rsp + 0x10], rax
0x19c41:    jmp     0x19b57
0x19c46:    mov     ebx, dword ptr [rsp + 0x3c]
0x19c4a:    mov     rax, qword ptr [rsp + 0x20]
0x19c4f:    nop
0x19c50:    mov     ecx, eax

```

```

0x19c52:      movsxd  r13, eax
0x19c55:      add     ecx, ebx
0x19c57:      lea     r15, [r14 + r13]
0x19c5b:      mov     dword ptr [rsp + 8], ecx
0x19c5f:      jmp     0x19a54
0x19c64:      xor     ebp, ebp
0x19c66:      jmp     0x19b84
0x19c6b:      nop     dword ptr [rax + rax]
0x19c70:      push    r15
0x19c72:      mov     r15, rsi
0x19c75:      push    r14
0x19c77:      mov     r14d, edx
0x19c7a:      push    r13
0x19c7c:      push    r12
0x19c7e:      mov     r12, rdi
0x19c81:      push    rbp
0x19c82:      push    rbx
0x19c83:      sub     rsp, 0x18
0x19c87:      movsxd  rsi, dword ptr [rsi]
0x19c8a:      lea     r13, [rdi + rsi]
0x19c8e:      mov     edi, ecx
0x19c90:      mov     ebx, esi
0x19c92:      mov     rbp, rsi
0x19c95:      movzx   eax, byte ptr [r13]
0x19c9a:      and     edi, 1
0x19c9d:      mov     rdx, rsi
0x19ca0:      test    al, al
0x19ca2:      jne     0x19cdc
0x19ca4:      jmp     0x19db9
0x19ca9:      nop     dword ptr [rax]
0x19cb0:      cmp     al, 0x5c
0x19cb2:      jne     0x19cc5
0x19cb4:      movsx   eax, byte ptr [r12 + rdx + 1]
0x19cba:      cmp     eax, r14d
0x19cbd:      sete    al
0x19cc0:      movzx   eax, al
0x19cc3:      add     ebx, eax
0x19cc5:      add     ebx, 1
0x19cc8:      movsxd  rdx, ebx
0x19ccb:      lea     r13, [r12 + rdx]
0x19ccf:      movzx   eax, byte ptr [r13]
0x19cd4:      test    al, al
0x19cd6:      je      0x19da0
0x19cdc:      movsx   ecx, al
0x19cdf:      cmp     ecx, r14d
0x19ce2:      jne     0x19cb0
0x19ce4:      cmp     ebp, ebx
0x19ce6:      jl      0x19d10

```



```

0x19ce8:      test    dil, dil
0x19ceb:      jne     0x19d10
0x19ced:      movzx   eax, byte ptr [r13]
0x19cf2:      xor     r9d, r9d
0x19cf5:      cmp     al, 1
0x19cf7:      sbb     ebx, -1
0x19cfa:      mov     dword ptr [r15], ebx
0x19cfd:      add     rsp, 0x18
0x19d01:      mov     rax, r9
0x19d04:      pop     rbx
0x19d05:      pop     rbp
0x19d06:      pop     r12
0x19d08:      pop     r13
0x19d0a:      pop     r14
0x19d0c:      pop     r15
0x19d0e:      ret
0x19d0f:      nop
0x19d10:      mov     edi, ebx
0x19d12:      mov     qword ptr [rsp], r8
0x19d16:      sub     edi, ebp
0x19d18:      mov     qword ptr [rsp + 8], rsi
0x19d1d:      add     edi, 1
0x19d20:      movsxd  rdi, edi
0x19d23:      call    0x1d670
0x19d28:      cmp     ebp, ebx
0x19d2a:      mov     r8, qword ptr [rsp]
0x19d2e:      mov     r9, rax
0x19d31:      jge     0x19dd6
0x19d37:      mov     rsi, qword ptr [rsp + 8]
0x19d3c:      mov     eax, 1
0x19d41:      jmp     0x19d5d
0x19d43:      nop     dword ptr [rax + rax]
0x19d48:      add     ebp, 1
0x19d4b:      mov     byte ptr [r9 + rax - 1], dl
0x19d50:      mov     edx, eax
0x19d52:      add     rax, 1
0x19d56:      cmp     ebp, ebx
0x19d58:      jge     0x19d80
0x19d5a:      movsxd  rsi, ebp
0x19d5d:      movzx   edx, byte ptr [r12 + rsi]
0x19d62:      cmp     dl, 0x5c
0x19d65:      jne     0x19d48
0x19d67:      movsx   esi, byte ptr [r12 + rsi + 1]
0x19d6d:      cmp     esi, r14d
0x19d70:      jne     0x19d48
0x19d72:      add     ebp, 1
0x19d75:      mov     edx, esi
0x19d77:      jmp     0x19d48

```

```

0x19d79:      nop      dword ptr [rax]
0x19d80:      movsxd   rax, edx
0x19d83:      add      rax, r9
0x19d86:      mov      byte ptr [rax], 0
0x19d89:      mov      dword ptr [r8], edx
0x19d8c:      movzx     eax, byte ptr [r13]
0x19d91:      jmp      0x19cf5
0x19d96:      nop      word ptr cs:[rax + rax]
0x19da0:      cmp      ebp, ebx
0x19da2:      jl       0x19d10
0x19da8:      test     dil, dil
0x19dab:      jne      0x19d10
0x19db1:      xor      r9d, r9d
0x19db4:      jmp      0x19cfa
0x19db9:      test     ecx, ecx
0x19dbb:      je       0x19db1
0x19dbd:      mov      edi, 1
0x19dc2:      mov      qword ptr [rsp], r8
0x19dc6:      call     0x1d670
0x19dcb:      mov      r8, qword ptr [rsp]
0x19dcf:      xor      edx, edx
0x19dd1:      mov      r9, rax
0x19dd4:      jmp      0x19d86
0x19dd6:      xor      edx, edx
0x19dd8:      jmp      0x19d86
0x19dda:      nop      word ptr [rax + rax]
0x19de0:      push     r15
0x19de2:      push     r14
0x19de4:      push     r13
0x19de6:      push     r12
0x19de8:      push     rbp
0x19de9:      push     rbx
0x19dea:      sub      rsp, 0x28
0x19dee:      mov      eax, dword ptr [rip + 0x11cf0]
0x19df4:      mov      ebx, dword ptr [rip + 0x11d02]
0x19dfa:      mov      dword ptr [rsp + 8], esi
0x19dfe:      mov      dword ptr [rsp + 0x18], edx
0x19e02:      mov      dword ptr [rsp + 0x1c], eax
0x19e06:      test     eax, eax
0x19e08:      je       0x19f9d
0x19e0e:      mov      rbp, rdi
0x19e11:      cmp      eax, ebx
0x19e13:      je       0x19fb0
0x19e19:      call     0x49b0
0x19e1e:      mov      ecx, dword ptr [rsp + 8]
0x19e22:      mov      dword ptr [rsp + 0xc], eax
0x19e26:      test     ecx, ecx
0x19e28:      js       0x19fbb

```

```

0x19e2e:    mov     rax, qword ptr [rip + 0x11ccb]
0x19e35:    movsxd  r14, dword ptr [rsp + 0xc]
0x19e3a:    mov     qword ptr [rsp + 0x10], rax
0x19e3f:    mov     edx, dword ptr [rsp + 8]
0x19e43:    test    edx, edx
0x19e45:    js      0x19f16
0x19e4b:    cmp     dword ptr [rsp + 0x1c], ebx
0x19e4f:    je      0x19f9d
0x19e55:    mov     rcx, qword ptr [rsp + 0x10]
0x19e5a:    movsxd  rax, ebx
0x19e5d:    mov     rax, qword ptr [rcx + rax*8]
0x19e61:    mov     r13, qword ptr [rax]
0x19e64:    mov     rdi, r13
0x19e67:    call    0x49b0
0x19e6c:    mov     r10, rax
0x19e6f:    cmp     dword ptr [rsp + 0xc], eax
0x19e73:    jg      0x19f07
0x19e79:    cmp     dword ptr [rsp + 0x18], 1
0x19e7e:    movzx   r12d, byte ptr [rbp]
0x19e83:    je      0x19ee0
0x19e85:    mov     r15d, dword ptr [rsp + 0x18]
0x19e8a:    sub     r10d, dword ptr [rsp + 0xc]
0x19e8f:    jmp     0x19ea5
0x19e91:    nop     dword ptr [rax]
0x19e98:    add     r15d, 1
0x19e9c:    add     r13, 1
0x19ea0:    cmp     r15d, r10d
0x19ea3:    jg      0x19f07
0x19ea5:    cmp     byte ptr [r13], r12b
0x19ea9:    jne     0x19e98
0x19eab:    mov     rdx, r14
0x19eae:    mov     rsi, r13
0x19eb1:    mov     rdi, rbp
0x19eb4:    mov     dword ptr [rsp + 4], r10d
0x19eb9:    call    0x4890
0x19ebe:    mov     r10d, dword ptr [rsp + 4]
0x19ec3:    test    eax, eax
0x19ec5:    jne     0x19e98
0x19ec7:    mov     dword ptr [rip + 0x11c2f], ebx
0x19ecd:    add     rsp, 0x28
0x19ed1:    mov     eax, r15d
0x19ed4:    pop     rbx
0x19ed5:    pop     rbp
0x19ed6:    pop     r12
0x19ed8:    pop     r13
0x19eda:    pop     r14
0x19edc:    pop     r15
0x19ede:    ret

```

```

0x19edf:      nop
0x19ee0:      cmp     byte ptr [r13], r12b
0x19ee4:      jne     0x19efb
0x19ee6:      mov     rdx, r14
0x19ee9:      mov     rsi, r13
0x19eec:      mov     rdi, rbp
0x19eef:      call    0x4890
0x19ef4:      mov     r15d, eax
0x19ef7:      test    eax, eax
0x19ef9:      je      0x19ec7
0x19efb:      mov     eax, dword ptr [rsp + 8]
0x19eff:      test    eax, eax
0x19f01:      js      0x19f90
0x19f07:      mov     edx, dword ptr [rsp + 8]
0x19f0b:      add     ebx, 1
0x19f0e:      test    edx, edx
0x19f10:      jns     0x19e4b
0x19f16:      mov     rcx, qword ptr [rsp + 0x10]
0x19f1b:      movsxd  rax, ebx
0x19f1e:      mov     rax, qword ptr [rcx + rax*8]
0x19f22:      mov     r13, qword ptr [rax]
0x19f25:      mov     rdi, r13
0x19f28:      call    0x49b0
0x19f2d:      mov     r8, rax
0x19f30:      cmp     dword ptr [rsp + 0xc], eax
0x19f34:      jg      0x19f90
0x19f36:      cmp     dword ptr [rsp + 0x18], 1
0x19f3b:      movzx   r12d, byte ptr [rbp]
0x19f40:      je      0x19ee0
0x19f42:      mov     r15d, r8d
0x19f45:      sub     r15d, dword ptr [rsp + 0xc]
0x19f4a:      movsxd  rcx, r15d
0x19f4d:      add     r13, rcx
0x19f50:      jmp     0x19f66
0x19f52:      nop     word ptr [rax + rax]
0x19f58:      sub     r15d, 1
0x19f5c:      sub     r13, 1
0x19f60:      cmp     r15d, -1
0x19f64:      je      0x19f90
0x19f66:      cmp     byte ptr [r13], r12b
0x19f6a:      jne     0x19f58
0x19f6c:      mov     rdx, r14
0x19f6f:      mov     rsi, r13
0x19f72:      mov     rdi, rbp
0x19f75:      call    0x4890
0x19f7a:      test    eax, eax
0x19f7c:      je      0x19ec7
0x19f82:      sub     r15d, 1

```

```

0x19f86:      sub     r13, 1
0x19f8a:      cmp     r15d, -1
0x19f8e:      jne     0x19f66
0x19f90:      sub     ebx, 1
0x19f93:      test    dword ptr [rsp + 8], ebx
0x19f97:      jns     0x19e3f
0x19f9d:      mov     r15d, 0xffffffff
0x19fa3:      jmp     0x19ecd
0x19fa8:      nop     dword ptr [rax + rax]
0x19fb0:      test    esi, esi
0x19fb2:      jns     0x19f9d
0x19fb4:      mov     eax, dword ptr [rsp + 0x1c]
0x19fb8:      lea     ebx, [rax - 1]
0x19fbb:      mov     rdi, rbp
0x19fbe:      call    0x49b0
0x19fc3:      mov     dword ptr [rsp + 0xc], eax
0x19fc7:      test    dword ptr [rsp + 8], ebx
0x19fcb:      jns     0x19e2e
0x19fd1:      jmp     0x19f9d
0x19fd3:      nop     word ptr cs:[rax + rax]
0x19fde:      nop
0x19fe0:      push    r14
0x19fe2:      push    r13
0x19fe4:      movsxd  r13, esi
0x19fe7:      sub     edx, r13d
0x19fea:      push    r12
0x19fec:      push    rbp
0x19fed:      mov     rbp, rdi
0x19ff0:      push    rbx
0x19ff1:      movsxd  rbx, edx
0x19ff4:      cmp     ecx, 2
0x19ff7:      je      0x1a070
0x19ff9:      mov     eax, 0x1d
0x19ffe:      lea     r14, [rip + 0x843a]
0x1a005:      cmp     ecx, 3
0x1a008:      je      0x1a028
0x1a00a:      xor     eax, eax
0x1a00c:      lea     r14, [rip + 0x83f5]
0x1a013:      lea     rdx, [rip + 0x83fe]
0x1a01a:      cmp     ecx, 1
0x1a01d:      sete    al
0x1a020:      cmove   r14, rdx
0x1a024:      lea     eax, [rax + rax*2 + 0xf]
0x1a028:      lea     edi, [rax + rbx + 3]
0x1a02c:      movsxd  rdi, edi
0x1a02f:      call    0x1d670
0x1a034:      lea     rsi, [rbp + r13]
0x1a039:      mov     rdx, rbx

```

```

0x1a03c:    mov     r12, rax
0x1a03f:    mov     rdi, rax
0x1a042:    call    0x4880
0x1a047:    mov     byte ptr [r12 + rbx], 0x3a
0x1a04c:    lea     rdi, [r12 + rbx + 2]
0x1a051:    mov     rsi, r14
0x1a054:    mov     byte ptr [r12 + rbx + 1], 0x20
0x1a05a:    call    0x48b0
0x1a05f:    pop     rbx
0x1a060:    mov     rax, r12
0x1a063:    pop     rbp
0x1a064:    pop     r12
0x1a066:    pop     r13
0x1a068:    pop     r14
0x1a06a:    ret
0x1a06b:    nop     dword ptr [rax + rax]
0x1a070:    mov     eax, 0x13
0x1a075:    lea     r14, [rip + 0x83af]
0x1a07c:    jmp     0x1a028
0x1a07e:    nop
0x1a080:    push    r12
0x1a082:    push    rbp
0x1a083:    push    rbx
0x1a084:    test    rdi, rdi
0x1a087:    je      0x1a0b0
0x1a089:    mov     rbp, rdi
0x1a08c:    call    0x49b0
0x1a091:    lea     rdi, [rax + 1]
0x1a095:    call    0x1d670
0x1a09a:    pop     rbx
0x1a09b:    mov     rsi, rbp
0x1a09e:    pop     rbp
0x1a09f:    mov     rdi, rax
0x1a0a2:    pop     r12
0x1a0a4:    jmp     0x48b0
0x1a0a9:    nop     dword ptr [rax]
0x1a0b0:    lea     rdi, [rip + 0x64a9]
0x1a0b7:    call    0x47e0
0x1a0bc:    mov     r12, rax
0x1a0bf:    test    rax, rax
0x1a0c2:    je      0x1a110
0x1a0c4:    mov     rdi, rax
0x1a0c7:    call    0x49b0
0x1a0cc:    lea     edi, [rax + 0xa]
0x1a0cf:    movsxd  rbp, eax
0x1a0d2:    movsxd  rdi, edi
0x1a0d5:    lea     rbx, [rbp + 1]
0x1a0d9:    call    0x1d670

```

```

0x1a0de:    mov     rsi, r12
0x1a0e1:    mov     rdi, rax
0x1a0e4:    call    0x48b0
0x1a0e9:    movabs  rdx, 0x79726f747369682e
0x1a0f3:    mov     byte ptr [rax + rbp], 0x2f
0x1a0f7:    mov     r8, rax
0x1a0fa:    lea     rax, [rax + rbx]
0x1a0fe:    mov     qword ptr [rax], rdx
0x1a101:    mov     byte ptr [rax + 8], 0
0x1a105:    mov     rax, r8
0x1a108:    pop     rbx
0x1a109:    pop     rbp
0x1a10a:    pop     r12
0x1a10c:    ret
0x1a10d:    nop     dword ptr [rax]
0x1a110:    mov     ebx, 2
0x1a115:    mov     ebp, 1
0x1a11a:    mov     edi, 0xb
0x1a11f:    lea     r12, [rip + 0x75b8]
0x1a126:    jmp     0x1a0d9
0x1a128:    nop     dword ptr [rax + rax]
0x1a130:    push    r15
0x1a132:    xor     eax, eax
0x1a134:    push    r14
0x1a136:    mov     r14d, esi
0x1a139:    push    r13
0x1a13b:    push    r12
0x1a13d:    push    rbp
0x1a13e:    push    rbx
0x1a13f:    mov     ebx, edx
0x1a141:    sub     rsp, 0x18
0x1a145:    call    0x1a080
0x1a14a:    cmp     ebx, 1
0x1a14d:    mov     edx, 0x1b6
0x1a152:    sbb     esi, esi
0x1a154:    mov     r13, rax
0x1a157:    mov     rdi, rax
0x1a15a:    xor     eax, eax
0x1a15c:    and     esi, 0x1c0
0x1a162:    add     esi, 0x241
0x1a168:    call    0x4d80
0x1a16d:    cmp     eax, -1
0x1a170:    je      0x1a270
0x1a176:    mov     ebp, eax
0x1a178:    mov     eax, dword ptr [rip + 0x11966]
0x1a17e:    cmp     eax, r14d
0x1a181:    mov     edx, eax
0x1a183:    cmovle  r14d, eax

```

```

0x1a187:    sub     edx, r14d
0x1a18a:    cmp     eax, edx
0x1a18c:    jle     0x1a290
0x1a192:    sub     eax, 1
0x1a195:    mov     rcx, qword ptr [rip + 0x11964]
0x1a19c:    movsxd  rsi, edx
0x1a19f:    xor     r12d, r12d
0x1a1a2:    sub     eax, edx
0x1a1a4:    add     rax, rsi
0x1a1a7:    lea     rbx, [rcx + rsi*8]
0x1a1ab:    lea     r15, [rcx + rax*8 + 8]
0x1a1b0:    mov     rax, qword ptr [rbx]
0x1a1b3:    add     rbx, 8
0x1a1b7:    mov     rdi, qword ptr [rax]
0x1a1ba:    call    0x49b0
0x1a1bf:    lea     r12d, [r12 + rax + 1]
0x1a1c4:    cmp     r15, rbx
0x1a1c7:    jne     0x1a1b0
0x1a1c9:    movsxd  rax, r12d
0x1a1cc:    mov     qword ptr [rsp + 8], rax
0x1a1d1:    mov     rdi, qword ptr [rsp + 8]
0x1a1d6:    call    0x1d670
0x1a1db:    mov     r15, rax
0x1a1de:    mov     eax, dword ptr [rip + 0x11900]
0x1a1e4:    mov     ebx, eax
0x1a1e6:    sub     ebx, r14d
0x1a1e9:    cmp     eax, ebx
0x1a1eb:    jle     0x1a238
0x1a1ed:    mov     r12, qword ptr [rip + 0x1190c]
0x1a1f4:    movsxd  rbx, ebx
0x1a1f7:    xor     r14d, r14d
0x1a1fa:    nop     word ptr [rax + rax]
0x1a200:    mov     rax, qword ptr [r12 + rbx*8]
0x1a204:    movsxd  rdi, r14d
0x1a207:    add     rdi, r15
0x1a20a:    mov     rsi, qword ptr [rax]
0x1a20d:    call    0x48b0
0x1a212:    mov     rax, qword ptr [r12 + rbx*8]
0x1a216:    add     rbx, 1
0x1a21a:    mov     rdi, qword ptr [rax]
0x1a21d:    call    0x49b0
0x1a222:    add     eax, r14d
0x1a225:    lea     r14d, [rax + 1]
0x1a229:    cdqe
0x1a22b:    mov     byte ptr [r15 + rax], 0xa
0x1a230:    cmp     dword ptr [rip + 0x118ae], ebx
0x1a236:    jg      0x1a200
0x1a238:    mov     rdx, qword ptr [rsp + 8]

```



```

0x1a23d:    mov     rsi, r15
0x1a240:    mov     edi, ebp
0x1a242:    call    0x4790
0x1a247:    mov     rdi, r15
0x1a24a:    call    0x18d10
0x1a24f:    mov     edi, ebp
0x1a251:    call    0x4ab0
0x1a256:    mov     rdi, r13
0x1a259:    call    0x18d10
0x1a25e:    add     rsp, 0x18
0x1a262:    xor     eax, eax
0x1a264:    pop     rbx
0x1a265:    pop     rbp
0x1a266:    pop     r12
0x1a268:    pop     r13
0x1a26a:    pop     r14
0x1a26c:    pop     r15
0x1a26e:    ret
0x1a26f:    nop
0x1a270:    mov     rdi, r13
0x1a273:    call    0x18d10
0x1a278:    call    0x4860
0x1a27d:    mov     eax, dword ptr [rax]
0x1a27f:    add     rsp, 0x18
0x1a283:    pop     rbx
0x1a284:    pop     rbp
0x1a285:    pop     r12
0x1a287:    pop     r13
0x1a289:    pop     r14
0x1a28b:    pop     r15
0x1a28d:    ret
0x1a28e:    nop
0x1a290:    mov     qword ptr [rsp + 8], 0
0x1a299:    jmp     0x1a1d1
0x1a29e:    nop
0x1a2a0:    endbr64
0x1a2a4:    sub     rsp, 8
0x1a2a8:    mov     edi, 0x18
0x1a2ad:    call    0x1d670
0x1a2b2:    mov     rdx, qword ptr [rip + 0x11847]
0x1a2b9:    mov     qword ptr [rax], rdx
0x1a2bc:    mov     edx, dword ptr [rip + 0x1183a]
0x1a2c2:    mov     dword ptr [rax + 8], edx
0x1a2c5:    mov     edx, dword ptr [rip + 0x11819]
0x1a2cb:    mov     dword ptr [rax + 0xc], edx
0x1a2ce:    mov     edx, dword ptr [rip + 0x11824]
0x1a2d4:    mov     dword ptr [rax + 0x10], edx
0x1a2d7:    add     rsp, 8

```

```

0x1a2db:    ret
0x1a2dc:    nop      dword ptr [rax]
0x1a2e0:    endbr64
0x1a2e4:    mov     rax, qword ptr [rdi]
0x1a2e7:    mov     qword ptr [rip + 0x11812], rax
0x1a2ee:    mov     eax, dword ptr [rdi + 8]
0x1a2f1:    mov     dword ptr [rip + 0x11805], eax
0x1a2f7:    mov     eax, dword ptr [rdi + 0xc]
0x1a2fa:    mov     dword ptr [rip + 0x117e4], eax
0x1a300:    mov     eax, dword ptr [rdi + 0x10]
0x1a303:    mov     dword ptr [rip + 0x117ef], eax
0x1a309:    ret
0x1a30a:    nop      word ptr [rax + rax]
0x1a310:    endbr64
0x1a314:    mov     eax, dword ptr [rip + 0x117ca]
0x1a31a:    mov     dword ptr [rip + 0x117dc], eax
0x1a320:    ret
0x1a321:    nop      word ptr cs:[rax + rax]
0x1a32c:    nop      dword ptr [rax]
0x1a330:    endbr64
0x1a334:    push    r12
0x1a336:    xor     r12d, r12d
0x1a339:    push    rbx
0x1a33a:    sub     rsp, 8
0x1a33e:    mov     rbx, qword ptr [rip + 0x117bb]
0x1a345:    test    rbx, rbx
0x1a348:    jne     0x1a35b
0x1a34a:    jmp     0x1a367
0x1a34c:    nop      dword ptr [rax]
0x1a350:    mov     rdi, qword ptr [rax]
0x1a353:    call    0x49b0
0x1a358:    add     r12d, eax
0x1a35b:    mov     rax, qword ptr [rbx]
0x1a35e:    add     rbx, 8
0x1a362:    test    rax, rax
0x1a365:    jne     0x1a350
0x1a367:    add     rsp, 8
0x1a36b:    mov     eax, r12d
0x1a36e:    pop     rbx
0x1a36f:    pop     r12
0x1a371:    ret
0x1a372:    nop      word ptr cs:[rax + rax]
0x1a37d:    nop      dword ptr [rax]
0x1a380:    endbr64
0x1a384:    push    rbp
0x1a385:    mov     rbp, rdi
0x1a388:    push    rbx
0x1a389:    sub     rsp, 8

```

```

0x1a38d:    mov     eax, dword ptr [rip + 0x11759]
0x1a393:    test    eax, eax
0x1a395:    je      0x1a3aa
0x1a397:    movsxd  rdx, dword ptr [rip + 0x11746]
0x1a39e:    cmp     edx, dword ptr [rip + 0x11744]
0x1a3a4:    je      0x1a458
0x1a3aa:    mov     eax, dword ptr [rip + 0x11748]
0x1a3b0:    test    eax, eax
0x1a3b2:    je      0x1a428
0x1a3b4:    mov     edx, dword ptr [rip + 0x1172a]
0x1a3ba:    lea     ecx, [rax - 1]
0x1a3bd:    cmp     ecx, edx
0x1a3bf:    je      0x1a4b0
0x1a3c5:    add     edx, 1
0x1a3c8:    mov     dword ptr [rip + 0x11716], edx
0x1a3ce:    mov     edi, 0x10
0x1a3d3:    call    0x1d670
0x1a3d8:    mov     rdi, rbp
0x1a3db:    mov     rbx, rax
0x1a3de:    call    0x49b0
0x1a3e3:    lea     rdi, [rax + 1]
0x1a3e7:    call    0x1d670
0x1a3ec:    mov     rsi, rbp
0x1a3ef:    mov     rdi, rax
0x1a3f2:    call    0x48b0
0x1a3f7:    movsxd  rdx, dword ptr [rip + 0x116e6]
0x1a3fe:    mov     qword ptr [rbx + 8], 0
0x1a406:    mov     qword ptr [rbx], rax
0x1a409:    mov     rax, qword ptr [rip + 0x116f0]
0x1a410:    mov     qword ptr [rax + rdx*8], 0
0x1a418:    mov     qword ptr [rax + rdx*8 - 8], rbx
0x1a41d:    add     rsp, 8
0x1a421:    pop     rbx
0x1a422:    pop     rbp
0x1a423:    ret
0x1a424:    nop     dword ptr [rax]
0x1a428:    mov     dword ptr [rip + 0x116c6], 0x32
0x1a432:    mov     edi, 0x190
0x1a437:    call    0x1d670
0x1a43c:    mov     dword ptr [rip + 0x1169e], 1
0x1a446:    mov     qword ptr [rip + 0x116b3], rax
0x1a44d:    jmp     0x1a3ce
0x1a452:    nop     word ptr [rax + rax]
0x1a458:    test    edx, edx
0x1a45a:    je      0x1a41d
0x1a45c:    mov     rax, qword ptr [rip + 0x1169d]
0x1a463:    mov     rax, qword ptr [rax]
0x1a466:    test    rax, rax

```

```

0x1a469:      je      0x1a489
0x1a46b:      mov     rdi, qword ptr [rax]
0x1a46e:      call    0x18d10
0x1a473:      mov     rax, qword ptr [rip + 0x11686]
0x1a47a:      mov     rdi, qword ptr [rax]
0x1a47d:      call    0x18d10
0x1a482:      movsxd   rdx, dword ptr [rip + 0x1165b]
0x1a489:      test    edx, edx
0x1a48b:      jle      0x1a4a1
0x1a48d:      mov     rdi, qword ptr [rip + 0x1166c]
0x1a494:      shl     rdx, 3
0x1a498:      lea     rsi, [rdi + 8]
0x1a49c:      call    0x4d40
0x1a4a1:      add     dword ptr [rip + 0x10340], 1
0x1a4a8:      jmp     0x1a3ce
0x1a4ad:      nop     dword ptr [rax]
0x1a4b0:      add     eax, 0x32
0x1a4b3:      mov     rdi, qword ptr [rip + 0x11646]
0x1a4ba:      mov     dword ptr [rip + 0x11638], eax
0x1a4c0:      cdqe
0x1a4c2:      lea     rsi, [rax*8]
0x1a4ca:      call    0x1d690
0x1a4cf:      mov     edx, dword ptr [rip + 0x1160f]
0x1a4d5:      mov     qword ptr [rip + 0x11624], rax
0x1a4dc:      jmp     0x1a3c5
0x1a4e1:      nop     word ptr cs:[rax + rax]
0x1a4ec:      nop     dword ptr [rax]
0x1a4f0:      endbr64
0x1a4f4:      push    r14
0x1a4f6:      xor     r14d, r14d
0x1a4f9:      push    r13
0x1a4fb:      mov     r13, rdx
0x1a4fe:      push    r12
0x1a500:      mov     r12, rsi
0x1a503:      push    rbp
0x1a504:      push    rbx
0x1a505:      movsxd   rbx, edi
0x1a508:      mov     edi, 0x10
0x1a50d:      call    0x1d670
0x1a512:      cmp     dword ptr [rip + 0x115cc], ebx
0x1a518:      jle      0x1a557
0x1a51a:      mov     rbp, rax
0x1a51d:      mov     rax, qword ptr [rip + 0x115dc]
0x1a524:      mov     rdi, r12
0x1a527:      mov     r14, qword ptr [rax + rbx*8]
0x1a52b:      call    0x49b0
0x1a530:      lea     rdi, [rax + 1]
0x1a534:      call    0x1d670

```

```

0x1a539:      mov     rsi, r12
0x1a53c:      mov     rdi, rax
0x1a53f:      call    0x48b0
0x1a544:      mov     qword ptr [rbp + 8], r13
0x1a548:      mov     qword ptr [rbp], rax
0x1a54c:      mov     rax, qword ptr [rip + 0x115ad]
0x1a553:      mov     qword ptr [rax + rbx*8], rbp
0x1a557:      pop     rbx
0x1a558:      mov     rax, r14
0x1a55b:      pop     rbp
0x1a55c:      pop     r12
0x1a55e:      pop     r13
0x1a560:      pop     r14
0x1a562:      ret
0x1a563:      nop     word ptr cs:[rax + rax]
0x1a56e:      nop
0x1a570:      endbr64
0x1a574:      mov     eax, dword ptr [rip + 0x11582]
0x1a57a:      ret
0x1a57b:      nop     dword ptr [rax + rax]
0x1a580:      endbr64
0x1a584:      xor     edx, edx
0x1a586:      xor     eax, eax
0x1a588:      jmp     0x19de0
0x1a58d:      nop     dword ptr [rax]
0x1a590:      endbr64
0x1a594:      mov     edx, 1
0x1a599:      xor     eax, eax
0x1a59b:      jmp     0x19de0
0x1a5a0:      endbr64
0x1a5a4:      push    r12
0x1a5a6:      push    rbx
0x1a5a7:      sub     rsp, 8
0x1a5ab:      mov     eax, dword ptr [rip + 0x11533]
0x1a5b1:      cmp     eax, edi
0x1a5b3:      jle     0x1a610
0x1a5b5:      test    eax, eax
0x1a5b7:      je      0x1a610
0x1a5b9:      lea     ebx, [rax - 1]
0x1a5bc:      movsxd  rdx, edi
0x1a5bf:      mov     rcx, qword ptr [rip + 0x1153a]
0x1a5c6:      mov     esi, ebx
0x1a5c8:      shl     rdx, 3
0x1a5cc:      sub     esi, edi
0x1a5ce:      cmp     eax, edi
0x1a5d0:      lea     r9, [rcx + rdx]
0x1a5d4:      mov     eax, 8
0x1a5d9:      lea     r8, [rsi*8 + 8]

```

```

0x1a5e1:    mov     r12, qword ptr [r9]
0x1a5e4:    mov     rdi, r9
0x1a5e7:    lea     rsi, [rcx + rdx + 8]
0x1a5ec:    cmovle  r8, rax
0x1a5f0:    mov     rdx, r8
0x1a5f3:    call    0x4d40
0x1a5f8:    mov     dword ptr [rip + 0x114e6], ebx
0x1a5fe:    add     rsp, 8
0x1a602:    mov     rax, r12
0x1a605:    pop     rbx
0x1a606:    pop     r12
0x1a608:    ret
0x1a609:    nop     dword ptr [rax]
0x1a610:    xor     r12d, r12d
0x1a613:    add     rsp, 8
0x1a617:    mov     rax, r12
0x1a61a:    pop     rbx
0x1a61b:    pop     r12
0x1a61d:    ret
0x1a61e:    nop
0x1a620:    endbr64
0x1a624:    push    r12
0x1a626:    mov     eax, dword ptr [rip + 0x114b8]
0x1a62c:    push    rbp
0x1a62d:    xor     ebp, ebp
0x1a62f:    test    edi, edi
0x1a631:    cmovns  ebp, edi
0x1a634:    push    rbx
0x1a635:    cmp     eax, ebp
0x1a637:    jle     0x1a6ce
0x1a63d:    sub     eax, ebp
0x1a63f:    mov     r12d, edi
0x1a642:    test    eax, eax
0x1a644:    jle     0x1a6e8
0x1a64a:    xor     ebx, ebx
0x1a64c:    nop     dword ptr [rax]
0x1a650:    mov     rax, qword ptr [rip + 0x114a9]
0x1a657:    mov     rax, qword ptr [rax + rbx*8]
0x1a65b:    mov     rdi, qword ptr [rax]
0x1a65e:    call    0x18d10
0x1a663:    mov     rax, qword ptr [rip + 0x11496]
0x1a66a:    mov     rdi, qword ptr [rax + rbx*8]
0x1a66e:    call    0x18d10
0x1a673:    mov     eax, dword ptr [rip + 0x1146b]
0x1a679:    lea     edx, [rbx + 1]
0x1a67c:    add     rbx, 1
0x1a680:    sub     eax, ebp
0x1a682:    cmp     eax, ebx

```

```

0x1a684:      jg      0x1a650
0x1a686:      mov     dword ptr [rip + 0x1015c], edx
0x1a68c:      mov     rcx, qword ptr [rip + 0x1146d]
0x1a693:      test    r12d, r12d
0x1a696:      jle     0x1a6f0
0x1a698:      cdqe
0x1a69a:      lea     rsi, [rcx + rax*8]
0x1a69e:      xor     eax, eax
0x1a6a0:      mov     rdx, qword ptr [rsi + rax*8]
0x1a6a4:      mov     qword ptr [rcx + rax*8], rdx
0x1a6a8:      add     rax, 1
0x1a6ac:      cmp     ebp, eax
0x1a6ae:      jg      0x1a6a0
0x1a6b0:      test    ebp, ebp
0x1a6b2:      mov     eax, 1
0x1a6b7:      cmovg   eax, ebp
0x1a6ba:      movsxd   rdx, eax
0x1a6bd:      lea     rcx, [rcx + rdx*8]
0x1a6c1:      mov     qword ptr [rcx], 0
0x1a6c8:      mov     dword ptr [rip + 0x11416], eax
0x1a6ce:      pop     rbx
0x1a6cf:      mov     dword ptr [rip + 0x11413], ebp
0x1a6d5:      mov     dword ptr [rip + 0x1140d], 1
0x1a6df:      pop     rbp
0x1a6e0:      pop     r12
0x1a6e2:      ret
0x1a6e3:      nop     dword ptr [rax + rax]
0x1a6e8:      xor     edx, edx
0x1a6ea:      jmp     0x1a686
0x1a6ec:      nop     dword ptr [rax]
0x1a6f0:      xor     eax, eax
0x1a6f2:      jmp     0x1a6c1
0x1a6f4:      nop     word ptr cs:[rax + rax]
0x1a6ff:      nop
0x1a700:      endbr64
0x1a704:      mov     edx, dword ptr [rip + 0x113e2]
0x1a70a:      mov     eax, dword ptr [rip + 0x113d8]
0x1a710:      test    edx, edx
0x1a712:      je      0x1a720
0x1a714:      mov     dword ptr [rip + 0x113ce], 0
0x1a71e:      neg     eax
0x1a720:      ret
0x1a721:      nop     word ptr cs:[rax + rax]
0x1a72c:      nop     dword ptr [rax]
0x1a730:      endbr64
0x1a734:      push    r15
0x1a736:      push    r14
0x1a738:      mov     r14d, esi

```

```

0x1a73b:    push    r13
0x1a73d:    push    r12
0x1a73f:    mov     r12d, edx
0x1a742:    push    rbp
0x1a743:    push    rbx
0x1a744:    sub     rsp, 0xa8
0x1a74b:    mov     rax, qword ptr fs:[0x28]
0x1a754:    mov     qword ptr [rsp + 0x98], rax
0x1a75c:    xor     eax, eax
0x1a75e:    call    0x1a080
0x1a763:    xor     esi, esi
0x1a765:    mov     edx, 0x1b6
0x1a76a:    mov     rbp, rax
0x1a76d:    mov     rdi, rax
0x1a770:    xor     eax, eax
0x1a772:    call    0x4d80
0x1a777:    test    eax, eax
0x1a779:    js      0x1a80f
0x1a77f:    mov     rsi, rsp
0x1a782:    mov     edi, eax
0x1a784:    mov     r15d, eax
0x1a787:    call    0x18ef0
0x1a78c:    cmp     eax, -1
0x1a78f:    je      0x1a807
0x1a791:    mov     eax, dword ptr [rsp + 0x30]
0x1a795:    lea     edi, [rax + 1]
0x1a798:    movsxd  rdi, edi
0x1a79b:    call    0x1d670
0x1a7a0:    mov     rdx, qword ptr [rsp + 0x30]
0x1a7a5:    mov     edi, r15d
0x1a7a8:    mov     rsi, rax
0x1a7ab:    mov     r13, rax
0x1a7ae:    call    0x47a0
0x1a7b3:    cmp     qword ptr [rsp + 0x30], rax
0x1a7b8:    je      0x1a819
0x1a7ba:    mov     edi, r15d
0x1a7bd:    call    0x4ab0
0x1a7c2:    mov     rdi, rbp
0x1a7c5:    call    0x18d10
0x1a7ca:    test    r13, r13
0x1a7cd:    je      0x1a7d7
0x1a7cf:    mov     rdi, r13
0x1a7d2:    call    0x18d10
0x1a7d7:    call    0x4860
0x1a7dc:    mov     eax, dword ptr [rax]
0x1a7de:    mov     rdx, qword ptr [rsp + 0x98]
0x1a7e6:    sub     rdx, qword ptr fs:[0x28]
0x1a7ef:    jne     0x1a8f0

```



```

0x1a7f5:      add     rsp, 0xa8
0x1a7fc:      pop     rbx
0x1a7fd:      pop     rbp
0x1a7fe:      pop     r12
0x1a800:      pop     r13
0x1a802:      pop     r14
0x1a804:      pop     r15
0x1a806:      ret
0x1a807:      mov     edi, r15d
0x1a80a:      call    0x4ab0
0x1a80f:      mov     rdi, rbp
0x1a812:      call    0x18d10
0x1a817:      jmp     0x1a7d7
0x1a819:      mov     edi, r15d
0x1a81c:      call    0x4ab0
0x1a821:      mov     rcx, qword ptr [rsp + 0x30]
0x1a826:      test    r12d, r12d
0x1a829:      jns     0x1a82e
0x1a82b:      mov     r12d, ecx
0x1a82e:      xor     ebx, ebx
0x1a830:      xor     esi, esi
0x1a832:      movsxd  rdx, esi
0x1a835:      cmp     ebx, r14d
0x1a838:      jge     0x1a885
0x1a83a:      cmp     rdx, rcx
0x1a83d:      jge     0x1a885
0x1a83f:      lea     eax, [rsi + 1]
0x1a842:      cdqe
0x1a844:      cmp     rdx, rcx
0x1a847:      jl      0x1a85c
0x1a849:      jmp     0x1a87b
0x1a84b:      nop     dword ptr [rax + rax]
0x1a850:      lea     rdx, [rax + 1]
0x1a854:      cmp     rax, rcx
0x1a857:      je      0x1a878
0x1a859:      mov     rax, rdx
0x1a85c:      cmp     byte ptr [r13 + rax - 1], 0xa
0x1a862:      jne     0x1a850
0x1a864:      add     ebx, 1
0x1a867:      movsxd  rsi, eax
0x1a86a:      cmp     r14d, ebx
0x1a86d:      je      0x1a832
0x1a86f:      lea     rdx, [rax + 1]
0x1a873:      cmp     rax, rcx
0x1a876:      jne     0x1a859
0x1a878:      movsxd  rdx, esi
0x1a87b:      cmp     rcx, rdx
0x1a87e:      jle     0x1a885

```

```

0x1a880:    cmp     r14d, ebx
0x1a883:    jg      0x1a83f
0x1a885:    lea     r14, [r13 + rdx]
0x1a88a:    lea     r15d, [rsi + 1]
0x1a88e:    cmp     rcx, rdx
0x1a891:    jg      0x1a8aa
0x1a893:    jmp     0x1a8d4
0x1a895:    nop     dword ptr [rax]
0x1a898:    movsxd  rax, r15d
0x1a89b:    add     r14, 1
0x1a89f:    add     r15d, 1
0x1a8a3:    cmp     rax, qword ptr [rsp + 0x30]
0x1a8a8:    jge     0x1a8d4
0x1a8aa:    cmp     byte ptr [r14], 0xa
0x1a8ae:    jne     0x1a898
0x1a8b0:    lea     rdi, [r13 + rsi]
0x1a8b5:    mov     byte ptr [r14], 0
0x1a8b9:    cmp     byte ptr [rdi], 0
0x1a8bc:    jne     0x1a8cb
0x1a8be:    add     ebx, 1
0x1a8c1:    cmp     r12d, ebx
0x1a8c4:    jle     0x1a8d4
0x1a8c6:    movsxd  rsi, r15d
0x1a8c9:    jmp     0x1a898
0x1a8cb:    xor     eax, eax
0x1a8cd:    call    0x1a380
0x1a8d2:    jmp     0x1a8be
0x1a8d4:    mov     rdi, rbp
0x1a8d7:    call    0x18d10
0x1a8dc:    test    r13, r13
0x1a8df:    je      0x1a8e9
0x1a8e1:    mov     rdi, r13
0x1a8e4:    call    0x18d10
0x1a8e9:    xor     eax, eax
0x1a8eb:    jmp     0x1a7de
0x1a8f0:    call    0x49d0
0x1a8f5:    nop     word ptr cs:[rax + rax]
0x1a900:    endbr64
0x1a904:    mov     edx, 0xffffffff
0x1a909:    xor     esi, esi
0x1a90b:    xor     eax, eax
0x1a90d:    jmp     0x1a730
0x1a912:    nop     word ptr cs:[rax + rax]
0x1a91d:    nop     dword ptr [rax]
0x1a920:    endbr64
0x1a924:    push    r14
0x1a926:    push    r13
0x1a928:    push    r12

```

```

0x1a92a:    mov     r12d, esi
0x1a92d:    push   rbp
0x1a92e:    push   rbx
0x1a92f:    sub     rsp, 0xa0
0x1a936:    mov     rax, qword ptr fs:[0x28]
0x1a93f:    mov     qword ptr [rsp + 0x98], rax
0x1a947:    xor     eax, eax
0x1a949:    call    0x1a080
0x1a94e:    xor     esi, esi
0x1a950:    mov     edx, 0x1b6
0x1a955:    mov     rbp, rax
0x1a958:    mov     rdi, rax
0x1a95b:    xor     eax, eax
0x1a95d:    call    0x4d80
0x1a962:    cmp     eax, -1
0x1a965:    je      0x1aa58
0x1a96b:    mov     rsi, rsp
0x1a96e:    mov     edi, eax
0x1a970:    mov     r14d, eax
0x1a973:    call    0x18ef0
0x1a978:    cmp     eax, -1
0x1a97b:    je      0x1aa58
0x1a981:    mov     eax, dword ptr [rsp + 0x30]
0x1a985:    lea     edi, [rax + 1]
0x1a988:    movsxd  rdi, edi
0x1a98b:    call    0x1d670
0x1a990:    mov     rdx, qword ptr [rsp + 0x30]
0x1a995:    mov     edi, r14d
0x1a998:    mov     rsi, rax
0x1a99b:    mov     r13, rax
0x1a99e:    call    0x47a0
0x1a9a3:    mov     edi, r14d
0x1a9a6:    mov     rbx, rax
0x1a9a9:    call    0x4ab0
0x1a9ae:    test    ebx, ebx
0x1a9b0:    jle     0x1aa88
0x1a9b6:    lea     eax, [rbx - 1]
0x1a9b9:    cmp     ebx, 1
0x1a9bc:    je      0x1a9f3
0x1a9be:    test    r12d, r12d
0x1a9c1:    je      0x1a9f3
0x1a9c3:    lea     edx, [rbx - 2]
0x1a9c6:    movsxd  rdx, edx
0x1a9c9:    nop     dword ptr [rax]
0x1a9d0:    cmp     byte ptr [r13 + rdx + 1], 0xa
0x1a9d6:    jne     0x1a9dc
0x1a9d8:    sub     r12d, 1
0x1a9dc:    test    r12d, r12d

```

```

0x1a9df:    mov     eax, edx
0x1a9e1:    setne   sil
0x1a9e5:    test    edx, edx
0x1a9e7:    setne   cl
0x1a9ea:    sub     rdx, 1
0x1a9ee:    test    sil, cl
0x1a9f1:    jne     0x1a9d0
0x1a9f3:    test    eax, eax
0x1a9f5:    je      0x1aa88
0x1a9fb:    cdqe
0x1a9fd:    jmp     0x1aa08
0x1a9ff:    nop
0x1aa00:    sub     rax, 1
0x1aa04:    test    eax, eax
0x1aa06:    je      0x1aa50
0x1aa08:    cmp     byte ptr [r13 + rax], 0xa
0x1aa0e:    mov     ebx, eax
0x1aa10:    jne     0x1aa00
0x1aa12:    mov     edx, 0x1b6
0x1aa17:    mov     esi, 0x201
0x1aa1c:    mov     rdi, rbp
0x1aa1f:    xor     eax, eax
0x1aa21:    call    0x4d80
0x1aa26:    mov     r12d, eax
0x1aa29:    cmp     eax, -1
0x1aa2c:    je      0x1aa50
0x1aa2e:    mov     rdx, qword ptr [rsp + 0x30]
0x1aa33:    lea     esi, [rbx + 1]
0x1aa36:    mov     edi, eax
0x1aa38:    movsxd  rsi, esi
0x1aa3b:    sub     rdx, rsi
0x1aa3e:    add     rsi, r13
0x1aa41:    call    0x4790
0x1aa46:    mov     edi, r12d
0x1aa49:    call    0x4ab0
0x1aa4e:    nop
0x1aa50:    mov     rdi, r13
0x1aa53:    call    0x18d10
0x1aa58:    mov     rdi, rbp
0x1aa5b:    call    0x18d10
0x1aa60:    mov     rax, qword ptr [rsp + 0x98]
0x1aa68:    sub     rax, qword ptr fs:[0x28]
0x1aa71:    jne     0x1aa8f
0x1aa73:    add     rsp, 0xa0
0x1aa7a:    xor     eax, eax
0x1aa7c:    pop     rbx
0x1aa7d:    pop     rbp
0x1aa7e:    pop     r12

```

```

0x1aa80:    pop     r13
0x1aa82:    pop     r14
0x1aa84:    ret
0x1aa85:    nop     dword ptr [rax]
0x1aa88:    test    r13, r13
0x1aa8b:    je      0x1aa58
0x1aa8d:    jmp     0x1aa50
0x1aa8f:    call    0x49d0
0x1aa94:    nop     word ptr cs:[rax + rax]
0x1aa9f:    nop
0x1aaa0:    endbr64
0x1aaa4:    mov     r8d, edi
0x1aaa7:    xor     edx, edx
0x1aaa9:    mov     rdi, rsi
0x1aaac:    xor     eax, eax
0x1aaae:    mov     esi, r8d
0x1aab1:    jmp     0x1a130
0x1aab6:    nop     word ptr cs:[rax + rax]
0x1aac0:    endbr64
0x1aac4:    mov     esi, dword ptr [rip + 0x1101a]
0x1aaca:    mov     edx, 1
0x1aacf:    xor     eax, eax
0x1aad1:    jmp     0x1a130
0x1aad6:    nop     word ptr cs:[rax + rax]
0x1aae0:    endbr64
0x1aae4:    movsxd  rdx, dword ptr [rip + 0x11011]
0x1aaeb:    cmp     edx, dword ptr [rip + 0x10ff3]
0x1aaf1:    je      0x1ab08
0x1aaf3:    mov     rax, qword ptr [rip + 0x11006]
0x1aafa:    test    rax, rax
0x1aafd:    je      0x1ab0a
0x1aaff:    mov     rax, qword ptr [rax + rdx*8]
0x1ab03:    ret
0x1ab04:    nop     dword ptr [rax]
0x1ab08:    xor     eax, eax
0x1ab0a:    ret
0x1ab0b:    nop     dword ptr [rax + rax]
0x1ab10:    endbr64
0x1ab14:    mov     eax, dword ptr [rip + 0x10fe2]
0x1ab1a:    xor     r8d, r8d
0x1ab1d:    test    eax, eax
0x1ab1f:    je      0x1ab37
0x1ab21:    sub     eax, 1
0x1ab24:    mov     rdx, qword ptr [rip + 0x10fd5]
0x1ab2b:    mov     dword ptr [rip + 0x10fcb], eax
0x1ab31:    cdqe
0x1ab33:    mov     r8, qword ptr [rdx + rax*8]
0x1ab37:    mov     rax, r8

```

```

0x1ab3a:    ret
0x1ab3b:    nop     dword ptr [rax + rax]
0x1ab40:    endbr64
0x1ab44:    mov     eax, dword ptr [rip + 0x10fb2]
0x1ab4a:    cmp     eax, dword ptr [rip + 0x10f94]
0x1ab50:    je      0x1ab70
0x1ab52:    add     eax, 1
0x1ab55:    mov     rdx, qword ptr [rip + 0x10fa4]
0x1ab5c:    mov     dword ptr [rip + 0x10f9a], eax
0x1ab62:    cdqe
0x1ab64:    mov     rax, qword ptr [rdx + rax*8]
0x1ab68:    ret
0x1ab69:    nop     dword ptr [rax]
0x1ab70:    xor     eax, eax
0x1ab72:    ret
0x1ab73:    nop     word ptr cs:[rax + rax]
0x1ab7e:    nop
0x1ab80:    endbr64
0x1ab84:    mov     rax, qword ptr [rip + 0x10f75]
0x1ab8b:    ret
0x1ab8c:    nop     dword ptr [rax]
0x1ab90:    endbr64
0x1ab94:    sub     edi, dword ptr [rip + 0xfc4e]
0x1ab9a:    cmp     dword ptr [rip + 0x10f44], edi
0x1aba0:    jle     0x1abc0
0x1aba2:    test    edi, edi
0x1aba4:    js      0x1abc0
0x1aba6:    mov     rax, qword ptr [rip + 0x10f53]
0x1abad:    test    rax, rax
0x1abb0:    je      0x1abc2
0x1abb2:    movsxd  rdi, edi
0x1abb5:    mov     rax, qword ptr [rax + rdi*8]
0x1abb9:    ret
0x1abba:    nop     word ptr [rax + rax]
0x1abc0:    xor     eax, eax
0x1abc2:    ret
0x1abc3:    nop     word ptr cs:[rax + rax]
0x1abce:    nop
0x1abd0:    endbr64
0x1abd4:    cmp     edx, dword ptr [rip + 0x10f0a]
0x1abda:    push    rbx
0x1abdb:    mov     ebx, dword ptr [rip + 0x10f1b]
0x1abe1:    jg      0x1abf7
0x1abe3:    test    edx, edx
0x1abe5:    js      0x1abf7
0x1abe7:    cmp     qword ptr [rip + 0x10f11], 0
0x1abef:    je      0x1abf7
0x1abf1:    mov     dword ptr [rip + 0x10f05], edx

```

```

0x1abf7:    xor     edx, edx
0x1abf9:    xor     eax, eax
0x1abfb:    call    0x19de0
0x1ac00:    mov     edx, ebx
0x1ac02:    shr     edx, 0x1f
0x1ac05:    cmp     eax, -1
0x1ac08:    je      0x1ac10
0x1ac0a:    mov     eax, dword ptr [rip + 0x10eec]
0x1ac10:    cmp     ebx, dword ptr [rip + 0x10ece]
0x1ac16:    jg      0x1ac2c
0x1ac18:    test    dl, dl
0x1ac1a:    jne     0x1ac2c
0x1ac1c:    cmp     qword ptr [rip + 0x10edc], 0
0x1ac24:    je      0x1ac2c
0x1ac26:    mov     dword ptr [rip + 0x10ed0], ebx
0x1ac2c:    pop     rbx
0x1ac2d:    ret
0x1ac2e:    nop
0x1ac30:    endbr64
0x1ac34:    cmp     dword ptr [rip + 0x10eaa], edi
0x1ac3a:    jl      0x1ac60
0x1ac3c:    test    edi, edi
0x1ac3e:    js      0x1ac60
0x1ac40:    cmp     qword ptr [rip + 0x10eb8], 0
0x1ac48:    je      0x1ac60
0x1ac4a:    mov     dword ptr [rip + 0x10eac], edi
0x1ac50:    mov     eax, 1
0x1ac55:    ret
0x1ac56:    nop     word ptr cs:[rax + rax]
0x1ac60:    xor     eax, eax
0x1ac62:    ret
0x1ac63:    nop     word ptr cs:[rax + rax]
0x1ac6e:    nop
0x1ac70:    endbr64
0x1ac74:    movsxd  rax, dword ptr [rsi]
0x1ac77:    movzx   r8d, byte ptr [rip + 0xfb6e]
0x1ac7f:    cmp     byte ptr [rdi + rax], r8b
0x1ac83:    jne     0x1afd0
0x1ac89:    push    r15
0x1ac8b:    lea     ecx, [rax + 1]
0x1ac8e:    mov     r9d, edx
0x1ac91:    mov     rdx, rax
0x1ac94:    push    r14
0x1ac96:    movsxd  r10, ecx
0x1ac99:    push    r13
0x1ac9b:    mov     r13, rsi
0x1ac9e:    lea     rsi, [rdi + r10]
0x1aca2:    push    r12

```

```

0x1aca4:    push    rbp
0x1aca5:    push    rbx
0x1aca6:    sub     rsp, 0x18
0x1acaa:    movzx   eax, byte ptr [rsi]
0x1acad:    cmp     r8b, al
0x1acb0:    je      0x1af20
0x1acb6:    mov     r8d, 1
0x1acbc:    cmp     al, 0x2d
0x1acbe:    je      0x1aef8
0x1acc4:    lea     edx, [rax - 0x30]
0x1acc7:    cmp     dl, 9
0x1acca:    jbe     0x1ae68
0x1acd0:    mov     ebp, 1
0x1acd5:    xor     ebx, ebx
0x1acd7:    cmp     al, 0x3f
0x1acd9:    jne     0x1acef
0x1acdb:    add     ecx, 1
0x1acde:    xor     ebp, ebp
0x1ace0:    mov     ebx, 1
0x1ace5:    movsxd  r10, ecx
0x1ace8:    lea     rsi, [rdi + r10]
0x1acec:    movzx   eax, byte ptr [rsi]
0x1acef:    test    al, al
0x1acf1:    je      0x1b042
0x1acf7:    lea     rdx, [rdi + r10 + 1]
0x1acfc:    mov     r12d, ecx
0x1acff:    mov     r8, rsi
0x1ad02:    movabs  r10, 0x400000100000200
0x1ad0c:    jmp     0x1ad48
0x1ad0e:    nop
0x1ad10:    cmp     al, 0x3a
0x1ad12:    ja      0x1ad1e
0x1ad14:    bt      r10, rax
0x1ad18:    jb      0x1afd8
0x1ad1e:    movsx   edi, al
0x1ad21:    cmp     edi, r9d
0x1ad24:    je      0x1b010
0x1ad2a:    cmp     al, 0xa
0x1ad2c:    je      0x1ae10
0x1ad32:    mov     r8, rdx
0x1ad35:    movzx   eax, byte ptr [rdx]
0x1ad38:    lea     rdx, [rdx + 1]
0x1ad3c:    add     r12d, 1
0x1ad40:    test    al, al
0x1ad42:    je      0x1ae10
0x1ad48:    test    ebx, ebx
0x1ad4a:    je      0x1ad10
0x1ad4c:    cmp     al, 0xa

```



```

0x1ad4e:    je      0x1ae10
0x1ad54:    cmp     al, 0x3f
0x1ad56:    jne     0x1ad32
0x1ad58:    mov     r15d, r12d
0x1ad5b:    mov     qword ptr [rsp + 8], r8
0x1ad60:    sub     r15d, ecx
0x1ad63:    mov     qword ptr [rsp], rsi
0x1ad67:    lea     edi, [r15 + 1]
0x1ad6b:    movsxd  r15, r15d
0x1ad6e:    movsxd  rdi, edi
0x1ad71:    call    0x1d670
0x1ad76:    mov     rsi, qword ptr [rsp]
0x1ad7a:    mov     rdx, r15
0x1ad7d:    mov     rdi, rax
0x1ad80:    mov     r14, rax
0x1ad83:    call    0x4880
0x1ad88:    mov     r8, qword ptr [rsp + 8]
0x1ad8d:    xor     eax, eax
0x1ad8f:    mov     byte ptr [r14 + r15], 0
0x1ad94:    cmp     byte ptr [r8], 0x3f
0x1ad98:    sete    al
0x1ad9b:    add     r12d, eax
0x1ad9e:    nop
0x1ada0:    mov     dword ptr [r13], r12d
0x1ada4:    mov     r12d, ebx
0x1ada7:    and     r12d, 1
0x1adab:    jmp     0x1add3
0x1adad:    nop     dword ptr [rax]
0x1adb0:    movsxd  rax, dword ptr [rip + 0x10d45]
0x1adb7:    je      0x1af78
0x1adb9:    test    r12b, r12b
0x1adc0:    jne     0x1af78
0x1adc6:    test    eax, eax
0x1adc8:    je      0x1ade8
0x1adca:    sub     eax, 1
0x1adcd:    mov     dword ptr [rip + 0x10d29], eax
0x1add3:    xor     eax, eax
0x1add5:    mov     edx, ebp
0x1add7:    mov     esi, 0xffffffff
0x1addc:    mov     rdi, r14
0x1addf:    call    0x19de0
0x1ade4:    test    eax, eax
0x1ade6:    jns     0x1adb0
0x1ade8:    mov     eax, dword ptr [rip + 0x10cf6]
0x1adee:    mov     rdi, r14
0x1adf1:    mov     dword ptr [rip + 0x10d05], eax
0x1adf7:    call    0x18d10
0x1adfc:    add     rsp, 0x18

```

```

0x1ae00:    xor     eax, eax
0x1ae02:    pop     rbx
0x1ae03:    pop     rbp
0x1ae04:    pop     r12
0x1ae06:    pop     r13
0x1ae08:    pop     r14
0x1ae0a:    pop     r15
0x1ae0c:    ret
0x1ae0d:    nop     dword ptr [rax]
0x1ae10:    mov     r15d, r12d
0x1ae13:    sub     r15d, ecx
0x1ae16:    lea     edi, [r15 + 1]
0x1ae1a:    movsxd  r15, r15d
0x1ae1d:    movsxd  rdi, edi
0x1ae20:    mov     qword ptr [rsp + 8], r8
0x1ae25:    mov     qword ptr [rsp], rsi
0x1ae29:    call    0x1d670
0x1ae2e:    mov     rsi, qword ptr [rsp]
0x1ae32:    mov     rdx, r15
0x1ae35:    mov     rdi, rax
0x1ae38:    mov     r14, rax
0x1ae3b:    call    0x4880
0x1ae40:    test    ebx, ebx
0x1ae42:    mov     byte ptr [r14 + r15], 0
0x1ae47:    mov     r8, qword ptr [rsp + 8]
0x1ae4c:    je      0x1ada0
0x1ae52:    xor     eax, eax
0x1ae54:    cmp     byte ptr [r8], 0x3f
0x1ae58:    sete    al
0x1ae5b:    add     r12d, eax
0x1ae5e:    jmp     0x1ada0
0x1ae63:    nop     dword ptr [rax + rax]
0x1ae68:    lea     edx, [rcx + 1]
0x1ae6b:    xor     ecx, ecx
0x1ae6d:    movsxd  rdx, edx
0x1ae70:    sub     eax, 0x30
0x1ae73:    lea     ecx, [rcx + rcx*4]
0x1ae76:    mov     r9d, edx
0x1ae79:    movsx   eax, al
0x1ae7c:    lea     ecx, [rax + rcx*2]
0x1ae7f:    movzx   eax, byte ptr [rdi + rdx]
0x1ae83:    add     rdx, 1
0x1ae87:    lea     esi, [rax - 0x30]
0x1ae8a:    cmp     sil, 9
0x1ae8e:    jbe     0x1ae70
0x1ae90:    mov     dword ptr [r13], r9d
0x1ae94:    mov     eax, dword ptr [rip + 0x10c4a]
0x1ae9a:    mov     edx, dword ptr [rip + 0xf948]

```

```

0x1aea0:      cmp     r8d, -1
0x1aea4:      jne     0x1aead
0x1aea6:      lea     esi, [rax + rdx]
0x1aea9:      sub     esi, ecx
0x1aeab:      mov     ecx, esi
0x1aead:      sub     ecx, edx
0x1aeaf:      cmp     ecx, eax
0x1aeb1:      jge     0x1adfc
0x1aeb7:      test    ecx, ecx
0x1aeb9:      js      0x1adfc
0x1aebf:      mov     rax, qword ptr [rip + 0x10c3a]
0x1aec6:      test    rax, rax
0x1aec9:      je      0x1adfc
0x1aecf:      movsxd   rcx, ecx
0x1aed2:      mov     rax, qword ptr [rax + rcx*8]
0x1aed6:      test    rax, rax
0x1aed9:      je      0x1adfc
0x1aedef:      mov     rax, qword ptr [rax]
0x1aee2:      add     rsp, 0x18
0x1aee6:      pop     rbx
0x1aee7:      pop     rbp
0x1aee8:      pop     r12
0x1aeea:      pop     r13
0x1aeec:      pop     r14
0x1aeef:      pop     r15
0x1aef0:      ret
0x1aef1:      nop     dword ptr [rax]
0x1aef8:      lea     ecx, [rdx + 2]
0x1aefb:      mov     r8d, 0xffffffff
0x1af01:      movsxd   r10, ecx
0x1af04:      lea     rsi, [rdi + r10]
0x1af08:      movzx   eax, byte ptr [rsi]
0x1af0b:      lea     edx, [rax - 0x30]
0x1af0e:      cmp     dl, 9
0x1af11:      ja      0x1acd0
0x1af17:      jmp     0x1ae68
0x1af1c:      nop     dword ptr [rax]
0x1af20:      mov     ecx, dword ptr [rip + 0x10bbe]
0x1af26:      mov     eax, dword ptr [rip + 0xf8bc]
0x1af2c:      add     edx, 2
0x1af2f:      mov     dword ptr [r13], edx
0x1af33:      lea     eax, [rcx + rax - 1]
0x1af37:      sub     eax, dword ptr [rip + 0xf8ab]
0x1af3d:      cmp     eax, dword ptr [rip + 0x10ba1]
0x1af43:      jge     0x1adfc
0x1af49:      test    eax, eax
0x1af4b:      js      0x1adfc
0x1af51:      mov     rdx, qword ptr [rip + 0x10ba8]

```

```

0x1af58:    test    rdx, rdx
0x1af5b:    je      0x1adfc
0x1af61:    cdqe
0x1af63:    mov     rax, qword ptr [rdx + rax*8]
0x1af67:    test    rax, rax
0x1af6a:    jne     0x1aedf
0x1af70:    jmp     0x1adfc
0x1af75:    nop     dword ptr [rax]
0x1af78:    mov     edx, dword ptr [rip + 0x10b66]
0x1af7e:    cmp     edx, eax
0x1af80:    je      0x1b03b
0x1af86:    mov     rbp, qword ptr [rip + 0x10b73]
0x1af8d:    test    rbp, rbp
0x1af90:    je      0x1af97
0x1af92:    mov     rbp, qword ptr [rbp + rax*8]
0x1af97:    mov     dword ptr [rip + 0x10b5f], edx
0x1af9d:    test    ebx, ebx
0x1af9f:    je      0x1b02e
0x1afa5:    mov     rdi, qword ptr [rip + 0x10b44]
0x1afac:    test    rdi, rdi
0x1afaf:    je      0x1afb6
0x1afb1:    call    0x18d10
0x1afb6:    mov     qword ptr [rip + 0x10b33], r14
0x1afbd:    mov     rax, qword ptr [rbp]
0x1afc1:    add     rsp, 0x18
0x1afc5:    pop     rbx
0x1afc6:    pop     rbp
0x1afc7:    pop     r12
0x1afc9:    pop     r13
0x1afcb:    pop     r14
0x1afcd:    pop     r15
0x1afcf:    ret
0x1afd0:    xor     eax, eax
0x1afd2:    ret
0x1afd3:    nop     dword ptr [rax + rax]
0x1afd8:    mov     eax, r12d
0x1afdb:    mov     qword ptr [rsp], rsi
0x1afdf:    sub     eax, ecx
0x1afe1:    movsxd  r15, eax
0x1afe4:    add     eax, 1
0x1afe7:    movsxd  rdi, eax
0x1afea:    call    0x1d670
0x1afef:    mov     r14, rax
0x1aff2:    mov     rsi, qword ptr [rsp]
0x1aff6:    mov     rdx, r15
0x1aff9:    mov     rdi, r14
0x1affc:    call    0x4880
0x1b001:    mov     byte ptr [r14 + r15], 0

```

```

0x1b006:      jmp      0x1ada0
0x1b00b:      nop      dword ptr [rax + rax]
0x1b010:      mov      r15d, r12d
0x1b013:      mov      qword ptr [rsp], rsi
0x1b017:      sub      r15d, ecx
0x1b01a:      lea      edi, [r15 + 1]
0x1b01e:      movsxd   r15, r15d
0x1b021:      movsxd   rdi, edi
0x1b024:      call     0x1d670
0x1b029:      mov      r14, rax
0x1b02c:      jmp      0x1aff2
0x1b02e:      mov      rdi, r14
0x1b031:      call     0x18d10
0x1b036:      jmp      0x1afbd
0x1b03b:      xor      ebp, ebp
0x1b03d:      jmp      0x1af97
0x1b042:      mov      r12d, ecx
0x1b045:      mov      r8, rsi
0x1b048:      xor      r15d, r15d
0x1b04b:      mov      edi, 1
0x1b050:      jmp      0x1ae20
0x1b055:      nop      word ptr cs:[rax + rax]
0x1b060:      endbr64
0x1b064:      push     r15
0x1b066:      push     r14
0x1b068:      push     r13
0x1b06a:      push     r12
0x1b06c:      push     rbp
0x1b06d:      push     rbx
0x1b06e:      sub      rsp, 0x28
0x1b072:      movzx    r14d, byte ptr [rdi]
0x1b076:      test     r14b, r14b
0x1b079:      je       0x1b30d
0x1b07f:      mov      qword ptr [rsp], 0
0x1b087:      mov      r12, rdi
0x1b08a:      mov      r13, rdi
0x1b08d:      xor      ebx, ebx
0x1b08f:      mov      dword ptr [rsp + 8], 0
0x1b097:      xor      ebp, ebp
0x1b099:      mov      dword ptr [rsp + 0xc], 0
0x1b0a1:      jmp      0x1b0c0
0x1b0a3:      nop      dword ptr [rax + rax]
0x1b0a8:      add      ebx, 1
0x1b0ab:      movsxd   rbp, ebx
0x1b0ae:      lea      r13, [r12 + rbp]
0x1b0b2:      movzx    r14d, byte ptr [r13]
0x1b0b7:      test     r14b, r14b
0x1b0ba:      je       0x1b315

```

```

0x1b0c0:      cmp     r14b, 0x20
0x1b0c4:      je      0x1b0a8
0x1b0c6:      cmp     r14b, 9
0x1b0ca:      je      0x1b0a8
0x1b0cc:      cmp     byte ptr [rip + 0x10a0d], r14b
0x1b0d3:      je      0x1b315
0x1b0d9:      movsx   r15d, r14b
0x1b0dd:      lea      rdi, [rip + 0x7379]
0x1b0e4:      mov     esi, r15d
0x1b0e7:      call    0x4a00
0x1b0ec:      test    rax, rax
0x1b0ef:      je      0x1b190
0x1b0f5:      add     ebx, 1
0x1b0f8:      mov     r14d, 1
0x1b0fe:      mov     r9d, 2
0x1b104:      movsxd  rbp, ebx
0x1b107:      lea      rcx, [r12 + rbp]
0x1b10b:      mov     eax, dword ptr [rsp + 8]
0x1b10f:      add     eax, 2
0x1b112:      cmp     eax, dword ptr [rsp + 0xc]
0x1b116:      jge     0x1b2a0
0x1b11c:      movsxd  r8, dword ptr [rsp + 8]
0x1b121:      mov     rdx, qword ptr [rsp]
0x1b125:      mov     rdi, r9
0x1b128:      mov     qword ptr [rsp + 0x18], rcx
0x1b12d:      shl     r8, 3
0x1b131:      lea      r15, [rdx + r8]
0x1b135:      mov     qword ptr [rsp + 0x10], r8
0x1b13a:      call    0x1d670
0x1b13f:      mov     rdx, r14
0x1b142:      mov     rsi, r13
0x1b145:      mov     qword ptr [r15], rax
0x1b148:      mov     rdi, rax
0x1b14b:      call    0x4880
0x1b150:      mov     rax, qword ptr [r15]
0x1b153:      mov     r8, qword ptr [rsp + 0x10]
0x1b158:      mov     rcx, qword ptr [rsp + 0x18]
0x1b15d:      add     dword ptr [rsp + 8], 1
0x1b162:      mov     byte ptr [rax + r14], 0
0x1b167:      mov     rax, qword ptr [rsp]
0x1b16b:      mov     qword ptr [rax + r8 + 8], 0
0x1b174:      movzx   r14d, byte ptr [rcx]
0x1b178:      test    r14b, r14b
0x1b17b:      je      0x1b315
0x1b181:      mov     r13, rcx
0x1b184:      jmp     0x1b0c0
0x1b189:      nop     dword ptr [rax]
0x1b190:      mov     esi, r15d

```

```

0x1b193:    lea     rdi, [rip + 0x72c7]
0x1b19a:    call    0x4a00
0x1b19f:    test    rax, rax
0x1b1a2:    je      0x1b369
0x1b1a8:    movsx   edx, byte ptr [r12 + rbp + 1]
0x1b1ae:    cmp     dl, r14b
0x1b1b1:    je      0x1b2d8
0x1b1b7:    cmp     edx, 0x26
0x1b1ba:    je      0x1b340
0x1b1c0:    cmp     edx, 0x3e
0x1b1c3:    jne     0x1b3ac
0x1b1c9:    cmp     r14b, 0x26
0x1b1cd:    je      0x1b34e
0x1b1d3:    cmp     r14b, 0x24
0x1b1d7:    jne     0x1b0f5
0x1b1dd:    mov     edx, ebx
0x1b1df:    xor     r15d, r15d
0x1b1e2:    movsxd  r8, edx
0x1b1e5:    lea     rcx, [r12 + r8]
0x1b1e9:    mov     rbp, r8
0x1b1ec:    movzx   eax, byte ptr [rcx]
0x1b1ef:    test    al, al
0x1b1f1:    je      0x1b404
0x1b1f7:    mov     qword ptr [rsp + 0x10], r13
0x1b1fc:    lea     r14, [rip + 0x7265]
0x1b203:    mov     r13, r8
0x1b206:    mov     dword ptr [rsp + 0x18], ebx
0x1b20a:    mov     rbx, r12
0x1b20d:    mov     r12d, edx
0x1b210:    jmp     0x1b265
0x1b212:    nop     word ptr [rax + rax]
0x1b218:    movsx   ebp, al
0x1b21b:    test    r15d, r15d
0x1b21e:    jne     0x1b290
0x1b220:    mov     esi, ebp
0x1b222:    lea     rdi, [rip + 0x7243]
0x1b229:    call    0x4a00
0x1b22e:    test    rax, rax
0x1b231:    jne     0x1b3c0
0x1b237:    mov     esi, ebp
0x1b239:    mov     rdi, r14
0x1b23c:    call    0x4a00
0x1b241:    lea     esi, [r12 + 1]
0x1b246:    test    rax, rax
0x1b249:    cmovne  r15d, ebp
0x1b24d:    movsxd  rbp, esi
0x1b250:    mov     r12d, esi
0x1b253:    lea     rcx, [rbx + rbp]

```

```

0x1b257:    mov     r13, rbp
0x1b25a:    movzx   eax, byte ptr [rcx]
0x1b25d:    test    al, al
0x1b25f:    je      0x1b389
0x1b265:    cmp     al, 0x5c
0x1b267:    jne     0x1b218
0x1b269:    cmp     byte ptr [rbx + r13 + 1], 0xa
0x1b26f:    lea     esi, [r12 + 1]
0x1b274:    je      0x1b330
0x1b27a:    xor     eax, eax
0x1b27c:    cmp     r15d, 0x27
0x1b280:    setne   al
0x1b283:    add     esi, eax
0x1b285:    jmp     0x1b24d
0x1b287:    nop     word ptr [rax + rax]
0x1b290:    xor     eax, eax
0x1b292:    cmp     r15d, ebp
0x1b295:    lea     esi, [r12 + 1]
0x1b29a:    cmov     r15d, eax
0x1b29e:    jmp     0x1b24d
0x1b2a0:    add     dword ptr [rsp + 0xc], 0xa
0x1b2a5:    movsxd   rsi, dword ptr [rsp + 0xc]
0x1b2aa:    mov     rdi, qword ptr [rsp]
0x1b2ae:    mov     qword ptr [rsp + 0x18], rcx
0x1b2b3:    shl     rsi, 3
0x1b2b7:    mov     qword ptr [rsp + 0x10], r9
0x1b2bc:    call    0x1d690
0x1b2c1:    mov     rcx, qword ptr [rsp + 0x18]
0x1b2c6:    mov     r9, qword ptr [rsp + 0x10]
0x1b2cb:    mov     qword ptr [rsp], rax
0x1b2cf:    jmp     0x1b11c
0x1b2d4:    nop     dword ptr [rax]
0x1b2d8:    cmp     edx, 0x24
0x1b2db:    je      0x1b1d3
0x1b2e1:    mov     eax, ebx
0x1b2e3:    cmp     edx, 0x3c
0x1b2e6:    je      0x1b3f0
0x1b2ec:    add     eax, 2
0x1b2ef:    mov     r14d, eax
0x1b2f2:    movsxd   rbp, eax
0x1b2f5:    sub     r14d, ebx
0x1b2f8:    lea     rcx, [r12 + rbp]
0x1b2fc:    mov     ebx, eax
0x1b2fe:    lea     r9d, [r14 + 1]
0x1b302:    movsxd   r14, r14d
0x1b305:    movsxd   r9, r9d
0x1b308:    jmp     0x1b10b
0x1b30d:    mov     qword ptr [rsp], 0

```



```

0x1b315:    mov     rax, qword ptr [rsp]
0x1b319:    add     rsp, 0x28
0x1b31d:    pop     rbx
0x1b31e:    pop     rbp
0x1b31f:    pop     r12
0x1b321:    pop     r13
0x1b323:    pop     r14
0x1b325:    pop     r15
0x1b327:    ret
0x1b328:    nop     dword ptr [rax + rax]
0x1b330:    lea     esi, [r12 + 2]
0x1b335:    jmp     0x1b24d
0x1b33a:    nop     word ptr [rax + rax]
0x1b340:    mov     eax, r14d
0x1b343:    and     eax, 0xffffffff
0x1b346:    cmp     al, 0x3c
0x1b348:    jne     0x1b1d3
0x1b34e:    add     ebx, 2
0x1b351:    mov     r14d, 2
0x1b357:    mov     r9d, 3
0x1b35d:    movsxd  rbp, ebx
0x1b360:    lea     rcx, [r12 + rbp]
0x1b364:    jmp     0x1b10b
0x1b369:    mov     esi, r15d
0x1b36c:    lea     rdi, [rip + 0x70f5]
0x1b373:    call    0x4a00
0x1b378:    test    rax, rax
0x1b37b:    je      0x1b1dd
0x1b381:    lea     edx, [rbx + 1]
0x1b384:    jmp     0x1b1e2
0x1b389:    mov     r12, rbx
0x1b38c:    mov     ebx, dword ptr [rsp + 0x18]
0x1b390:    mov     r14d, esi
0x1b393:    mov     r13, qword ptr [rsp + 0x10]
0x1b398:    sub     r14d, ebx
0x1b39b:    mov     ebx, esi
0x1b39d:    lea     r9d, [r14 + 1]
0x1b3a1:    movsxd  r14, r14d
0x1b3a4:    movsxd  r9, r9d
0x1b3a7:    jmp     0x1b10b
0x1b3ac:    cmp     r14b, 0x24
0x1b3b0:    jne     0x1b0f5
0x1b3b6:    cmp     edx, 0x28
0x1b3b9:    je      0x1b34e
0x1b3bb:    jmp     0x1b1d3
0x1b3c0:    mov     edx, r12d
0x1b3c3:    mov     r12, rbx
0x1b3c6:    mov     ebx, dword ptr [rsp + 0x18]

```

```

0x1b3ca:    mov     r8, r13
0x1b3cd:    mov     r14d, edx
0x1b3d0:    mov     r13, qword ptr [rsp + 0x10]
0x1b3d5:    mov     rbp, r8
0x1b3d8:    lea     rcx, [r12 + r8]
0x1b3dc:    sub     r14d, ebx
0x1b3df:    mov     ebx, edx
0x1b3e1:    lea     r9d, [r14 + 1]
0x1b3e5:    movsxd  r14, r14d
0x1b3e8:    movsxd  r9, r9d
0x1b3eb:    jmp     0x1b10b
0x1b3f0:    cmp     byte ptr [r12 + rbp + 2], 0x2d
0x1b3f6:    jne     0x1b2ec
0x1b3fc:    lea     eax, [rbx + 1]
0x1b3ff:    jmp     0x1b2ec
0x1b404:    mov     r14d, edx
0x1b407:    sub     r14d, ebx
0x1b40a:    mov     ebx, edx
0x1b40c:    lea     r9d, [r14 + 1]
0x1b410:    movsxd  r14, r14d
0x1b413:    movsxd  r9, r9d
0x1b416:    jmp     0x1b10b
0x1b41b:    nop     dword ptr [rax + rax]
0x1b420:    endbr64
0x1b424:    push    r15
0x1b426:    xor     eax, eax
0x1b428:    push    r14
0x1b42a:    push    r13
0x1b42c:    push    r12
0x1b42e:    mov     r12d, esi
0x1b431:    push    rbp
0x1b432:    push    rbx
0x1b433:    mov     ebx, edi
0x1b435:    mov     rdi, rdx
0x1b438:    sub     rsp, 8
0x1b43c:    call    0x1b060
0x1b441:    test    rax, rax
0x1b444:    je      0x1b598
0x1b44a:    cmp     qword ptr [rax], 0
0x1b44e:    mov     rbp, rax
0x1b451:    je      0x1b590
0x1b457:    mov     eax, 1
0x1b45c:    nop     dword ptr [rax]
0x1b460:    mov     r14d, eax
0x1b463:    add     rax, 1
0x1b467:    cmp     qword ptr [rbp + rax*8 - 8], 0
0x1b46d:    jne     0x1b460
0x1b46f:    lea     eax, [r14 + r12 - 1]

```

```

0x1b474:      test    r12d, r12d
0x1b477:      cmovs   r12d, eax
0x1b47b:      lea     eax, [r14 + rbx - 1]
0x1b480:      test    ebx, ebx
0x1b482:      cmovs   ebx, eax
0x1b485:      lea     eax, [r12 + 1]
0x1b48a:      cmp     r12d, 0x24
0x1b48e:      jne     0x1b497
0x1b490:      lea     r12d, [r14 - 1]
0x1b494:      mov     eax, r14d
0x1b497:      lea     edx, [r14 - 1]
0x1b49b:      cmp     ebx, 0x24
0x1b49e:      cmov     ebx, edx
0x1b4a1:      mov     edx, eax
0x1b4a3:      cmp     ebx, eax
0x1b4a5:      cmovge   edx, ebx
0x1b4a8:      xor     r13d, r13d
0x1b4ab:      cmp     edx, r14d
0x1b4ae:      jg      0x1b547
0x1b4b4:      mov     ecx, ebx
0x1b4b6:      or      ecx, eax
0x1b4b8:      js      0x1b547
0x1b4be:      cmp     ebx, eax
0x1b4c0:      jge     0x1b59d
0x1b4c6:      movsxd  rbx, ebx
0x1b4c9:      xor     r15d, r15d
0x1b4cc:      mov     r13, rbx
0x1b4cf:      nop
0x1b4d0:      mov     rdi, qword ptr [rbp + r13*8]
0x1b4d5:      add     r13, 1
0x1b4d9:      call    0x49b0
0x1b4de:      lea     r15d, [r15 + rax + 1]
0x1b4e3:      lea     eax, [r13 - 1]
0x1b4e7:      cmp     r12d, eax
0x1b4ea:      jg      0x1b4d0
0x1b4ec:      lea     edi, [r15 + 1]
0x1b4f0:      xor     r15d, r15d
0x1b4f3:      movsxd  rdi, edi
0x1b4f6:      call    0x1d670
0x1b4fb:      mov     r13, rax
0x1b4fe:      nop
0x1b500:      mov     rsi, qword ptr [rbp + rbx*8]
0x1b505:      movsxd  rdi, r15d
0x1b508:      add     rdi, r13
0x1b50b:      call    0x48b0
0x1b510:      mov     rdi, qword ptr [rbp + rbx*8]
0x1b515:      call    0x49b0
0x1b51a:      add     r15d, eax

```

```

0x1b51d:      cmp     r12d, ebx
0x1b520:      jle     0x1b53b
0x1b522:      lea     eax, [r15 + 1]
0x1b526:      movsxd  r15, r15d
0x1b529:      movsxd  rdx, eax
0x1b52c:      mov     byte ptr [r13 + r15], 0x20
0x1b532:      mov     r15d, eax
0x1b535:      mov     byte ptr [r13 + rdx], 0
0x1b53b:      add     rbx, 1
0x1b53f:      lea     eax, [rbx - 1]
0x1b542:      cmp     r12d, eax
0x1b545:      jg      0x1b500
0x1b547:      test    r14d, r14d
0x1b54a:      je      0x1b571
0x1b54c:      lea     eax, [r14 - 1]
0x1b550:      mov     rbx, rbp
0x1b553:      lea     r12, [rbp + rax*8 + 8]
0x1b558:      nop     dword ptr [rax + rax]
0x1b560:      mov     rdi, qword ptr [rbx]
0x1b563:      add     rbx, 8
0x1b567:      call    0x18d10
0x1b56c:      cmp     r12, rbx
0x1b56f:      jne     0x1b560
0x1b571:      mov     rdi, rbp
0x1b574:      call    0x18d10
0x1b579:      add     rsp, 8
0x1b57d:      mov     rax, r13
0x1b580:      pop     rbx
0x1b581:      pop     rbp
0x1b582:      pop     r12
0x1b584:      pop     r13
0x1b586:      pop     r14
0x1b588:      pop     r15
0x1b58a:      ret
0x1b58b:      nop     dword ptr [rax + rax]
0x1b590:      xor     r14d, r14d
0x1b593:      jmp     0x1b46f
0x1b598:      xor     r13d, r13d
0x1b59b:      jmp     0x1b579
0x1b59d:      mov     edi, 1
0x1b5a2:      call    0x1d670
0x1b5a7:      mov     r13, rax
0x1b5aa:      jmp     0x1b547
0x1b5ac:      nop     dword ptr [rax]
0x1b5b0:      endbr64
0x1b5b4:      push    r15
0x1b5b6:      push    r14
0x1b5b8:      push    r13

```

```

0x1b5ba:    push    r12
0x1b5bc:    push    rbp
0x1b5bd:    push    rbx
0x1b5be:    mov     rbx, rdi
0x1b5c1:    sub     rsp, 0x88
0x1b5c8:    mov     qword ptr [rsp + 0x20], rdi
0x1b5cd:    mov     qword ptr [rsp + 0x30], rsi
0x1b5d2:    mov     rax, qword ptr fs:[0x28]
0x1b5db:    mov     qword ptr [rsp + 0x78], rax
0x1b5e0:    xor     eax, eax
0x1b5e2:    cmp     byte ptr [rip + 0xf204], 0
0x1b5e9:    je      0x1b656
0x1b5eb:    mov     edi, 0x100
0x1b5f0:    call    0x1d670
0x1b5f5:    mov     rbx, qword ptr [rsp + 0x20]
0x1b5fa:    mov     byte ptr [rax], 0
0x1b5fd:    mov     r15, rax
0x1b600:    mov     rdi, rbx
0x1b603:    call    0x49b0
0x1b608:    mov     dword ptr [rsp + 0xc], eax
0x1b60c:    mov     r12, rax
0x1b60f:    movzx   eax, byte ptr [rbx]
0x1b612:    cmp     al, byte ptr [rip + 0xf1d4]
0x1b618:    je      0x1c1e0
0x1b61e:    movzx   ebp, byte ptr [rip + 0xf1c8]
0x1b625:    mov     r13, qword ptr [rip + 0xf1c4]
0x1b62c:    lea     rbx, [rbx + 1]
0x1b630:    test    al, al
0x1b632:    jne     0x1b6c7
0x1b638:    nop     dword ptr [rax + rax]
0x1b640:    test    bpl, bpl
0x1b643:    je      0x1b6d8
0x1b649:    mov     rdi, r15
0x1b64c:    call    0x18d10
0x1b651:    mov     rbx, qword ptr [rsp + 0x20]
0x1b656:    mov     rdi, rbx
0x1b659:    call    0x49b0
0x1b65e:    lea     rdi, [rax + 1]
0x1b662:    call    0x1d670
0x1b667:    mov     rsi, rbx
0x1b66a:    mov     rdi, rax
0x1b66d:    call    0x48b0
0x1b672:    mov     rcx, qword ptr [rsp + 0x30]
0x1b677:    mov     qword ptr [rcx], rax
0x1b67a:    xor     eax, eax
0x1b67c:    mov     rdx, qword ptr [rsp + 0x78]
0x1b681:    sub     rdx, qword ptr fs:[0x28]
0x1b68a:    jne     0x1c487

```

```

0x1b690:    add     rsp, 0x88
0x1b697:    pop     rbx
0x1b698:    pop     rbp
0x1b699:    pop     r12
0x1b69b:    pop     r13
0x1b69d:    pop     r14
0x1b69f:    pop     r15
0x1b6a1:    ret
0x1b6a2:    nop     word ptr [rax + rax]
0x1b6a8:    mov     rdi, r13
0x1b6ab:    call    0x4a00
0x1b6b0:    test    rax, rax
0x1b6b3:    je      0x1b6d8
0x1b6b5:    nop     dword ptr [rax]
0x1b6b8:    movzx   eax, byte ptr [rbx]
0x1b6bb:    add     rbx, 1
0x1b6bf:    test    al, al
0x1b6c1:    je      0x1b640
0x1b6c7:    movsx   esi, byte ptr [rbx]
0x1b6ca:    cmp     al, bpl
0x1b6cd:    jne     0x1b6b8
0x1b6cf:    test    esi, esi
0x1b6d1:    je      0x1b6b8
0x1b6d3:    jmp     0x1b6a8
0x1b6d5:    nop     dword ptr [rax]
0x1b6d8:    mov     rbp, qword ptr [rsp + 0x20]
0x1b6dd:    test    r12d, r12d
0x1b6e0:    jle     0x1c40a
0x1b6e6:    mov     dword ptr [rsp + 4], 0x100
0x1b6ee:    xor     r13d, r13d
0x1b6f1:    xor     ebx, ebx
0x1b6f3:    mov     dword ptr [rsp + 0x28], 0
0x1b6fb:    mov     dword ptr [rsp + 0x3c], 0
0x1b703:    mov     dword ptr [rsp + 8], 0
0x1b70b:    jmp     0x1b758
0x1b70d:    nop     dword ptr [rax]
0x1b710:    cmp     r12b, byte ptr [rip + 0xf0d6]
0x1b717:    je      0x1b7d0
0x1b71d:    cmp     r12b, 0xfd
0x1b721:    je      0x1b7d0
0x1b727:    cmp     r12b, 0x5c
0x1b72b:    je      0x1bde0
0x1b731:    cmp     ebx, dword ptr [rsp + 4]
0x1b735:    jge     0x1bfe0
0x1b73b:    lea     eax, [rbx + 1]
0x1b73e:    mov     byte ptr [r15 + rbx], r12b
0x1b742:    movsxd  rdx, eax
0x1b745:    movsxd  rbx, eax

```

```

0x1b748:    mov     byte ptr [r15 + rdx], 0
0x1b74d:    add     r13d, 1
0x1b751:    cmp     r13d, dword ptr [rsp + 0xc]
0x1b756:    jge     0x1b79d
0x1b758:    movsxd  r14, r13d
0x1b75b:    mov     r10d, dword ptr [rsp + 8]
0x1b760:    lea     rdx, [rbp + r14]
0x1b765:    movzx   r12d, byte ptr [rdx]
0x1b769:    test    r10d, r10d
0x1b76c:    je      0x1b710
0x1b76e:    cmp     ebx, dword ptr [rsp + 4]
0x1b772:    jge     0x1be10
0x1b778:    lea     eax, [rbx + 1]
0x1b77b:    mov     byte ptr [r15 + rbx], r12b
0x1b77f:    add     r13d, 1
0x1b783:    movsxd  rdx, eax
0x1b786:    mov     dword ptr [rsp + 8], 0
0x1b78e:    movsxd  rbx, eax
0x1b791:    mov     byte ptr [r15 + rdx], 0
0x1b796:    cmp     r13d, dword ptr [rsp + 0xc]
0x1b79b:    jl      0x1b758
0x1b79d:    mov     rax, qword ptr [rsp + 0x30]
0x1b7a2:    mov     qword ptr [rax], r15
0x1b7a5:    cmp     qword ptr [rsp + 0x20], rbp
0x1b7aa:    je      0x1b7b4
0x1b7ac:    mov     rdi, rbp
0x1b7af:    call    0x18d10
0x1b7b4:    mov     ecx, dword ptr [rsp + 0x28]
0x1b7b8:    test    ecx, ecx
0x1b7ba:    jne     0x1c35b
0x1b7c0:    mov     edx, dword ptr [rsp + 0x3c]
0x1b7c4:    xor     eax, eax
0x1b7c6:    test    edx, edx
0x1b7c8:    setne   al
0x1b7cb:    jmp     0x1b67c
0x1b7d0:    lea     rcx, [rbp + r14 + 1]
0x1b7d5:    mov     qword ptr [rsp + 0x10], rdx
0x1b7da:    movsx   esi, byte ptr [rcx]
0x1b7dd:    test    esi, esi
0x1b7df:    je      0x1b830
0x1b7e1:    mov     rdi, qword ptr [rip + 0xf008]
0x1b7e8:    mov     qword ptr [rsp + 0x18], rcx
0x1b7ed:    call    0x4a00
0x1b7f2:    mov     rcx, qword ptr [rsp + 0x18]
0x1b7f7:    test    rax, rax
0x1b7fa:    je      0x1b830
0x1b7fc:    cmp     ebx, dword ptr [rsp + 4]
0x1b800:    jl      0x1b73b

```

```

0x1b806:      add     dword ptr [rsp + 4], 0x40
0x1b80b:      movsxd  rsi, dword ptr [rsp + 4]
0x1b810:      mov     rdi, r15
0x1b813:      call    0x1d690
0x1b818:      mov     rdx, qword ptr [rsp + 0x10]
0x1b81d:      mov     r15, rax
0x1b820:      movzx   r12d, byte ptr [rdx]
0x1b824:      jmp     0x1b73b
0x1b829:      nop     dword ptr [rax]
0x1b830:      mov     edi, 0x80
0x1b835:      mov     qword ptr [rsp + 0x18], rcx
0x1b83a:      call    0x1d670
0x1b83f:      mov     rcx, qword ptr [rsp + 0x18]
0x1b844:      mov     dword ptr [rsp + 0x6c], r13d
0x1b849:      mov     qword ptr [rsp + 0x10], rax
0x1b84e:      movzx   r12d, byte ptr [rcx]
0x1b852:      test    r12b, r12b
0x1b855:      je      0x1be30
0x1b85b:      movsx   esi, r12b
0x1b85f:      lea     rdi, [rip + 0x6c11]
0x1b866:      call    0x4a00
0x1b86b:      test    rax, rax
0x1b86e:      je      0x1bfc0
0x1b874:      lea     eax, [r13 + 1]
0x1b878:      lea     rsi, [rsp + 0x70]
0x1b87d:      xor     edx, edx
0x1b87f:      mov     byte ptr [rsp + 0x77], 0
0x1b884:      mov     dword ptr [rsp + 0x6c], eax
0x1b888:      movzx   eax, byte ptr [rip + 0xef5e]
0x1b88f:      lea     rdi, [rsp + 0x75]
0x1b894:      mov     dword ptr [rsp + 0x70], 0
0x1b89c:      mov     byte ptr [rsp + 0x76], al
0x1b8a0:      mov     byte ptr [rsp + 0x75], al
0x1b8a4:      xor     eax, eax
0x1b8a6:      call    0x1ac70
0x1b8ab:      mov     r12d, dword ptr [rsp + 0x6c]
0x1b8b0:      mov     r14, rax
0x1b8b3:      test    r14, r14
0x1b8b6:      je      0x1c435
0x1b8bc:      movsxd  rax, r12d
0x1b8bf:      xor     esi, esi
0x1b8c1:      mov     edx, r12d
0x1b8c4:      movzx   eax, byte ptr [rbp + rax]
0x1b8c9:      cmp     al, 0x3a
0x1b8cb:      jne     0x1b8df
0x1b8cd:      lea     edx, [r12 + 1]
0x1b8d2:      mov     esi, 1
0x1b8d7:      movsxd  rax, edx

```



```

0x1b8da:    movzx    eax, byte ptr [rbp + rax]
0x1b8df:    cmp     al, 0x25
0x1b8e1:    je      0x1c010
0x1b8e7:    cmp     al, 0x2d
0x1b8e9:    je      0x1c100
0x1b8ef:    jg      0x1bd48
0x1b8f5:    cmp     al, 0x24
0x1b8f7:    je      0x1bf80
0x1b8fd:    cmp     al, 0x2a
0x1b8ff:    jne     0x1bd54
0x1b905:    add     edx, 1
0x1b908:    mov     esi, 0x24
0x1b90d:    mov     edi, 1
0x1b912:    xor     eax, eax
0x1b914:    mov     dword ptr [rsp + 0x6c], edx
0x1b918:    mov     rdx, r14
0x1b91b:    call    0x1b420
0x1b920:    mov     r13, rax
0x1b923:    test    rax, rax
0x1b926:    je      0x1c080
0x1b92c:    lea     rax, [rip + 0x101ed]
0x1b933:    cmp     r13, rax
0x1b936:    je      0x1bdae
0x1b93c:    mov     rdi, r13
0x1b93f:    call    0x49b0
0x1b944:    lea     rdi, [rax + 1]
0x1b948:    call    0x1d670
0x1b94d:    mov     rsi, r13
0x1b950:    mov     rdi, rax
0x1b953:    call    0x48b0
0x1b958:    mov     rdi, r13
0x1b95b:    mov     r14, rax
0x1b95e:    call    0x18d10
0x1b963:    movsxd  rcx, dword ptr [rsp + 0x6c]
0x1b968:    cmp     byte ptr [rbp + rcx], 0x3a
0x1b96d:    mov     dword ptr [rsp + 0x44], ecx
0x1b971:    jne     0x1c3d4
0x1b977:    mov     dword ptr [rsp + 0x18], 0
0x1b97f:    lea     r12, [rip + 0x6afe]
0x1b986:    mov     dword ptr [rsp + 0x38], 0
0x1b98e:    mov     dword ptr [rsp + 0x40], ebx
0x1b992:    mov     qword ptr [rsp + 0x28], r15
0x1b997:    mov     r15, r14
0x1b99a:    mov     r14d, ecx
0x1b99d:    nop     dword ptr [rax]
0x1b9a0:    movzx   eax, byte ptr [rbp + rcx + 1]
0x1b9a5:    cmp     al, 0x67
0x1b9a7:    jne     0x1b9bf

```

```

0x1b9a9:      add     r14d, 1
0x1b9ad:      mov     dword ptr [rsp + 0x18], 1
0x1b9b5:      mov     dword ptr [rsp + 0x6c], r14d
0x1b9ba:      movzx   eax, byte ptr [rbp + rcx + 2]
0x1b9bf:      cmp     al, 0x26
0x1b9c1:      je      0x1be70
0x1b9c7:      sub     eax, 0x65
0x1b9ca:      cmp     al, 0xf
0x1b9cc:      ja      0x1b9e0
0x1b9ce:      movzx   eax, al
0x1b9d1:      movsxd   rax, dword ptr [r12 + rax*4]
0x1b9d5:      add     rax, r12
0x1b9d8:      jmp     rax
0x1b9db:      -----

```

```

0x1b9de:      -----
0x1b9df:      -----
0x1b9e0:      mov     esi, dword ptr [rsp + 0x6c]
0x1b9e4:      mov     r14, r15
0x1b9e7:      mov     r15, qword ptr [rsp + 0x28]
0x1b9ec:      mov     ecx, 3
0x1b9f1:      lea     edx, [rsi + 2]
0x1b9f4:      add     esi, 1
0x1b9f7:      mov     rdi, rbp
0x1b9fa:      xor     eax, eax
0x1b9fc:      call    0x19fe0
0x1ba01:      mov     rdi, qword ptr [rsp + 0x10]
0x1ba06:      mov     rbx, rax
0x1ba09:      call    0x18d10
0x1ba0e:      mov     rdi, r14
0x1ba11:      call    0x18d10
0x1ba16:      mov     rax, qword ptr [rsp + 0x30]
0x1ba1b:      mov     rdi, r15
0x1ba1e:      mov     qword ptr [rax], rbx
0x1ba21:      call    0x18d10
0x1ba26:      mov     eax, 0xffffffff
0x1ba2b:      cmp     qword ptr [rsp + 0x20], rbp
0x1ba30:      je      0x1b67c
0x1ba36:      mov     rdi, rbp
0x1ba39:      mov     dword ptr [rsp + 4], eax
0x1ba3d:      call    0x18d10
0x1ba42:      mov     eax, dword ptr [rsp + 4]
0x1ba46:      jmp     0x1b67c
0x1ba4b:      nop     dword ptr [rax + rax]
0x1ba50:      mov     esi, 0x2f
0x1ba55:      mov     rdi, r15

```

```

0x1ba58:      call    0x4a20
0x1ba5d:      lea     r13, [rax + 1]
0x1ba61:      test   rax, rax
0x1ba64:      je      0x1bb70
0x1ba6a:      mov     rdi, r13
0x1ba6d:      call    0x49b0
0x1ba72:      lea     rdi, [rax + 1]
0x1ba76:      call    0x1d670
0x1ba7b:      mov     rsi, r13
0x1ba7e:      mov     rdi, rax
0x1ba81:      call    0x48b0
0x1ba86:      mov     rdi, r15
0x1ba89:      mov     rbx, rax
0x1ba8c:      call    0x18d10
0x1ba91:      mov     eax, dword ptr [rsp + 0x6c]
0x1ba95:      mov     r15, rbx
0x1ba98:      lea     r14d, [rax + 2]
0x1ba9c:      nop     dword ptr [rax]
0x1baa0:      mov     dword ptr [rsp + 0x6c], r14d
0x1baa5:      movsxd  rcx, r14d
0x1baa8:      cmp     byte ptr [rbp + rcx], 0x3a
0x1baad:      je      0x1b9a0
0x1bab3:      xor     eax, eax
0x1bab5:      cmp     dword ptr [rsp + 0x38], 1
0x1baba:      mov     edx, r14d
0x1babd:      mov     r14, r15
0x1bac0:      sete    al
0x1bac3:      mov     r15, qword ptr [rsp + 0x28]
0x1bac8:      movsxd  rbx, dword ptr [rsp + 0x40]
0x1bacd:      mov     dword ptr [rsp + 0x28], eax
0x1bad1:      sub     edx, 1
0x1bad4:      mov     rdi, r14
0x1bad7:      mov     dword ptr [rsp + 0x6c], edx
0x1badb:      call    0x49b0
0x1bae0:      cmp     eax, 0x80
0x1bae5:      jg      0x1c269
0x1baeb:      mov     r12, qword ptr [rsp + 0x10]
0x1baf0:      mov     rsi, r14
0x1baf3:      mov     rdi, r12
0x1baf6:      call    0x48b0
0x1bafb:      mov     rdi, r14
0x1bafe:      call    0x18d10
0x1bb03:      add     dword ptr [rsp + 0x3c], 1
0x1bb08:      cmp     byte ptr [r12], 0
0x1bb0d:      mov     r13d, dword ptr [rsp + 0x6c]
0x1bb12:      jne     0x1c098
0x1bb18:      mov     rdi, qword ptr [rsp + 0x10]
0x1bb1d:      call    0x18d10

```

```

0x1bb22:      jmp      0x1b74d
0x1bb27:      nop      word ptr [rax + rax]
0x1bb30:      mov      esi, 0x2e
0x1bb35:      mov      rdi, r15
0x1bb38:      call     0x4a20
0x1bb3d:      test     rax, rax
0x1bb40:      je       0x1bb70
0x1bb42:      mov      byte ptr [rax], 0
0x1bb45:      mov      eax, dword ptr [rsp + 0x6c]
0x1bb49:      lea      r14d, [rax + 2]
0x1bb4d:      jmp      0x1baa0
0x1bb52:      nop      word ptr [rax + rax]
0x1bb58:      mov      esi, 0x2f
0x1bb5d:      mov      rdi, r15
0x1bb60:      call     0x4a20
0x1bb65:      test     rax, rax
0x1bb68:      jne      0x1bb42
0x1bb6a:      nop      word ptr [rax + rax]
0x1bb70:      mov      eax, dword ptr [rsp + 0x6c]
0x1bb74:      lea      r14d, [rax + 2]
0x1bb78:      jmp      0x1baa0
0x1bb7d:      nop      dword ptr [rax]
0x1bb80:      mov      esi, 0x2e
0x1bb85:      mov      rdi, r15
0x1bb88:      call     0x4a20
0x1bb8d:      mov      r13, rax
0x1bb90:      test     rax, rax
0x1bb93:      jne      0x1ba6a
0x1bb99:      jmp      0x1bb70
0x1bb9b:      nop      dword ptr [rax + rax]
0x1bba0:      mov      eax, dword ptr [rsp + 0x6c]
0x1bba4:      add      dword ptr [rsp + 0x38], 1
0x1bba9:      lea      r14d, [rax + 2]
0x1bbad:      jmp      0x1baa0
0x1bbb2:      nop      word ptr [rax + rax]
0x1bbb8:      mov      ebx, dword ptr [rsp + 0x6c]
0x1bbbc:      mov      rdi, rbp
0x1bbbf:      lea      r14d, [rbx + 2]
0x1bbc3:      call     0x49b0
0x1bbc8:      cmp      r14d, eax
0x1bbcb:      jge      0x1baa0
0x1bbd1:      movsxd   rax, ebx
0x1bbd4:      lea      r14, [rsp + 0x6c]
0x1bbd9:      add      ebx, 3
0x1bbdc:      xor      ecx, ecx
0x1bbde:      movsx    r13d, byte ptr [rbp + rax + 2]
0x1bbe4:      lea      r8, [rip + 0xff21]
0x1bbeb:      mov      rsi, r14

```

```

0x1bbee:    mov     rdi, rbp
0x1bbf1:    xor     eax, eax
0x1bbf3:    mov     dword ptr [rsp + 0x6c], ebx
0x1bbf7:    mov     edx, r13d
0x1bbfa:    call    0x19c70
0x1bbff:    mov     rbx, rax
0x1bc02:    test    rax, rax
0x1bc05:    je      0x1c387
0x1bc0b:    mov     rdi, qword ptr [rip + 0xff06]
0x1bc12:    test    rdi, rdi
0x1bc15:    je      0x1bc1c
0x1bc17:    call    0x18d10
0x1bc1c:    mov     qword ptr [rip + 0xfef5], rbx
0x1bc23:    mov     eax, dword ptr [rip + 0xfef3]
0x1bc29:    test    eax, eax
0x1bc2b:    je      0x1c400
0x1bc31:    mov     rdi, qword ptr [rip + 0xfed8]
0x1bc38:    test    rdi, rdi
0x1bc3b:    je      0x1bc42
0x1bc3d:    call    0x18d10
0x1bc42:    mov     rsi, r14
0x1bc45:    mov     rdi, rbp
0x1bc48:    xor     eax, eax
0x1bc4a:    mov     ecx, 1
0x1bc4f:    lea     r8, [rip + 0xfef2]
0x1bc56:    mov     edx, r13d
0x1bc59:    call    0x19c70
0x1bc5e:    mov     esi, 0x26
0x1bc63:    mov     rdi, rax
0x1bc66:    mov     qword ptr [rip + 0xfea3], rax
0x1bc6d:    call    0x4a00
0x1bc72:    test    rax, rax
0x1bc75:    je      0x1be75
0x1bc7b:    mov     edx, dword ptr [rip + 0xfe8b]
0x1bc81:    add     edx, dword ptr [rip + 0xfe81]
0x1bc87:    movsxd  rdi, edx
0x1bc8a:    mov     dword ptr [rsp + 0x48], edx
0x1bc8e:    call    0x1d670
0x1bc93:    mov     r9d, dword ptr [rip + 0xfe6e]
0x1bc9a:    mov     r10, qword ptr [rip + 0xfe6f]
0x1bca1:    mov     r8, rax
0x1bca4:    test    r9d, r9d
0x1bca7:    jle     0x1c2d0
0x1bcad:    mov     qword ptr [rsp + 0x50], rbp
0x1bcb2:    xor     ebx, ebx
0x1bcb4:    xor     r13d, r13d
0x1bcb7:    mov     rbp, rax
0x1bcba:    mov     qword ptr [rsp + 0x58], r15

```

```

0x1bcbf:    mov     r15d, dword ptr [rsp + 0x48]
0x1bcc4:    jmp     0x1bd01
0x1bcc6:    nop     word ptr cs:[rax + rax]
0x1bcd0:    cmp     al, 0x5c
0x1bcd2:    je      0x1c300
0x1bcd8:    cmp     r15d, ebx
0x1bcdb:    jle     0x1c320
0x1bce1:    movzx   ecx, byte ptr [r10 + r14]
0x1bce6:    movsxd  rax, ebx
0x1bce9:    add     ebx, 1
0x1bcec:    mov     byte ptr [rbp + rax], cl
0x1bcf0:    add     r13d, 1
0x1bcf4:    cmp     r13d, dword ptr [rip + 0xfe0d]
0x1bcfb:    jge     0x1c36f
0x1bd01:    movsxd  r14, r13d
0x1bd04:    movzx   eax, byte ptr [r10 + r14]
0x1bd09:    cmp     al, 0x26
0x1bd0b:    jne     0x1bcd0
0x1bd0d:    mov     eax, dword ptr [rip + 0xfdf9]
0x1bd13:    lea     ecx, [rax + rbx]
0x1bd16:    cmp     ecx, r15d
0x1bd19:    jge     0x1c33d
0x1bd1f:    movsxd  rdi, ebx
0x1bd22:    mov     rsi, qword ptr [rip + 0xfdef]
0x1bd29:    mov     qword ptr [rsp + 0x48], r10
0x1bd2e:    add     rdi, rbp
0x1bd31:    call    0x48b0
0x1bd36:    mov     r10, qword ptr [rsp + 0x48]
0x1bd3b:    add     ebx, dword ptr [rip + 0xfdc b]
0x1bd41:    jmp     0x1bcf0
0x1bd43:    nop     dword ptr [rax + rax]
0x1bd48:    cmp     al, 0x39
0x1bd4a:    jg      0x1bd80
0x1bd4c:    cmp     al, 0x2f
0x1bd4e:    jg      0x1c150
0x1bd54:    mov     rdi, r14
0x1bd57:    call    0x49b0
0x1bd5c:    lea     rdi, [rax + 1]
0x1bd60:    call    0x1d670
0x1bd65:    mov     rsi, r14
0x1bd68:    mov     rdi, rax
0x1bd6b:    call    0x48b0
0x1bd70:    mov     r14, rax
0x1bd73:    jmp     0x1b963
0x1bd78:    nop     dword ptr [rax + rax]
0x1bd80:    cmp     al, 0x5e
0x1bd82:    jne     0x1bd54
0x1bd84:    add     edx, 1

```

```

0x1bd87:      mov     edi, 1
0x1bd8c:      mov     esi, 1
0x1bd91:      mov     dword ptr [rsp + 0x6c], edx
0x1bd95:      nop     dword ptr [rax]
0x1bd98:      mov     rdx, r14
0x1bd9b:      xor     eax, eax
0x1bd9d:      call    0x1b420
0x1bda2:      mov     r13, rax
0x1bda5:      test    rax, rax
0x1bda8:      jne     0x1b92c
0x1bdae:      mov     r8d, dword ptr [rsp + 0x6c]
0x1bdb3:      mov     ecx, 1
0x1bdb8:      mov     edx, r8d
0x1bdbb:      mov     esi, r12d
0x1bdbe:      mov     rdi, rbp
0x1bdc1:      xor     eax, eax
0x1bdc3:      call    0x19fe0
0x1bdc8:      mov     rdi, qword ptr [rsp + 0x10]
0x1bdcd:      mov     rbx, rax
0x1bdd0:      call    0x18d10
0x1bdd5:      jmp     0x1ba16
0x1bdda:      nop     word ptr [rax + rax]
0x1bde0:      cmp     ebx, dword ptr [rsp + 4]
0x1bde4:      jge     0x1bff0
0x1bdea:      lea     eax, [rbx + 1]
0x1bded:      mov     byte ptr [r15 + rbx], 0x5c
0x1bdf2:      movsxd  rdx, eax
0x1bdf5:      mov     dword ptr [rsp + 8], 1
0x1bdfd:      movsxd  rbx, eax
0x1be00:      mov     byte ptr [r15 + rdx], 0
0x1be05:      jmp     0x1b74d
0x1be0a:      nop     word ptr [rax + rax]
0x1be10:      add     dword ptr [rsp + 4], 0x40
0x1be15:      movsxd  rsi, dword ptr [rsp + 4]
0x1be1a:      mov     rdi, r15
0x1be1d:      call    0x1d690
0x1be22:      mov     r15, rax
0x1be25:      jmp     0x1b778
0x1be2a:      nop     word ptr [rax + rax]
0x1be30:      xor     edx, edx
0x1be32:      test    r13d, r13d
0x1be35:      je      0x1be4d
0x1be37:      movzx   eax, byte ptr [rbp + r14 - 1]
0x1be3d:      cmp     al, 0x27
0x1be3f:      je      0x1c078
0x1be45:      cmp     al, 0x22
0x1be47:      je      0x1c078
0x1be4d:      lea     rsi, [rsp + 0x6c]

```

```

0x1be52:    mov     rdi, rbp
0x1be55:    xor     eax, eax
0x1be57:    call    0x1ac70
0x1be5c:    mov     r12d, dword ptr [rsp + 0x6c]
0x1be61:    mov     r14, rax
0x1be64:    jmp     0x1b8b3
0x1be69:    nop     dword ptr [rax]
0x1be70:    add     dword ptr [rsp + 0x6c], 2
0x1be75:    mov     rdi, r15
0x1be78:    call    0x49b0
0x1be7d:    mov     r13d, dword ptr [rip + 0xfc88]
0x1be84:    cmp     eax, r13d
0x1be87:    jl      0x1c400
0x1be8d:    mov     qword ptr [rsp + 0x48], rbp
0x1be92:    xor     r14d, r14d
0x1be95:    mov     rbp, r15
0x1be98:    mov     r15d, eax
0x1be9b:    mov     dword ptr [rsp + 0x50], 1
0x1bea3:    jmp     0x1beba
0x1bea5:    nop     dword ptr [rax]
0x1bea8:    add     r14d, 1
0x1beac:    lea     eax, [r13 + r14]
0x1beb1:    cmp     eax, r15d
0x1beb4:    jg      0x1c050
0x1beba:    mov     rsi, qword ptr [rip + 0xfc57]
0x1bec1:    movsxd  rbx, r14d
0x1bec4:    lea     rdi, [rbp + rbx]
0x1bec9:    movzx   eax, byte ptr [rsi]
0x1becc:    cmp     byte ptr [rdi], al
0x1bece:    jne     0x1bea8
0x1bed0:    movsxd  rdx, r13d
0x1bed3:    call    0x4890
0x1bed8:    test    eax, eax
0x1beda:    jne     0x1bea8
0x1bedc:    mov     ecx, dword ptr [rip + 0xfc26]
0x1bee2:    add     ecx, r15d
0x1bee5:    sub     ecx, r13d
0x1bee8:    lea     edi, [rcx + 1]
0x1beeb:    mov     dword ptr [rsp + 0x50], ecx
0x1beef:    movsxd  rdi, edi
0x1bef2:    call    0x1d670
0x1bef7:    mov     rdx, rbx
0x1befa:    mov     rsi, rbp
0x1befd:    mov     rdi, rax
0x1bf00:    mov     r13, rax
0x1bf03:    call    0x4880
0x1bf08:    movsxd  rdx, dword ptr [rip + 0xfbf9]
0x1bf0f:    lea     rdi, [r13 + rbx]

```



```

0x1bf14:      mov     rsi, qword ptr [rip + 0xfbf5]
0x1bf1b:      call    0x4880
0x1bf20:      movsxd  rsi, dword ptr [rip + 0xfbe5]
0x1bf27:      movsxd  rdi, dword ptr [rip + 0xfbd5]
0x1bf2e:      lea     edx, [rsi + r14]
0x1bf32:      add     rdi, rbx
0x1bf35:      add     rsi, rbx
0x1bf38:      sub     r15d, edx
0x1bf3b:      add     rsi, rbp
0x1bf3e:      add     rdi, r13
0x1bf41:      movsxd  rdx, r15d
0x1bf44:      call    0x4880
0x1bf49:      movsxd  rcx, dword ptr [rsp + 0x50]
0x1bf4e:      mov     rdi, rbp
0x1bf51:      mov     byte ptr [r13 + rcx], 0
0x1bf57:      call    0x18d10
0x1bf5c:      mov     edi, dword ptr [rsp + 0x18]
0x1bf60:      test    edi, edi
0x1bf62:      jne     0x1c1a8
0x1bf68:      mov     rbp, qword ptr [rsp + 0x48]
0x1bf6d:      mov     r14d, dword ptr [rsp + 0x6c]
0x1bf72:      mov     r15, r13
0x1bf75:      jmp     0x1baa5
0x1bf7a:      nop     word ptr [rax + rax]
0x1bf80:      add     edx, 1
0x1bf83:      mov     esi, 0x24
0x1bf88:      mov     edi, 0x24
0x1bf8d:      xor     eax, eax
0x1bf8f:      mov     dword ptr [rsp + 0x6c], edx
0x1bf93:      mov     rdx, r14
0x1bf96:      call    0x1b420
0x1bf9b:      mov     r13, rax
0x1bf9e:      lea     rax, [rip + 0xf7b]
0x1bfa5:      cmp     r13, rax
0x1bfa8:      je      0x1bdae
0x1bfae:      test    r13, r13
0x1bfb1:      jne     0x1b93c
0x1bfb7:      jmp     0x1bd54
0x1fbfb:      nop     dword ptr [rax]
0x1bfc0:      cmp     r12b, 0x23
0x1bfc4:      jne     0x1be30
0x1bfca:      lea     r12d, [r13 + 2]
0x1bfce:      mov     r14, r15
0x1bfd1:      mov     dword ptr [rsp + 0x6c], r12d
0x1bfd6:      jmp     0x1b8b3
0x1bfdb:      nop     dword ptr [rax + rax]
0x1bfe0:      mov     qword ptr [rsp + 0x10], rdx
0x1bfe5:      jmp     0x1b806

```

```

0x1bfea:      nop      word ptr [rax + rax]
0x1bff0:      add      dword ptr [rsp + 4], 0x40
0x1bff5:      movsxd   rsi, dword ptr [rsp + 4]
0x1bffa:      mov      rdi, r15
0x1bffd:      call     0x1d690
0x1c002:      mov      r15, rax
0x1c005:      jmp      0x1bdea
0x1c00a:      nop      word ptr [rax + rax]
0x1c010:      mov      r13, qword ptr [rip + 0xfad9]
0x1c017:      add      edx, 1
0x1c01a:      mov      dword ptr [rsp + 0x6c], edx
0x1c01e:      test     r13, r13
0x1c021:      je       0x1c080
0x1c023:      mov      rdi, r13
0x1c026:      call     0x49b0
0x1c02b:      lea      rdi, [rax + 1]
0x1c02f:      call     0x1d670
0x1c034:      mov      rsi, r13
0x1c037:      mov      rdi, rax
0x1c03a:      call     0x48b0
0x1c03f:      mov      r13, rax
0x1c042:      jmp      0x1b92c
0x1c047:      nop      word ptr [rax + rax]
0x1c050:      cmp      dword ptr [rsp + 0x18], 1
0x1c055:      mov      r15, rbp
0x1c058:      mov      r14d, dword ptr [rsp + 0x6c]
0x1c05d:      mov      rbp, qword ptr [rsp + 0x48]
0x1c062:      jle      0x1c417
0x1c068:      mov      dword ptr [rsp + 0x18], 0
0x1c070:      jmp      0x1baa5
0x1c075:      nop      dword ptr [rax]
0x1c078:      movsx    edx, al
0x1c07b:      jmp      0x1be4d
0x1c080:      mov      edi, 1
0x1c085:      call     0x1d670
0x1c08a:      mov      byte ptr [rax], 0
0x1c08d:      mov      r13, rax
0x1c090:      jmp      0x1b92c
0x1c095:      nop      dword ptr [rax]
0x1c098:      mov      rdi, qword ptr [rsp + 0x10]
0x1c09d:      call     0x49b0
0x1c0a2:      lea      r12d, [rbx + rax]
0x1c0a6:      mov      eax, dword ptr [rsp + 4]
0x1c0aa:      cmp      r12d, eax
0x1c0ad:      jl       0x1c0e0
0x1c0af:      movsxd   r14, eax
0x1c0b2:      mov      rdi, r15
0x1c0b5:      mov      r15d, eax

```

```

0x1c0b8:      sub     r14, -0x80
0x1c0bc:      nop     dword ptr [rax]
0x1c0c0:      mov     rsi, r14
0x1c0c3:      sub     r15d, -0x80
0x1c0c7:      sub     r14, -0x80
0x1c0cb:      call    0x1d690
0x1c0d0:      mov     rdi, rax
0x1c0d3:      cmp     r12d, r15d
0x1c0d6:      jge     0x1c0c0
0x1c0d8:      mov     dword ptr [rsp + 4], r15d
0x1c0dd:      mov     r15, rax
0x1c0e0:      movsxd   rdi, ebx
0x1c0e3:      mov     rsi, qword ptr [rsp + 0x10]
0x1c0e8:      movsxd   rbx, r12d
0x1c0eb:      add     rdi, r15
0x1c0ee:      call    0x48b0
0x1c0f3:      jmp     0x1bb18
0x1c0f8:      nop     dword ptr [rax + rax]
0x1c100:      xor     edi, edi
0x1c102:      lea     r8d, [rdx + 1]
0x1c106:      movsxd   rax, r8d
0x1c109:      movzx   eax, byte ptr [rbp + rax]
0x1c10e:      lea     ecx, [rax - 0x30]
0x1c111:      cmp     cl, 9
0x1c114:      jbe     0x1c283
0x1c11a:      cmp     al, 0x24
0x1c11c:      je      0x1c3e5
0x1c122:      mov     dword ptr [rsp + 0x6c], r8d
0x1c127:      test    al, al
0x1c129:      je      0x1c3f6
0x1c12f:      cmp     al, 0x3a
0x1c131:      je      0x1c3f6
0x1c137:      test    edi, edi
0x1c139:      jne     0x1bdb3
0x1c13f:      xor     esi, esi
0x1c141:      jmp     0x1bd98
0x1c146:      nop     word ptr cs:[rax + rax]
0x1c150:      test    esi, esi
0x1c152:      je      0x1bd54
0x1c158:      add     edx, 1
0x1c15b:      xor     edi, edi
0x1c15d:      movsxd   rcx, edx
0x1c160:      sub     eax, 0x30
0x1c163:      lea     edx, [rdi + rdi*4]
0x1c166:      movsx   eax, al
0x1c169:      lea     edi, [rax + rdx*2]
0x1c16c:      movzx   eax, byte ptr [rbp + rcx]
0x1c171:      mov     edx, ecx

```

```

0x1c173:    add     rcx, 1
0x1c177:    lea     r8d, [rax - 0x30]
0x1c17b:    cmp     r8b, 9
0x1c17f:    jbe     0x1c160
0x1c181:    cmp     al, 0x5e
0x1c183:    je      0x1c478
0x1c189:    cmp     al, 0x2a
0x1c18b:    je      0x1c250
0x1c191:    cmp     al, 0x2d
0x1c193:    je      0x1c102
0x1c199:    mov     dword ptr [rsp + 0x6c], edx
0x1c19d:    mov     esi, edi
0x1c19f:    jmp     0x1bd98
0x1c1a4:    nop     dword ptr [rax]
0x1c1a8:    mov     rdi, r13
0x1c1ab:    mov     rbp, r13
0x1c1ae:    add     r14d, dword ptr [rip + 0xf953]
0x1c1b5:    call    0x49b0
0x1c1ba:    add     dword ptr [rsp + 0x18], 1
0x1c1bf:    mov     r13d, dword ptr [rip + 0xf946]
0x1c1c6:    mov     dword ptr [rsp + 0x50], 0
0x1c1ce:    mov     r15d, eax
0x1c1d1:    jmp     0x1bea8
0x1c1d6:    nop     word ptr cs:[rax + rax]
0x1c1e0:    lea     edi, [r12 + 5]
0x1c1e5:    movsxd  rdi, edi
0x1c1e8:    call    0x1d670
0x1c1ed:    mov     r11d, 0x733a
0x1c1f3:    mov     rsi, qword ptr [rsp + 0x20]
0x1c1f8:    mov     rbp, rax
0x1c1fb:    movzx   eax, byte ptr [rip + 0xe5eb]
0x1c202:    mov     word ptr [rbp + 2], r11w
0x1c207:    lea     rdi, [rbp + 4]
0x1c20b:    mov     byte ptr [rbp + 1], al
0x1c20e:    mov     byte ptr [rbp], al
0x1c211:    call    0x48b0
0x1c216:    lea     eax, [r12 + 4]
0x1c21b:    mov     dword ptr [rsp + 0xc], eax
0x1c21f:    test    eax, eax
0x1c221:    jg      0x1b6e6
0x1c227:    mov     rax, qword ptr [rsp + 0x30]
0x1c22c:    mov     qword ptr [rax], r15
0x1c22f:    cmp     qword ptr [rsp + 0x20], rbp
0x1c234:    je      0x1b67a
0x1c23a:    mov     rdi, rbp
0x1c23d:    call    0x18d10
0x1c242:    mov     dword ptr [rsp + 0x3c], 0
0x1c24a:    jmp     0x1b7c0

```

```

0x1c24f:      nop
0x1c250:      cmp     al, 0x5e
0x1c252:      je      0x1c478
0x1c258:      add     edx, 1
0x1c25b:      mov     esi, 0x24
0x1c260:      mov     dword ptr [rsp + 0x6c], edx
0x1c264:      jmp     0x1bd98
0x1c269:      add     eax, 1
0x1c26c:      mov     rdi, qword ptr [rsp + 0x10]
0x1c271:      movsxd  rsi, eax
0x1c274:      call    0x1d690
0x1c279:      mov     qword ptr [rsp + 0x10], rax
0x1c27e:      jmp     0x1baeb
0x1c283:      add     edx, 2
0x1c286:      xor     esi, esi
0x1c288:      movsxd  rdx, edx
0x1c28b:      nop     dword ptr [rax + rax]
0x1c290:      sub     eax, 0x30
0x1c293:      lea     ecx, [rsi + rsi*4]
0x1c296:      mov     r8d, edx
0x1c299:      movsx   eax, al
0x1c29c:      lea     esi, [rax + rcx*2]
0x1c29f:      movzx   eax, byte ptr [rbp + rdx]
0x1c2a4:      add     rdx, 1
0x1c2a8:      lea     ecx, [rax - 0x30]
0x1c2ab:      cmp     cl, 9
0x1c2ae:      jbe     0x1c290
0x1c2b0:      cmp     esi, 0x24
0x1c2b3:      sete    al
0x1c2b6:      cmp     esi, edi
0x1c2b8:      setge   dl
0x1c2bb:      or      eax, edx
0x1c2bd:      mov     dword ptr [rsp + 0x6c], r8d
0x1c2c2:      test    al, al
0x1c2c4:      je      0x1bdb3
0x1c2ca:      jmp     0x1bd98
0x1c2cf:      nop
0x1c2d0:      xor     ebx, ebx
0x1c2d2:      mov     byte ptr [rax], 0
0x1c2d5:      mov     rdi, r10
0x1c2d8:      mov     qword ptr [rsp + 0x48], r8
0x1c2dd:      call    0x18d10
0x1c2e2:      mov     r8, qword ptr [rsp + 0x48]
0x1c2e7:      mov     dword ptr [rip + 0xf81b], ebx
0x1c2ed:      mov     qword ptr [rip + 0xf81c], r8
0x1c2f4:      jmp     0x1be75
0x1c2f9:      nop     dword ptr [rax]
0x1c300:      cmp     byte ptr [r10 + r14 + 1], 0x26

```

```

0x1c306:    lea     rax, [r14 + 1]
0x1c30a:    jne     0x1bcd8
0x1c310:    add     r13d, 1
0x1c314:    mov     r14, rax
0x1c317:    jmp     0x1bcd8
0x1c31c:    nop     dword ptr [rax]
0x1c320:    add     r15d, r15d
0x1c323:    mov     rdi, rbp
0x1c326:    movsxd  rsi, r15d
0x1c329:    call    0x1d690
0x1c32e:    mov     r10, qword ptr [rip + 0xf7db]
0x1c335:    mov     rbp, rax
0x1c338:    jmp     0x1bce1
0x1c33d:    lea     r15d, [rax + r15*2]
0x1c341:    mov     rdi, rbp
0x1c344:    movsxd  rsi, r15d
0x1c347:    call    0x1d690
0x1c34c:    mov     r10, qword ptr [rip + 0xf7bd]
0x1c353:    mov     rbp, rax
0x1c356:    jmp     0x1bd1f
0x1c35b:    xor     eax, eax
0x1c35d:    mov     rdi, r15
0x1c360:    call    0x1a380
0x1c365:    mov     eax, 2
0x1c36a:    jmp     0x1b67c
0x1c36f:    mov     r8, rbp
0x1c372:    movsxd  rax, ebx
0x1c375:    mov     r15, qword ptr [rsp + 0x58]
0x1c37a:    mov     rbp, qword ptr [rsp + 0x50]
0x1c37f:    add     rax, r8
0x1c382:    jmp     0x1c2d2
0x1c387:    cmp     qword ptr [rip + 0xf789], 0
0x1c38f:    mov     eax, dword ptr [rip + 0xf777]
0x1c395:    jne     0x1bc29
0x1c39b:    mov     rbx, qword ptr [rip + 0xf74e]
0x1c3a2:    test    rbx, rbx
0x1c3a5:    je      0x1c3b0
0x1c3a7:    cmp     byte ptr [rbx], 0
0x1c3aa:    jne     0x1c442
0x1c3b0:    mov     dword ptr [rip + 0xf752], 0
0x1c3ba:    mov     r14, r15
0x1c3bd:    mov     r15, qword ptr [rsp + 0x28]
0x1c3c2:    mov     edx, dword ptr [rsp + 0x6c]
0x1c3c6:    mov     ecx, 2
0x1c3cb:    mov     esi, dword ptr [rsp + 0x44]
0x1c3cf:    jmp     0x1b9f7
0x1c3d4:    mov     dword ptr [rsp + 0x28], 0
0x1c3dc:    mov     edx, dword ptr [rsp + 0x44]

```

```

0x1c3e0:    jmp     0x1bad1
0x1c3e5:    add     edx, 2
0x1c3e8:    mov     esi, 0x24
0x1c3ed:    mov     dword ptr [rsp + 0x6c], edx
0x1c3f1:    jmp     0x1bd98
0x1c3f6:    mov     esi, 0xffffffff
0x1c3fb:    jmp     0x1bd98
0x1c400:    mov     r14, r15
0x1c403:    mov     r15, qword ptr [rsp + 0x28]
0x1c408:    jmp     0x1c3c2
0x1c40a:    mov     rax, qword ptr [rsp + 0x30]
0x1c40f:    mov     qword ptr [rax], r15
0x1c412:    jmp     0x1b67a
0x1c417:    mov     esi, dword ptr [rsp + 0x50]
0x1c41b:    test    esi, esi
0x1c41d:    je      0x1baa5
0x1c423:    mov     edx, r14d
0x1c426:    mov     ecx, 2
0x1c42b:    mov     r14, r15
0x1c42e:    mov     r15, qword ptr [rsp + 0x28]
0x1c433:    jmp     0x1c3cb
0x1c435:    xor     ecx, ecx
0x1c437:    mov     edx, r12d
0x1c43a:    mov     esi, r13d
0x1c43d:    jmp     0x1bdbbe
0x1c442:    mov     rdi, rbx
0x1c445:    call    0x49b0
0x1c44a:    lea     rdi, [rax + 1]
0x1c44e:    call    0x1d670
0x1c453:    mov     rsi, rbx
0x1c456:    mov     rdi, rax
0x1c459:    call    0x48b0
0x1c45e:    mov     rdi, rax
0x1c461:    mov     qword ptr [rip + 0xf6b0], rax
0x1c468:    call    0x49b0
0x1c46d:    mov     dword ptr [rip + 0xf699], eax
0x1c473:    jmp     0x1bc29
0x1c478:    cmp     edi, 1
0x1c47b:    lea     r8d, [rdx + 1]
0x1c47f:    setle   al
0x1c482:    jmp     0x1c2bd
0x1c487:    call    0x49d0
0x1c48c:    nop     dword ptr [rax]
0x1c490:    endbr64
0x1c494:    push    rbx
0x1c495:    mov     rbx, rdx
0x1c498:    add     rsp, -0x80
0x1c49c:    mov     rax, qword ptr fs:[0x28]

```

```

0x1c4a5:    mov     qword ptr [rsp + 0x78], rax
0x1c4aa:    xor     eax, eax
0x1c4ac:    mov     rsi, rsp
0x1c4af:    call    0x4cb0
0x1c4b4:    test    eax, eax
0x1c4b6:    js      0x1c514
0x1c4b8:    mov     rax, qword ptr [rsp + 8]
0x1c4bd:    test    rax, rax
0x1c4c0:    cmov     rax, qword ptr [rsp]
0x1c4c5:    mov     qword ptr [rbx], rax
0x1c4c8:    mov     rax, qword ptr [rsp + 0x10]
0x1c4cd:    mov     qword ptr [rbx + 8], rax
0x1c4d1:    mov     rax, qword ptr [rsp + 0x18]
0x1c4d6:    mov     qword ptr [rbx + 0x10], rax
0x1c4da:    mov     rax, qword ptr [rsp + 0x20]
0x1c4df:    mov     qword ptr [rbx + 0x18], rax
0x1c4e3:    shr     rax, 0x3f
0x1c4e7:    mov     byte ptr [rbx + 0x20], al
0x1c4ea:    mov     rax, qword ptr [rsp + 0x28]
0x1c4ef:    mov     qword ptr [rbx + 0x28], rax
0x1c4f3:    mov     rax, qword ptr [rsp + 0x30]
0x1c4f8:    mov     qword ptr [rbx + 0x30], rax
0x1c4fc:    xor     eax, eax
0x1c4fe:    mov     rdx, qword ptr [rsp + 0x78]
0x1c503:    sub     rdx, qword ptr fs:[0x28]
0x1c50c:    jne     0x1c51b
0x1c50e:    sub     rsp, -0x80
0x1c512:    pop     rbx
0x1c513:    ret
0x1c514:    mov     eax, 0xffffffff
0x1c519:    jmp     0x1c4fe
0x1c51b:    call    0x49d0
0x1c520:    endbr64
0x1c524:    push    r15
0x1c526:    push    r14
0x1c528:    push    r13
0x1c52a:    mov     r13, r8
0x1c52d:    push    r12
0x1c52f:    mov     r12, rcx
0x1c532:    push    rbp
0x1c533:    push    rbx
0x1c534:    mov     rbx, rdi
0x1c537:    sub     rsp, 0x98
0x1c53e:    mov     qword ptr [rsp], rsi
0x1c542:    mov     qword ptr [rsp + 0x18], r8
0x1c547:    mov     dword ptr [rsp + 0x48], edx
0x1c54b:    mov     rax, qword ptr fs:[0x28]
0x1c554:    mov     qword ptr [rsp + 0x88], rax

```



```

0x1c55c:    mov     eax, edx
0x1c55e:    and     edx, 0x20
0x1c561:    mov     dword ptr [rsp + 0x30], edx
0x1c565:    and     eax, 3
0x1c568:    cmp     edx, 1
0x1c56b:    mov     dword ptr [rsp + 0x10], eax
0x1c56f:    sbb     eax, eax
0x1c571:    and     eax, 0xffffffff8
0x1c574:    add     eax, 0x400
0x1c579:    mov     dword ptr [rsp + 0x34], eax
0x1c57d:    call    0x4900
0x1c582:    mov     r15, qword ptr [rax]
0x1c585:    mov     r14, rax
0x1c588:    mov     rdi, r15
0x1c58b:    call    0x49b0
0x1c590:    mov     rcx, qword ptr [r14 + 0x10]
0x1c594:    mov     r14, qword ptr [r14 + 8]
0x1c598:    mov     rbp, rax
0x1c59b:    sub     rax, 1
0x1c59f:    cmp     rax, 0x10
0x1c5a3:    mov     eax, 1
0x1c5a8:    mov     rdi, r14
0x1c5ab:    mov     qword ptr [rsp + 0x38], rcx
0x1c5b0:    cmovae  rbp, rax
0x1c5b4:    lea     rax, [rip + 0x5123]
0x1c5bb:    cmovae  r15, rax
0x1c5bf:    call    0x49b0
0x1c5c4:    cmp     rax, 0x10
0x1c5c8:    lea     rax, [rip + 0x5e16]
0x1c5cf:    cmova   r14, rax
0x1c5d3:    mov     rax, qword ptr [rsp]
0x1c5d7:    add     rax, 0x288
0x1c5dd:    mov     qword ptr [rsp + 8], rax
0x1c5e2:    cmp     r13, r12
0x1c5e5:    ja      0x1c740
0x1c5eb:    mov     rax, r12
0x1c5ee:    xor     edx, edx
0x1c5f0:    div     r13
0x1c5f3:    test    rdx, rdx
0x1c5f6:    jne     0x1c604
0x1c5f8:    mul     rbx
0x1c5fb:    mov     rcx, rax
0x1c5fe:    jno     0x1cb98
0x1c604:    mov     qword ptr [rsp + 0x20], r12
0x1c609:    fild    qword ptr [rsp + 0x20]
0x1c60d:    test    r12, r12
0x1c610:    js      0x1c928
0x1c616:    mov     rax, qword ptr [rsp + 0x18]

```

```

0x1c61b:    mov     qword ptr [rsp + 0x20], rax
0x1c620:    fild    qword ptr [rsp + 0x20]
0x1c624:    test    rax, rax
0x1c627:    js      0x1c908
0x1c62d:    mov     qword ptr [rsp + 0x20], rbx
0x1c632:    fdivp    st(1)
0x1c634:    fild    qword ptr [rsp + 0x20]
0x1c638:    test    rbx, rbx
0x1c63b:    js      0x1c8f0
0x1c641:    fmulp    st(1)
0x1c643:    test    byte ptr [rsp + 0x48], 0x10
0x1c648:    je      0x1c938
0x1c64e:    fild    dword ptr [rsp + 0x34]
0x1c652:    xor     ebx, ebx
0x1c654:    fld     st(0)
0x1c656:    jmp     0x1c664
0x1c658:    nop     dword ptr [rax + rax]
0x1c660:    fstp    st(1)
0x1c662:    fxch    st(2)
0x1c664:    fld     st(0)
0x1c666:    add     ebx, 1
0x1c669:    fmul    st(2)
0x1c66b:    fxch    st(3)
0x1c66d:    fcomi    st(3)
0x1c66f:    jb      0x1c680
0x1c671:    cmp     ebx, 8
0x1c674:    jne     0x1c660
0x1c676:    fstp    st(2)
0x1c678:    fstp    st(2)
0x1c67a:    jmp     0x1c684
0x1c67c:    nop     dword ptr [rax]
0x1c680:    fstp    st(2)
0x1c682:    fstp    st(2)
0x1c684:    lea     r15, [rbp + 1]
0x1c688:    fdivrp   st(1)
0x1c68a:    cmp     dword ptr [rsp + 0x30], 1
0x1c68f:    adc     rbp, 2
0x1c693:    cmp     dword ptr [rsp + 0x10], 1
0x1c698:    je      0x1cc70
0x1c69e:    fld     xword ptr [rip + 0x5e9c]
0x1c6a4:    fcomip   st(1)
0x1c6a6:    jbe     0x1cb10
0x1c6ac:    fld     dword ptr [rip + 0x5e7e]
0x1c6b2:    fxch    st(1)
0x1c6b4:    fcomi    st(1)
0x1c6b6:    jae     0x1ce00
0x1c6bc:    fstp    st(1)
0x1c6be:    fnstcw  word ptr [rsp + 0x4e]

```

```

0x1c6c2:    movzx    eax, word ptr [rsp + 0x4e]
0x1c6c7:    or      ah, 0xc
0x1c6ca:    mov     word ptr [rsp + 0x4c], ax
0x1c6cf:    fld     st(0)
0x1c6d1:    fldcw   word ptr [rsp + 0x4c]
0x1c6d5:    fistp   qword ptr [rsp + 0x20]
0x1c6d9:    fldcw   word ptr [rsp + 0x4e]
0x1c6dd:    mov     rax, qword ptr [rsp + 0x20]
0x1c6e2:    mov     qword ptr [rsp + 0x20], rax
0x1c6e7:    fild    qword ptr [rsp + 0x20]
0x1c6eb:    test    rax, rax
0x1c6ee:    jns     0x1c6f6
0x1c6f0:    fadd    dword ptr [rip + 0x5e36]
0x1c6f6:    mov     ecx, dword ptr [rsp + 0x10]
0x1c6fa:    test    ecx, ecx
0x1c6fc:    jne     0x1cb18
0x1c702:    fxch    st(1)
0x1c704:    fucomi  st(1)
0x1c706:    jp      0x1c712
0x1c708:    je      0x1cb20
0x1c70e:    fstp    st(1)
0x1c710:    jmp     0x1c718
0x1c712:    fstp    st(1)
0x1c714:    nop     dword ptr [rax]
0x1c718:    add     rax, 1
0x1c71c:    mov     qword ptr [rsp + 0x20], rax
0x1c721:    fild    qword ptr [rsp + 0x20]
0x1c725:    jns     0x1cb1c
0x1c72b:    fadd    dword ptr [rip + 0x5dfb]
0x1c731:    fxch    st(1)
0x1c733:    jmp     0x1cb20
0x1c738:    nop     dword ptr [rax + rax]
0x1c740:    test    r12, r12
0x1c743:    je      0x1c604
0x1c749:    mov     rax, qword ptr [rsp + 0x18]
0x1c74e:    xor     edx, edx
0x1c750:    div     r12
0x1c753:    mov     r8, rax
0x1c756:    test    rdx, rdx
0x1c759:    jne     0x1c604
0x1c75f:    mov     rax, rbx
0x1c762:    xor     edx, edx
0x1c764:    div     r8
0x1c767:    mov     rcx, rax
0x1c76a:    lea     rax, [rdx + rdx*4]
0x1c76e:    xor     edx, edx
0x1c770:    add     rax, rax
0x1c773:    div     r8

```

```

0x1c776:      add     rdx, rdx
0x1c779:      mov     edi, eax
0x1c77b:      cmp     r8, rdx
0x1c77e:      jbe     0x1cde8
0x1c784:      xor     esi, esi
0x1c786:      test    rdx, rdx
0x1c789:      setne   sil
0x1c78d:      mov     r10d, dword ptr [rsp + 0x48]
0x1c792:      and     r10d, 0x10
0x1c796:      je      0x1cbab
0x1c79c:      mov     r8d, dword ptr [rsp + 0x34]
0x1c7a1:      xor     ebx, ebx
0x1c7a3:      mov     r11, r8
0x1c7a6:      cmp     r8, rcx
0x1c7a9:      jbe     0x1cacf
0x1c7af:      cmp     dword ptr [rsp + 0x10], 1
0x1c7b4:      mov     r8, qword ptr [rsp + 8]
0x1c7b9:      je      0x1cdc0
0x1c7bf:      mov     r11d, dword ptr [rsp + 0x10]
0x1c7c4:      test    r11d, r11d
0x1c7c7:      jne     0x1c7e8
0x1c7c9:      add     esi, edi
0x1c7cb:      test    esi, esi
0x1c7cd:      jle     0x1c7e8
0x1c7cf:      add     rcx, 1
0x1c7d3:      test    r10d, r10d
0x1c7d6:      je      0x1c7e8
0x1c7d8:      mov     eax, dword ptr [rsp + 0x34]
0x1c7dc:      cmp     rax, rcx
0x1c7df:      je      0x1cec2
0x1c7e5:      nop     dword ptr [rax]
0x1c7e8:      movabs  rsi, 0xffffffffffffffff
0x1c7f2:      mov     rbp, r8
0x1c7f5:      nop     dword ptr [rax]
0x1c7f8:      mov     rax, rcx
0x1c7fb:      sub     rbp, 1
0x1c7ff:      mul     rsi
0x1c802:      mov     rax, rcx
0x1c805:      shr     rdx, 3
0x1c809:      lea     rdi, [rdx + rdx*4]
0x1c80d:      add     rdi, rdi
0x1c810:      sub     rax, rdi
0x1c813:      add     eax, 0x30
0x1c816:      mov     byte ptr [rbp], al
0x1c819:      mov     rax, rcx
0x1c81c:      mov     rcx, rdx
0x1c81f:      cmp     rax, 9
0x1c823:      ja      0x1c7f8

```

```

0x1c825:    test    byte ptr [rsp + 0x48], 4
0x1c82a:    je      0x1ca2c
0x1c830:    mov     rdi, r14
0x1c833:    mov     r12, r8
0x1c836:    mov     qword ptr [rsp + 0x40], r8
0x1c83b:    mov     r13, -1
0x1c842:    call    0x49b0
0x1c847:    sub     r12, rbp
0x1c84a:    mov     rsi, rbp
0x1c84d:    mov     ecx, 0x29
0x1c852:    mov     r15, rax
0x1c855:    lea     rax, [rsp + 0x50]
0x1c85a:    mov     rdx, r12
0x1c85d:    mov     rdi, rax
0x1c860:    mov     qword ptr [rsp + 0x10], rax
0x1c865:    call    0x4b70
0x1c86a:    mov     dword ptr [rsp + 0x20], ebx
0x1c86e:    mov     rbp, qword ptr [rsp + 0x40]
0x1c873:    mov     rbx, r12
0x1c876:    mov     r12, qword ptr [rsp + 0x38]
0x1c87b:    jmp     0x1c8bb
0x1c87d:    nop     dword ptr [rax]
0x1c880:    cmp     r13, rbx
0x1c883:    mov     rax, qword ptr [rsp + 0x10]
0x1c888:    cmova   r13, rbx
0x1c88c:    sub     rbx, r13
0x1c88f:    lea     rsi, [rax + rbx]
0x1c893:    sub     rbp, r13
0x1c896:    mov     rdx, r13
0x1c899:    mov     rdi, rbp
0x1c89c:    call    0x4bb0
0x1c8a1:    test    rbx, rbx
0x1c8a4:    je      0x1caa0
0x1c8aa:    sub     rbp, r15
0x1c8ad:    mov     rdx, r15
0x1c8b0:    mov     rsi, r14
0x1c8b3:    mov     rdi, rbp
0x1c8b6:    call    0x4bb0
0x1c8bb:    movzx   eax, byte ptr [r12]
0x1c8c0:    test    al, al
0x1c8c2:    je      0x1c880
0x1c8c4:    cmp     al, 0x7e
0x1c8c6:    ja      0x1ca88
0x1c8cc:    cmp     rax, rbx
0x1c8cf:    cmova   rax, rbx
0x1c8d3:    sub     rbx, rax
0x1c8d6:    mov     r13, rax
0x1c8d9:    mov     rax, qword ptr [rsp + 0x10]

```

```

0x1c8de:    lea     rsi, [rax + rbx]
0x1c8e2:    add     r12, 1
0x1c8e6:    jmp     0x1c893
0x1c8e8:    nop     dword ptr [rax + rax]
0x1c8f0:    fadd    dword ptr [rip + 0x5c36]
0x1c8f6:    fmulp   st(1)
0x1c8f8:    test    byte ptr [rsp + 0x48], 0x10
0x1c8fd:    jne     0x1c64e
0x1c903:    jmp     0x1c938
0x1c905:    nop     dword ptr [rax]
0x1c908:    fadd    dword ptr [rip + 0x5c1e]
0x1c90e:    mov     qword ptr [rsp + 0x20], rbx
0x1c913:    fdivp   st(1)
0x1c915:    fild    qword ptr [rsp + 0x20]
0x1c919:    test    rbx, rbx
0x1c91c:    jns     0x1c641
0x1c922:    jmp     0x1c8f0
0x1c924:    nop     dword ptr [rax]
0x1c928:    fadd    dword ptr [rip + 0x5bfe]
0x1c92e:    jmp     0x1c616
0x1c933:    nop     dword ptr [rax + rax]
0x1c938:    cmp     dword ptr [rsp + 0x10], 1
0x1c93d:    je      0x1c9ca
0x1c943:    fld     xword ptr [rip + 0x5bf7]
0x1c949:    fcomip  st(1)
0x1c94b:    jbe     0x1c9ca
0x1c94d:    fld     dword ptr [rip + 0x5bdd]
0x1c953:    fxch    st(1)
0x1c955:    fcomi   st(1)
0x1c957:    jae     0x1ce60
0x1c95d:    fstp    st(1)
0x1c95f:    fnstcw  word ptr [rsp + 0x4e]
0x1c963:    movzx   eax, word ptr [rsp + 0x4e]
0x1c968:    or      ah, 0xc
0x1c96b:    mov     word ptr [rsp + 0x4c], ax
0x1c970:    fld     st(0)
0x1c972:    fldcw   word ptr [rsp + 0x4c]
0x1c976:    fistp   qword ptr [rsp + 0x20]
0x1c97a:    fldcw   word ptr [rsp + 0x4e]
0x1c97e:    mov     rax, qword ptr [rsp + 0x20]
0x1c983:    mov     qword ptr [rsp + 0x20], rax
0x1c988:    fild    qword ptr [rsp + 0x20]
0x1c98c:    test    rax, rax
0x1c98f:    jns     0x1c997
0x1c991:    fadd    dword ptr [rip + 0x5b95]
0x1c997:    mov     r8d, dword ptr [rsp + 0x10]
0x1c99c:    test    r8d, r8d
0x1c99f:    jne     0x1c9c8

```

```

0x1c9a1:    fxch    st(1)
0x1c9a3:    fucomip st(1)
0x1c9a5:    jp      0x1c9ad
0x1c9a7:    je      0x1c9ca
0x1c9a9:    fstp    st(0)
0x1c9ab:    jmp     0x1c9af
0x1c9ad:    fstp    st(0)
0x1c9af:    add     rax, 1
0x1c9b3:    mov     qword ptr [rsp + 0x10], rax
0x1c9b8:    fild    qword ptr [rsp + 0x10]
0x1c9bc:    js      0x1cee7
0x1c9c2:    jmp     0x1c9ca
0x1c9c4:    nop     dword ptr [rax]
0x1c9c8:    fstp    st(1)
0x1c9ca:    sub     rsp, 0x10
0x1c9ce:    mov     rdx, -1
0x1c9d5:    mov     esi, 1
0x1c9da:    xor     eax, eax
0x1c9dc:    mov     rbx, qword ptr [rsp + 0x10]
0x1c9e1:    lea     rcx, [rip + 0x5adc]
0x1c9e8:    fstp    xword ptr [rsp]
0x1c9eb:    mov     rdi, rbx
0x1c9ee:    call    0x4f10
0x1c9f3:    mov     rdi, rbx
0x1c9f6:    mov     ebx, 0xffffffff
0x1c9fb:    call    0x49b0
0x1ca00:    pop     rsi
0x1ca01:    pop     rdi
0x1ca02:    mov     rdx, rax
0x1ca05:    mov     r12, rax
0x1ca08:    mov     rbp, qword ptr [rsp + 8]
0x1ca0d:    mov     rsi, qword ptr [rsp]
0x1ca11:    sub     rbp, rdx
0x1ca14:    mov     rdi, rbp
0x1ca17:    call    0x4d40
0x1ca1c:    lea     r8, [rbp + r12]
0x1ca21:    test    byte ptr [rsp + 0x48], 4
0x1ca26:    jne     0x1c830
0x1ca2c:    test    byte ptr [rsp + 0x48], 0x80
0x1ca31:    je      0x1ca4f
0x1ca33:    cmp     ebx, -1
0x1ca36:    je      0x1cbb8
0x1ca3c:    mov     eax, dword ptr [rsp + 0x48]
0x1ca40:    and     eax, 0x100
0x1ca45:    mov     esi, eax
0x1ca47:    or      esi, ebx
0x1ca49:    jne     0x1cd08
0x1ca4f:    mov     rax, qword ptr [rsp + 8]

```

```

0x1ca54:    mov     byte ptr [rax], 0
0x1ca57:    mov     rax, qword ptr [rsp + 0x88]
0x1ca5f:    sub     rax, qword ptr fs:[0x28]
0x1ca68:    jne     0x1d0eb
0x1ca6e:    add     rsp, 0x98
0x1ca75:    mov     rax, rbp
0x1ca78:    pop     rbx
0x1ca79:    pop     rbp
0x1ca7a:    pop     r12
0x1ca7c:    pop     r13
0x1ca7e:    pop     r14
0x1ca80:    pop     r15
0x1ca82:    ret
0x1ca83:    nop     dword ptr [rax + rax]
0x1ca88:    mov     r13, rbx
0x1ca8b:    mov     rsi, qword ptr [rsp + 0x10]
0x1ca90:    xor     ebx, ebx
0x1ca92:    jmp     0x1c8e2
0x1ca97:    nop     word ptr [rax + rax]
0x1caa0:    mov     ebx, dword ptr [rsp + 0x20]
0x1caa4:    jmp     0x1ca2c
0x1caa6:    nop     word ptr cs:[rax + rax]
0x1cab0:    test    esi, esi
0x1cab2:    setne   sil
0x1cab6:    movzx   esi, sil
0x1caba:    add     ebx, 1
0x1cabd:    cmp     r8, r9
0x1cac0:    ja      0x1cf18
0x1cac6:    cmp     ebx, 8
0x1cac9:    je      0x1c7af
0x1cacf:    mov     rax, rcx
0x1cad2:    xor     edx, edx
0x1cad4:    mov     ecx, esi
0x1cad6:    div     r8
0x1cad9:    sar     ecx, 1
0x1cadb:    mov     r9, rax
0x1cade:    lea     eax, [rdx + rdx*4]
0x1cae1:    xor     edx, edx
0x1cae3:    lea     eax, [rdi + rax*2]
0x1cae6:    div     r11d
0x1cae9:    lea     edx, [rcx + rdx*2]
0x1caec:    mov     edi, eax
0x1caee:    mov     rcx, r9
0x1caf1:    add     esi, edx
0x1caf3:    cmp     r11d, edx
0x1caf6:    ja      0x1cab0
0x1caf8:    cmp     r11d, esi
0x1cafb:    setb    sil

```



```

0x1caff:    movzx    esi, sil
0x1cb03:    add     esi, 2
0x1cb06:    jmp     0x1caba
0x1cb08:    nop     dword ptr [rax + rax]
0x1cb10:    fld     st(0)
0x1cb12:    jmp     0x1cb20
0x1cb14:    nop     dword ptr [rax]
0x1cb18:    fxch    st(1)
0x1cb1a:    jmp     0x1cb20
0x1cb1c:    fxch    st(1)
0x1cb1e:    nop
0x1cb20:    fstp    xword ptr [rsp + 0x20]
0x1cb24:    sub     rsp, 0x10
0x1cb28:    mov     esi, 1
0x1cb2d:    xor     eax, eax
0x1cb2f:    mov     r12, qword ptr [rsp + 0x10]
0x1cb34:    lea     rcx, [rip + 0x598f]
0x1cb3b:    mov     rdx, -1
0x1cb42:    mov     rdi, r12
0x1cb45:    fstp    xword ptr [rsp]
0x1cb48:    call    0x4f10
0x1cb4d:    mov     rdi, r12
0x1cb50:    call    0x49b0
0x1cb55:    mov     rdx, rax
0x1cb58:    pop     rax
0x1cb59:    pop     rcx
0x1cb5a:    fld     xword ptr [rsp + 0x20]
0x1cb5e:    cmp     rdx, rbp
0x1cb61:    ja      0x1cd20
0x1cb67:    test    byte ptr [rsp + 0x48], 8
0x1cb6c:    je      0x1cb88
0x1cb6e:    mov     rax, qword ptr [rsp]
0x1cb72:    cmp     byte ptr [rax + rdx - 1], 0x30
0x1cb77:    je      0x1ce40
0x1cb7d:    fstp    st(0)
0x1cb7f:    jmp     0x1cb8a
0x1cb81:    nop     dword ptr [rax]
0x1cb88:    fstp    st(0)
0x1cb8a:    mov     r12, rdx
0x1cb8d:    sub     r12, r15
0x1cb90:    jmp     0x1ca08
0x1cb95:    nop     dword ptr [rax]
0x1cb98:    mov     r10d, dword ptr [rsp + 0x48]
0x1cb9d:    xor     esi, esi
0x1cb9f:    xor     edi, edi
0x1cba1:    and     r10d, 0x10
0x1cba5:    jne     0x1c79c
0x1cbab:    mov     ebx, 0xffffffff

```

```

0x1cbb0:      jmp      0x1c7af
0x1cbb5:      nop      dword ptr [rax]
0x1cbb8:      mov      rcx, qword ptr [rsp + 0x18]
0x1cbbd:      cmp      rcx, 1
0x1cbc1:      jbe      0x1cfca
0x1cbc7:      mov      edx, dword ptr [rsp + 0x34]
0x1cbcb:      mov      ebx, 1
0x1cbd0:      mov      eax, 1
0x1cbd5:      nop      dword ptr [rax]
0x1cbd8:      imul     rax, rdx
0x1cbdc:      cmp      rcx, rax
0x1cbdf:      jbe      0x1cbe9
0x1cbe1:      add      ebx, 1
0x1cbe4:      cmp      ebx, 8
0x1cbe7:      jne      0x1cbd8
0x1cbe9:      mov      esi, dword ptr [rsp + 0x48]
0x1cbed:      mov      eax, esi
0x1cbef:      and      eax, 0x100
0x1cbf4:      and      esi, 0x40
0x1cbf7:      je       0x1cc18
0x1cbf9:      mov      rsi, qword ptr [rsp]
0x1cbfd:      lea      rdi, [rsi + 0x289]
0x1cc04:      mov      byte ptr [rsi + 0x288], 0x20
0x1cc0b:      mov      qword ptr [rsp + 8], rdi
0x1cc10:      test     ebx, ebx
0x1cc12:      je       0x1d105
0x1cc18:      mov      rsi, qword ptr [rsp + 8]
0x1cc1d:      mov      r9d, dword ptr [rsp + 0x30]
0x1cc22:      lea      rdx, [rsi + 1]
0x1cc26:      test     r9d, r9d
0x1cc29:      jne      0x1cc34
0x1cc2b:      cmp      ebx, 1
0x1cc2e:      je       0x1cf00
0x1cc34:      movsxd   rbx, ebx
0x1cc37:      lea      rcx, [rip + 0x58e2]
0x1cc3e:      movzx    ecx, byte ptr [rcx + rbx]
0x1cc42:      mov      byte ptr [rsi], cl
0x1cc44:      test     eax, eax
0x1cc46:      je       0x1cf0b
0x1cc4c:      mov      r8d, dword ptr [rsp + 0x30]
0x1cc51:      test     r8d, r8d
0x1cc54:      jne      0x1ce30
0x1cc5a:      lea      rax, [rdx + 1]
0x1cc5e:      mov      byte ptr [rdx], 0x42
0x1cc61:      mov      qword ptr [rsp + 8], rax
0x1cc66:      jmp      0x1ca4f
0x1cc6b:      nop      dword ptr [rax + rax]
0x1cc70:      sub      rsp, 0x10

```

```

0x1cc74:      mov     rdx, -1
0x1cc7b:      mov     esi, 1
0x1cc80:      xor     eax, eax
0x1cc82:      fld     st(0)
0x1cc84:      fstp    xword ptr [rsp]
0x1cc87:      mov     r12, qword ptr [rsp + 0x10]
0x1cc8c:      lea     rcx, [rip + 0x5837]
0x1cc93:      mov     rdi, r12
0x1cc96:      fstp    xword ptr [rsp + 0x30]
0x1cc9a:      call    0x4f10
0x1cc9f:      mov     rdi, r12
0x1cca2:      call    0x49b0
0x1cca7:      pop     rsi
0x1cca8:      pop     rdi
0x1cca9:      fld     xword ptr [rsp + 0x20]
0x1ccad:      cmp     rax, rbp
0x1ccb0:      mov     rdx, rax
0x1ccb3:      jbe     0x1cb67
0x1ccb9:      fld     dword ptr [rip + 0x5875]
0x1ccbfb:     fmul    st(1), st(0)
0x1ccc1:     fdivp   st(1)
0x1ccc3:     sub     rsp, 0x10
0x1ccc7:     mov     rdx, -1
0x1ccce:     xor     eax, eax
0x1ccd0:     mov     r15, qword ptr [rsp + 0x10]
0x1ccd5:     lea     rcx, [rip + 0x57e8]
0x1ccdc:     mov     esi, 1
0x1cce1:     mov     rdi, r15
0x1cce4:     fstp    xword ptr [rsp]
0x1cce7:     call    0x4f10
0x1ccecb:     mov     rdi, r15
0x1ccf4:     call    0x49b0
0x1ccf4:     pop     r13
0x1ccf6:     pop     r15
0x1ccf8:     mov     rdx, rax
0x1ccfb:     mov     r12, rax
0x1ccfe:     jmp     0x1ca08
0x1cd03:     nop     dword ptr [rax + rax]
0x1cd08:     test    byte ptr [rsp + 0x48], 0x40
0x1cd0d:     je      0x1cc10
0x1cd13:     jmp     0x1cbf9
0x1cd18:     nop     dword ptr [rax + rax]
0x1cd20:     fld     dword ptr [rip + 0x580e]
0x1cd26:     fmul    st(1), st(0)
0x1cd28:     fld     xword ptr [rip + 0x5812]
0x1cd2e:     fcomip  st(2)
0x1cd30:     jbe     0x1ccc1
0x1cd32:     fld     dword ptr [rip + 0x57f8]

```

```

0x1cd38:      fxch      st(2)
0x1cd3a:      fcomi     st(2)
0x1cd3c:      jae       0x1ce90
0x1cd42:      fstp      st(2)
0x1cd44:      fxch      st(1)
0x1cd46:      fnstcw    word ptr [rsp + 0x4e]
0x1cd4a:      movzx     eax, word ptr [rsp + 0x4e]
0x1cd4f:      or        ah, 0xc
0x1cd52:      mov       word ptr [rsp + 0x4c], ax
0x1cd57:      fld        st(0)
0x1cd59:      fldcw     word ptr [rsp + 0x4c]
0x1cd5d:      fistp     qword ptr [rsp + 0x20]
0x1cd61:      fldcw     word ptr [rsp + 0x4e]
0x1cd65:      mov       rax, qword ptr [rsp + 0x20]
0x1cd6a:      mov       qword ptr [rsp + 0x20], rax
0x1cd6f:      fild      qword ptr [rsp + 0x20]
0x1cd73:      test      rax, rax
0x1cd76:      jns        0x1cd7e
0x1cd78:      fadd      dword ptr [rip + 0x57ae]
0x1cd7e:      mov       edx, dword ptr [rsp + 0x10]
0x1cd82:      test      edx, edx
0x1cd84:      jne        0x1cdb0
0x1cd86:      fucomi     st(1)
0x1cd88:      fstp      st(1)
0x1cd8a:      jp         0x1cd92
0x1cd8c:      je         0x1cdb2
0x1cd8e:      fstp      st(0)
0x1cd90:      jmp        0x1cd94
0x1cd92:      fstp      st(0)
0x1cd94:      add       rax, 1
0x1cd98:      mov       qword ptr [rsp + 0x10], rax
0x1cd9d:      fild      qword ptr [rsp + 0x10]
0x1cda1:      js         0x1cef2
0x1cda7:      jmp        0x1cdb2
0x1cda9:      nop       dword ptr [rax]
0x1cdb0:      fstp      st(1)
0x1cdb2:      fxch      st(1)
0x1cdb4:      jmp        0x1ccc1
0x1cdb9:      nop       dword ptr [rax]
0x1cdc0:      mov       rax, rcx
0x1cdc3:      movsxd    rsi, esi
0x1cdc6:      and       eax, 1
0x1cdc9:      add       rax, rsi
0x1cdcc:      setne     al
0x1cdcf:      movzx     eax, al
0x1cdd2:      add       eax, edi
0x1cdd4:      cmp       eax, 5
0x1cdd7:      jg         0x1c7cf

```

```

0x1cddd:      jmp      0x1c7e8
0x1cde2:      nop      word ptr [rax + rax]
0x1cde8:      mov      esi, 2
0x1cded:      mov      eax, 3
0x1cdf2:      cmovb    esi, eax
0x1cdf5:      jmp      0x1c78d
0x1cdfa:      nop      word ptr [rax + rax]
0x1ce00:      fnstcw   word ptr [rsp + 0x4e]
0x1ce04:      fsubr     st(1), st(0)
0x1ce06:      fxch      st(1)
0x1ce08:      movzx     eax, word ptr [rsp + 0x4e]
0x1ce0d:      or        ah, 0xc
0x1ce10:      mov      word ptr [rsp + 0x4c], ax
0x1ce15:      fldcw     word ptr [rsp + 0x4c]
0x1ce19:      fistp     qword ptr [rsp + 0x20]
0x1ce1d:      fldcw     word ptr [rsp + 0x4e]
0x1ce21:      mov      rax, qword ptr [rsp + 0x20]
0x1ce26:      btc       rax, 0x3f
0x1ce2b:      jmp      0x1c6e2
0x1ce30:      mov      byte ptr [rsi + 1], 0x69
0x1ce34:      lea      rdx, [rsi + 2]
0x1ce38:      jmp      0x1cc5a
0x1ce3d:      nop      dword ptr [rax]
0x1ce40:      fld       dword ptr [rip + 0x56ee]
0x1ce46:      cmp      dword ptr [rsp + 0x10], 1
0x1ce4b:      fmul      st(1), st(0)
0x1ce4d:      jne      0x1cd28
0x1ce53:      jmp      0x1ccc1
0x1ce58:      nop      dword ptr [rax + rax]
0x1ce60:      fnstcw   word ptr [rsp + 0x4e]
0x1ce64:      fsubr     st(1), st(0)
0x1ce66:      fxch      st(1)
0x1ce68:      movzx     eax, word ptr [rsp + 0x4e]
0x1ce6d:      or        ah, 0xc
0x1ce70:      mov      word ptr [rsp + 0x4c], ax
0x1ce75:      fldcw     word ptr [rsp + 0x4c]
0x1ce79:      fistp     qword ptr [rsp + 0x20]
0x1ce7d:      fldcw     word ptr [rsp + 0x4e]
0x1ce81:      mov      rax, qword ptr [rsp + 0x20]
0x1ce86:      btc       rax, 0x3f
0x1ce8b:      jmp      0x1c983
0x1ce90:      fnstcw   word ptr [rsp + 0x4e]
0x1ce94:      fsubr     st(2), st(0)
0x1ce96:      fxch      st(2)
0x1ce98:      movzx     eax, word ptr [rsp + 0x4e]
0x1ce9d:      or        ah, 0xc
0x1cea0:      mov      word ptr [rsp + 0x4c], ax
0x1cea5:      fldcw     word ptr [rsp + 0x4c]

```

```

0x1cea9:      fistp    qword ptr [rsp + 0x20]
0x1cead:      fldcw   word ptr [rsp + 0x4e]
0x1ceb1:      fxch    st(1)
0x1ceb3:      mov     rax, qword ptr [rsp + 0x20]
0x1ceb8:      btc     rax, 0x3f
0x1cebd:      jmp     0x1cd6a
0x1cec2:      cmp     ebx, 8
0x1cec5:      je      0x1c7e8
0x1cecb:      add     ebx, 1
0x1cece:      test    byte ptr [rsp + 0x48], 8
0x1ced3:      je      0x1d039
0x1ced9:      mov     byte ptr [r8 - 1], 0x31
0x1cede:      lea     rbp, [r8 - 1]
0x1cee2:      jmp     0x1ca21
0x1cee7:      fadd    dword ptr [rip + 0x563f]
0x1ceed:      jmp     0x1c9ca
0x1cef2:      fadd    dword ptr [rip + 0x5634]
0x1cef8:      fxch    st(1)
0x1cefa:      jmp     0x1ccc1
0x1ceff:      nop
0x1cf00:      mov     byte ptr [rsi], 0x6b
0x1cf03:      test    eax, eax
0x1cf05:      jne     0x1cc5a
0x1cf0b:      mov     qword ptr [rsp + 8], rdx
0x1cf10:      jmp     0x1ca4f
0x1cf15:      nop     dword ptr [rax]
0x1cf18:      cmp     r9, 9
0x1cf1c:      ja      0x1c7af
0x1cf22:      cmp     dword ptr [rsp + 0x10], 1
0x1cf27:      je      0x1cfa0
0x1cf29:      mov     r12d, dword ptr [rsp + 0x10]
0x1cf2e:      test    r12d, r12d
0x1cf31:      jne     0x1cfac
0x1cf33:      test    esi, esi
0x1cf35:      je      0x1cfac
0x1cf37:      lea     edx, [rax + 1]
0x1cf3a:      cmp     eax, 9
0x1cf3d:      je      0x1d024
0x1cf43:      lea     eax, [rdx + 0x30]
0x1cf46:      mov     rdi, qword ptr [rsp]
0x1cf4a:      lea     r8, [rdi + 0x287]
0x1cf51:      mov     byte ptr [rdi + 0x287], al
0x1cf57:      mov     eax, ebp
0x1cf59:      sub     r8, rbp
0x1cf5c:      cmp     ebp, 8
0x1cf5f:      jae     0x1cfd1
0x1cf61:      test    bpl, 4
0x1cf65:      jne     0x1d0d6

```

```

0x1cf6b:    test    eax, eax
0x1cf6d:    je      0x1cf7a
0x1cf6f:    movzx   edx, byte ptr [r15]
0x1cf73:    mov     byte ptr [r8], dl
0x1cf76:    test    al, 2
0x1cf78:    jne     0x1cf8e
0x1cf7a:    xor     esi, esi
0x1cf7c:    xor     edi, edi
0x1cf7e:    cmp     dword ptr [rsp + 0x10], 1
0x1cf83:    jne     0x1c7bf
0x1cf89:    jmp     0x1c7e8
0x1cf8e:    movzx   edx, word ptr [r15 + rax - 2]
0x1cf94:    mov     word ptr [r8 + rax - 2], dx
0x1cf9a:    jmp     0x1cf7a
0x1cf9c:    nop     dword ptr [rax]
0x1cfa0:    mov     edx, eax
0x1cfa2:    and     edx, 1
0x1cfa5:    add     edx, esi
0x1cfa7:    cmp     edx, 2
0x1cfaa:    jg      0x1cf37
0x1cfac:    test    eax, eax
0x1cfae:    jne     0x1d117
0x1cfb4:    mov     r8, qword ptr [rsp + 8]
0x1cfb9:    test    byte ptr [rsp + 0x48], 8
0x1cfbe:    jne     0x1cf7c
0x1cfc0:    mov     eax, 0x30
0x1cfc5:    jmp     0x1cf46
0x1cfca:    xor     ebx, ebx
0x1cfcc:    jmp     0x1ca3c
0x1cfd1:    mov     rax, qword ptr [r15]
0x1cfd4:    lea     rsi, [r8 + 8]
0x1cfd8:    mov     rdi, r15
0x1cfdb:    and     rsi, 0xfffffffffffffff8
0x1cddf:    mov     qword ptr [r8], rax
0x1cfe2:    mov     eax, ebp
0x1cfe4:    mov     rdx, qword ptr [r15 + rax - 8]
0x1cfe9:    mov     qword ptr [r8 + rax - 8], rdx
0x1cfee:    mov     rax, r8
0x1cff1:    sub     rax, rsi
0x1cff4:    sub     rdi, rax
0x1cff7:    add     eax, ebp
0x1cff9:    and     eax, 0xffffffff8
0x1cffc:    cmp     eax, 8
0x1cfff:    jb      0x1cf7a
0x1d005:    and     eax, 0xffffffff8
0x1d008:    mov     r9d, eax
0x1d00b:    xor     eax, eax
0x1d00d:    mov     edx, eax

```

```

0x1d00f:      add     eax, 8
0x1d012:      mov     r11, qword ptr [rdi + rdx]
0x1d016:      mov     qword ptr [rsi + rdx], r11
0x1d01a:      cmp     eax, r9d
0x1d01d:      jb      0x1d00d
0x1d01f:      jmp     0x1cf7a
0x1d024:      lea     rcx, [r9 + 1]
0x1d028:      cmp     r9, 9
0x1d02c:      je      0x1d0cc
0x1d032:      xor     esi, esi
0x1d034:      jmp     0x1cfb4
0x1d039:      mov     rax, rbp
0x1d03c:      mov     byte ptr [r8 - 1], 0x30
0x1d041:      not     rax
0x1d044:      add     r8, rax
0x1d047:      mov     eax, ebp
0x1d049:      cmp     ebp, 8
0x1d04c:      jae     0x1d080
0x1d04e:      and     ebp, 4
0x1d051:      jne     0x1d0f0
0x1d057:      test    eax, eax
0x1d059:      je      0x1ced9
0x1d05f:      movzx   edx, byte ptr [r15]
0x1d063:      mov     byte ptr [r8], dl
0x1d066:      test    al, 2
0x1d068:      je      0x1ced9
0x1d06e:      movzx   edx, word ptr [r15 + rax - 2]
0x1d074:      mov     word ptr [r8 + rax - 2], dx
0x1d07a:      jmp     0x1ced9
0x1d07f:      nop
0x1d080:      mov     rax, qword ptr [r15]
0x1d083:      lea     rcx, [r8 + 8]
0x1d087:      and     rcx, 0xfffffffffffffff8
0x1d08b:      mov     qword ptr [r8], rax
0x1d08e:      mov     eax, ebp
0x1d090:      mov     rdx, qword ptr [r15 + rax - 8]
0x1d095:      mov     qword ptr [r8 + rax - 8], rdx
0x1d09a:      mov     rax, r8
0x1d09d:      sub     rax, rcx
0x1d0a0:      sub     r15, rax
0x1d0a3:      add     eax, ebp
0x1d0a5:      and     eax, 0xfffffffff8
0x1d0a8:      cmp     eax, 8
0x1d0ab:      jb      0x1ced9
0x1d0b1:      and     eax, 0xfffffffff8
0x1d0b4:      xor     edx, edx
0x1d0b6:      mov     esi, edx
0x1d0b8:      add     edx, 8

```



```

0x1d0bb:    mov     rdi, qword ptr [r15 + rsi]
0x1d0bf:    mov     qword ptr [rcx + rsi], rdi
0x1d0c3:    cmp     edx, eax
0x1d0c5:    jb      0x1d0b6
0x1d0c7:    jmp     0x1ced9
0x1d0cc:    mov     r8, qword ptr [rsp + 8]
0x1d0d1:    jmp     0x1cf7a
0x1d0d6:    mov     edx, dword ptr [r15]
0x1d0d9:    mov     dword ptr [r8], edx
0x1d0dc:    mov     edx, dword ptr [r15 + rax - 4]
0x1d0e1:    mov     dword ptr [r8 + rax - 4], edx
0x1d0e6:    jmp     0x1cf7a
0x1d0eb:    call    0x49d0
0x1d0f0:    mov     edx, dword ptr [r15]
0x1d0f3:    mov     dword ptr [r8], edx
0x1d0f6:    mov     edx, dword ptr [r15 + rax - 4]
0x1d0fb:    mov     dword ptr [r8 + rax - 4], edx
0x1d100:    jmp     0x1ced9
0x1d105:    mov     rdx, qword ptr [rsp + 8]
0x1d10a:    test    eax, eax
0x1d10c:    jne     0x1cc5a
0x1d112:    jmp     0x1ca4f
0x1d117:    add     eax, 0x30
0x1d11a:    jmp     0x1cf46
0x1d11f:    nop
0x1d120:    endbr64
0x1d124:    push    r14
0x1d126:    push    r13
0x1d128:    mov     r13, rsi
0x1d12b:    push    r12
0x1d12d:    push    rbp
0x1d12e:    mov     rbp, rdx
0x1d131:    push    rbx
0x1d132:    mov     rbx, rdi
0x1d135:    sub     rsp, 0x10
0x1d139:    mov     rax, qword ptr fs:[0x28]
0x1d142:    mov     qword ptr [rsp + 8], rax
0x1d147:    xor     eax, eax
0x1d149:    test    rdi, rdi
0x1d14c:    je      0x1d270
0x1d152:    xor     r12d, r12d
0x1d155:    cmp     byte ptr [rbx], 0x27
0x1d158:    jne     0x1d164
0x1d15a:    add     rbx, 1
0x1d15e:    mov     r12d, 4
0x1d164:    lea     r14, [rip + 0x53ad]
0x1d16b:    mov     ecx, 4
0x1d170:    lea     rsi, [rip + 0xb819]

```

```

0x1d177:    mov     rdi, rbx
0x1d17a:    mov     rdx, r14
0x1d17d:    call    0x1dc80
0x1d182:    test    eax, eax
0x1d184:    js      0x1d1c0
0x1d186:    cdqe
0x1d188:    mov     qword ptr [rbp], 1
0x1d190:    or      r12d, dword ptr [r14 + rax*4]
0x1d194:    xor     eax, eax
0x1d196:    mov     dword ptr [r13], r12d
0x1d19a:    mov     rdx, qword ptr [rsp + 8]
0x1d19f:    sub     rdx, qword ptr fs:[0x28]
0x1d1a8:    jne     0x1d2fd
0x1d1ae:    add     rsp, 0x10
0x1d1b2:    pop     rbx
0x1d1b3:    pop     rbp
0x1d1b4:    pop     r12
0x1d1b6:    pop     r13
0x1d1b8:    pop     r14
0x1d1ba:    ret
0x1d1bb:    nop     dword ptr [rax + rax]
0x1d1c0:    xor     edx, edx
0x1d1c2:    mov     rsi, rsp
0x1d1c5:    lea     r8, [rip + 0x5329]
0x1d1cc:    mov     rcx, rbp
0x1d1cf:    mov     rdi, rbx
0x1d1d2:    call    0x1d7e0
0x1d1d7:    test    eax, eax
0x1d1d9:    jne     0x1d228
0x1d1db:    movzx    ecx, byte ptr [rbx]
0x1d1de:    lea     edx, [rcx - 0x30]
0x1d1e1:    mov     rcx, qword ptr [rsp]
0x1d1e5:    cmp     dl, 9
0x1d1e8:    ja      0x1d200
0x1d1ea:    jmp     0x1d217
0x1d1ec:    nop     dword ptr [rax]
0x1d1f0:    movzx    edi, byte ptr [rbx + 1]
0x1d1f4:    add     rbx, 1
0x1d1f8:    lea     edx, [rdi - 0x30]
0x1d1fb:    cmp     dl, 9
0x1d1fe:    jbe     0x1d217
0x1d200:    cmp     rcx, rbx
0x1d203:    jne     0x1d1f0
0x1d205:    cmp     byte ptr [rcx - 1], 0x42
0x1d209:    je      0x1d2d0
0x1d20f:    or      r12b, 0x80
0x1d213:    or      r12d, 0x20
0x1d217:    mov     rdx, qword ptr [rbp]

```

```

0x1d21b:    mov     dword ptr [r13], r12d
0x1d21f:    jmp     0x1d234
0x1d221:    nop     dword ptr [rax]
0x1d228:    mov     dword ptr [r13], 0
0x1d230:    mov     rdx, qword ptr [rbp]
0x1d234:    test    rdx, rdx
0x1d237:    jne     0x1d19a
0x1d23d:    lea     rdi, [rip + 0x52a1]
0x1d244:    call    0x47e0
0x1d249:    cmp     rax, 1
0x1d24d:    sbb     rax, rax
0x1d250:    and     eax, 0x200
0x1d255:    add     rax, 0x200
0x1d25b:    mov     qword ptr [rbp], rax
0x1d25f:    mov     eax, 4
0x1d264:    jmp     0x1d19a
0x1d269:    nop     dword ptr [rax]
0x1d270:    lea     rdi, [rip + 0x5259]
0x1d277:    call    0x47e0
0x1d27c:    mov     rbx, rax
0x1d27f:    test    rax, rax
0x1d282:    jne     0x1d152
0x1d288:    lea     rdi, [rip + 0x524c]
0x1d28f:    call    0x47e0
0x1d294:    mov     rbx, rax
0x1d297:    test    rax, rax
0x1d29a:    jne     0x1d152
0x1d2a0:    lea     rdi, [rip + 0x523e]
0x1d2a7:    call    0x47e0
0x1d2ac:    test    rax, rax
0x1d2af:    je      0x1d2e6
0x1d2b1:    mov     qword ptr [rbp], 0x200
0x1d2b9:    xor     eax, eax
0x1d2bb:    mov     dword ptr [r13], 0
0x1d2c3:    jmp     0x1d19a
0x1d2c8:    nop     dword ptr [rax + rax]
0x1d2d0:    or      r12d, 0x180
0x1d2d7:    cmp     byte ptr [rcx - 2], 0x69
0x1d2db:    jne     0x1d217
0x1d2e1:    jmp     0x1d213
0x1d2e6:    mov     qword ptr [rbp], 0x400
0x1d2ee:    xor     eax, eax
0x1d2f0:    mov     dword ptr [r13], 0
0x1d2f8:    jmp     0x1d19a
0x1d2fd:    call    0x49d0
0x1d302:    nop     word ptr cs:[rax + rax]
0x1d30c:    nop     dword ptr [rax]
0x1d310:    endbr64

```

```

0x1d314:    push    r12
0x1d316:    push    rbp
0x1d317:    mov     ebp, edi
0x1d319:    push    rbx
0x1d31a:    mov     rbx, qword ptr [rip + 0xe81f]
0x1d321:    test    rbx, rbx
0x1d324:    jne     0x1d339
0x1d326:    jmp     0x1d350
0x1d328:    nop     dword ptr [rax + rax]
0x1d330:    mov     rbx, qword ptr [rbx + 8]
0x1d334:    test    rbx, rbx
0x1d337:    je      0x1d350
0x1d339:    cmp     dword ptr [rbx], ebp
0x1d33b:    jne     0x1d330
0x1d33d:    xor     eax, eax
0x1d33f:    cmp     byte ptr [rbx + 0x10], 0
0x1d343:    je      0x1d349
0x1d345:    lea     rax, [rbx + 0x10]
0x1d349:    pop     rbx
0x1d34a:    pop     rbp
0x1d34b:    pop     r12
0x1d34d:    ret
0x1d34e:    nop
0x1d350:    mov     edi, ebp
0x1d352:    lea     r12, [rip + 0x508c]
0x1d359:    call    0x4970
0x1d35e:    mov     edi, 0x11
0x1d363:    test    rax, rax
0x1d366:    je      0x1d377
0x1d368:    mov     r12, qword ptr [rax]
0x1d36b:    mov     rdi, r12
0x1d36e:    call    0x49b0
0x1d373:    lea     rdi, [rax + 0x11]
0x1d377:    call    0x1d670
0x1d37c:    mov     rsi, r12
0x1d37f:    mov     dword ptr [rax], ebp
0x1d381:    lea     rdi, [rax + 0x10]
0x1d385:    mov     rbx, rax
0x1d388:    call    0x48b0
0x1d38d:    mov     rax, qword ptr [rip + 0xe7ac]
0x1d394:    mov     qword ptr [rip + 0xe7a5], rbx
0x1d39b:    mov     qword ptr [rbx + 8], rax
0x1d39f:    jmp     0x1d33d
0x1d3a1:    nop     word ptr cs:[rax + rax]
0x1d3ac:    nop     dword ptr [rax]
0x1d3b0:    endbr64
0x1d3b4:    push    r12
0x1d3b6:    push    rbp

```

```

0x1d3b7:      mov     rbp, rdi
0x1d3ba:      push   rbx
0x1d3bb:      mov     rbx, qword ptr [rip + 0xe77e]
0x1d3c2:      test    rbx, rbx
0x1d3c5:      je       0x1d400
0x1d3c7:      movzx    r12d, byte ptr [rdi]
0x1d3cb:      jmp      0x1d3d9
0x1d3cd:      nop      dword ptr [rax]
0x1d3d0:      mov     rbx, qword ptr [rbx + 8]
0x1d3d4:      test    rbx, rbx
0x1d3d7:      je       0x1d400
0x1d3d9:      cmp      byte ptr [rbx + 0x10], r12b
0x1d3dd:      jne      0x1d3d0
0x1d3df:      lea      rdi, [rbx + 0x10]
0x1d3e3:      mov     rsi, rbp
0x1d3e6:      call     0x4b30
0x1d3eb:      test    eax, eax
0x1d3ed:      jne      0x1d3d0
0x1d3ef:      mov     r12, rbx
0x1d3f2:      pop      rbx
0x1d3f3:      pop      rbp
0x1d3f4:      mov     rax, r12
0x1d3f7:      pop      r12
0x1d3f9:      ret
0x1d3fa:      nop      word ptr [rax + rax]
0x1d400:      mov     rbx, qword ptr [rip + 0xe731]
0x1d407:      test    rbx, rbx
0x1d40a:      je       0x1d448
0x1d40c:      movzx    r12d, byte ptr [rbp]
0x1d411:      jmp      0x1d421
0x1d413:      nop      dword ptr [rax + rax]
0x1d418:      mov     rbx, qword ptr [rbx + 8]
0x1d41c:      test    rbx, rbx
0x1d41f:      je       0x1d448
0x1d421:      cmp      byte ptr [rbx + 0x10], r12b
0x1d425:      jne      0x1d418
0x1d427:      lea      rdi, [rbx + 0x10]
0x1d42b:      mov     rsi, rbp
0x1d42e:      call     0x4b30
0x1d433:      test    eax, eax
0x1d435:      jne      0x1d418
0x1d437:      xor      r12d, r12d
0x1d43a:      pop      rbx
0x1d43b:      pop      rbp
0x1d43c:      mov     rax, r12
0x1d43f:      pop      r12
0x1d441:      ret
0x1d442:      nop      word ptr [rax + rax]

```

```

0x1d448:    mov     rdi, rbp
0x1d44b:    call   0x4b60
0x1d450:    mov     rdi, rbp
0x1d453:    mov     r12, rax
0x1d456:    call   0x49b0
0x1d45b:    lea     rdi, [rax + 0x11]
0x1d45f:    call   0x1d670
0x1d464:    mov     rsi, rbp
0x1d467:    lea     rdi, [rax + 0x10]
0x1d46b:    mov     rbx, rax
0x1d46e:    call   0x48b0
0x1d473:    test    r12, r12
0x1d476:    je      0x1d4a0
0x1d478:    mov     eax, dword ptr [r12 + 0x10]
0x1d47d:    mov     dword ptr [rbx], eax
0x1d47f:    mov     rax, qword ptr [rip + 0xe6ba]
0x1d486:    mov     qword ptr [rip + 0xe6b3], rbx
0x1d48d:    mov     qword ptr [rbx + 8], rax
0x1d491:    jmp     0x1d3ef
0x1d496:    nop     word ptr cs:[rax + rax]
0x1d4a0:    mov     rax, qword ptr [rip + 0xe691]
0x1d4a7:    mov     qword ptr [rip + 0xe68a], rbx
0x1d4ae:    mov     qword ptr [rbx + 8], rax
0x1d4b2:    mov     rax, r12
0x1d4b5:    pop     rbx
0x1d4b6:    pop     rbp
0x1d4b7:    pop     r12
0x1d4b9:    ret
0x1d4ba:    nop     word ptr [rax + rax]
0x1d4c0:    endbr64
0x1d4c4:    push    r12
0x1d4c6:    push    rbp
0x1d4c7:    mov     ebp, edi
0x1d4c9:    push    rbx
0x1d4ca:    mov     rbx, qword ptr [rip + 0xe65f]
0x1d4d1:    test    rbx, rbx
0x1d4d4:    jne     0x1d4e9
0x1d4d6:    jmp     0x1d500
0x1d4d8:    nop     dword ptr [rax + rax]
0x1d4e0:    mov     rbx, qword ptr [rbx + 8]
0x1d4e4:    test    rbx, rbx
0x1d4e7:    je      0x1d500
0x1d4e9:    cmp     dword ptr [rbx], ebp
0x1d4eb:    jne     0x1d4e0
0x1d4ed:    xor     eax, eax
0x1d4ef:    cmp     byte ptr [rbx + 0x10], 0
0x1d4f3:    je      0x1d4f9
0x1d4f5:    lea     rax, [rbx + 0x10]

```

```

0x1d4f9:    pop     rbx
0x1d4fa:    pop     rbp
0x1d4fb:    pop     r12
0x1d4fd:    ret
0x1d4fe:    nop
0x1d500:    mov     edi, ebp
0x1d502:    lea     r12, [rip + 0x4edc]
0x1d509:    call    0x4a10
0x1d50e:    mov     edi, 0x11
0x1d513:    test    rax, rax
0x1d516:    je      0x1d527
0x1d518:    mov     r12, qword ptr [rax]
0x1d51b:    mov     rdi, r12
0x1d51e:    call    0x49b0
0x1d523:    lea     rdi, [rax + 0x11]
0x1d527:    call    0x1d670
0x1d52c:    mov     rsi, r12
0x1d52f:    mov     dword ptr [rax], ebp
0x1d531:    lea     rdi, [rax + 0x10]
0x1d535:    mov     rbx, rax
0x1d538:    call    0x48b0
0x1d53d:    mov     rax, qword ptr [rip + 0xe5ec]
0x1d544:    mov     qword ptr [rip + 0xe5e5], rbx
0x1d54b:    mov     qword ptr [rbx + 8], rax
0x1d54f:    jmp     0x1d4ed
0x1d551:    nop     word ptr cs:[rax + rax]
0x1d55c:    nop     dword ptr [rax]
0x1d560:    endbr64
0x1d564:    push    r12
0x1d566:    push    rbp
0x1d567:    mov     rbp, rdi
0x1d56a:    push    rbx
0x1d56b:    mov     rbx, qword ptr [rip + 0xe5be]
0x1d572:    test    rbx, rbx
0x1d575:    je      0x1d5b0
0x1d577:    movzx   r12d, byte ptr [rdi]
0x1d57b:    jmp     0x1d589
0x1d57d:    nop     dword ptr [rax]
0x1d580:    mov     rbx, qword ptr [rbx + 8]
0x1d584:    test    rbx, rbx
0x1d587:    je      0x1d5b0
0x1d589:    cmp     byte ptr [rbx + 0x10], r12b
0x1d58d:    jne     0x1d580
0x1d58f:    lea     rdi, [rbx + 0x10]
0x1d593:    mov     rsi, rbp
0x1d596:    call    0x4b30
0x1d59b:    test    eax, eax
0x1d59d:    jne     0x1d580

```

```

0x1d59f:    mov     r12, rbx
0x1d5a2:    pop     rbx
0x1d5a3:    pop     rbp
0x1d5a4:    mov     rax, r12
0x1d5a7:    pop     r12
0x1d5a9:    ret
0x1d5aa:    nop     word ptr [rax + rax]
0x1d5b0:    mov     rbx, qword ptr [rip + 0xe571]
0x1d5b7:    test    rbx, rbx
0x1d5ba:    je      0x1d5f8
0x1d5bc:    movzx   r12d, byte ptr [rbp]
0x1d5c1:    jmp     0x1d5d1
0x1d5c3:    nop     dword ptr [rax + rax]
0x1d5c8:    mov     rbx, qword ptr [rbx + 8]
0x1d5cc:    test    rbx, rbx
0x1d5cf:    je      0x1d5f8
0x1d5d1:    cmp     byte ptr [rbx + 0x10], r12b
0x1d5d5:    jne     0x1d5c8
0x1d5d7:    lea     rdi, [rbx + 0x10]
0x1d5db:    mov     rsi, rbp
0x1d5de:    call    0x4b30
0x1d5e3:    test    eax, eax
0x1d5e5:    jne     0x1d5c8
0x1d5e7:    xor     r12d, r12d
0x1d5ea:    pop     rbx
0x1d5eb:    pop     rbp
0x1d5ec:    mov     rax, r12
0x1d5ef:    pop     r12
0x1d5f1:    ret
0x1d5f2:    nop     word ptr [rax + rax]
0x1d5f8:    mov     rdi, rbp
0x1d5fb:    call    0x4bc0
0x1d600:    mov     rdi, rbp
0x1d603:    mov     r12, rax
0x1d606:    call    0x49b0
0x1d60b:    lea     rdi, [rax + 0x11]
0x1d60f:    call    0x1d670
0x1d614:    mov     rsi, rbp
0x1d617:    lea     rdi, [rax + 0x10]
0x1d61b:    mov     rbx, rax
0x1d61e:    call    0x48b0
0x1d623:    test    r12, r12
0x1d626:    je      0x1d650
0x1d628:    mov     eax, dword ptr [r12 + 0x10]
0x1d62d:    mov     dword ptr [rbx], eax
0x1d62f:    mov     rax, qword ptr [rip + 0xe4fa]
0x1d636:    mov     qword ptr [rip + 0xe4f3], rbx
0x1d63d:    mov     qword ptr [rbx + 8], rax

```



```

0x1d641:      jmp      0x1d59f
0x1d646:      nop      word ptr cs:[rax + rax]
0x1d650:      mov      rax, qword ptr [rip + 0xe4d1]
0x1d657:      mov      qword ptr [rip + 0xe4ca], rbx
0x1d65e:      mov      qword ptr [rbx + 8], rax
0x1d662:      mov      rax, r12
0x1d665:      pop      rbx
0x1d666:      pop      rbp
0x1d667:      pop      r12
0x1d669:      ret
0x1d66a:      nop      word ptr [rax + rax]
0x1d670:      endbr64
0x1d674:      push     rbx
0x1d675:      mov      rbx, rdi
0x1d678:      call     0x4c20
0x1d67d:      test     rax, rax
0x1d680:      jne      0x1d687
0x1d682:      test     rbx, rbx
0x1d685:      jne      0x1d689
0x1d687:      pop      rbx
0x1d688:      ret
0x1d689:      call     0x18cf0
0x1d68e:      nop
0x1d690:      endbr64
0x1d694:      push     rbx
0x1d695:      mov      rbx, rsi
0x1d698:      call     0x4ce0
0x1d69d:      test     rax, rax
0x1d6a0:      jne      0x1d6a7
0x1d6a2:      test     rbx, rbx
0x1d6a5:      jne      0x1d6a9
0x1d6a7:      pop      rbx
0x1d6a8:      ret
0x1d6a9:      call     0x18cf0
0x1d6ae:      nop
0x1d6b0:      endbr64
0x1d6b4:      push     rbx
0x1d6b5:      mov      rax, rsi
0x1d6b8:      mov      rsi, qword ptr [rsi]
0x1d6bb:      test     rdi, rdi
0x1d6be:      je       0x1d6f0
0x1d6c0:      movabs   rdx, 0xaaaaaaaaaaaaaaaa9
0x1d6ca:      cmp      rsi, rdx
0x1d6cd:      ja       0x1d701
0x1d6cf:      lea      rdx, [rsi + 1]
0x1d6d3:      shr      rdx, 1
0x1d6d6:      add      rsi, rdx
0x1d6d9:      setne    bl

```

```

0x1d6dc:    mov     qword ptr [rax], rsi
0x1d6df:    call   0x4ce0
0x1d6e4:    test    rax, rax
0x1d6e7:    jne     0x1d6ed
0x1d6e9:    test    bl, bl
0x1d6eb:    jne     0x1d701
0x1d6ed:    pop     rbx
0x1d6ee:    ret
0x1d6ef:    nop
0x1d6f0:    mov     ebx, 1
0x1d6f5:    test    rsi, rsi
0x1d6f8:    jne     0x1d6dc
0x1d6fa:    mov     esi, 0x40
0x1d6ff:    jmp     0x1d6dc
0x1d701:    call   0x18cf0
0x1d706:    nop     word ptr cs:[rax + rax]
0x1d710:    endbr64
0x1d714:    push    rbx
0x1d715:    mov     esi, 1
0x1d71a:    mov     rbx, rdi
0x1d71d:    call   0x4b10
0x1d722:    test    rax, rax
0x1d725:    jne     0x1d72c
0x1d727:    test    rbx, rbx
0x1d72a:    jne     0x1d72e
0x1d72c:    pop     rbx
0x1d72d:    ret
0x1d72e:    call   0x18cf0
0x1d733:    nop     word ptr cs:[rax + rax]
0x1d73e:    nop
0x1d740:    endbr64
0x1d744:    sub     rsp, 8
0x1d748:    call   0x4b10
0x1d74d:    test    rax, rax
0x1d750:    je      0x1d757
0x1d752:    add     rsp, 8
0x1d756:    ret
0x1d757:    call   0x18cf0
0x1d75c:    nop     dword ptr [rax]
0x1d760:    endbr64
0x1d764:    push    r12
0x1d766:    mov     r12, rsi
0x1d769:    push    rbp
0x1d76a:    mov     rbp, rdi
0x1d76d:    mov     rdi, rsi
0x1d770:    sub     rsp, 8
0x1d774:    call   0x4c20
0x1d779:    mov     rdi, rax

```

```

0x1d77c:      test    rax, rax
0x1d77f:      jne     0x1d786
0x1d781:      test    r12, r12
0x1d784:      jne     0x1d798
0x1d786:      add     rsp, 8
0x1d78a:      mov     rdx, r12
0x1d78d:      mov     rsi, rbp
0x1d790:      pop     rbp
0x1d791:      pop     r12
0x1d793:      jmp     0x4bb0
0x1d798:      call    0x18cf0
0x1d79d:      nop     dword ptr [rax]
0x1d7a0:      endbr64
0x1d7a4:      push    r12
0x1d7a6:      push    rbp
0x1d7a7:      mov     rbp, rdi
0x1d7aa:      sub     rsp, 8
0x1d7ae:      call    0x49b0
0x1d7b3:      lea     r12, [rax + 1]
0x1d7b7:      mov     rdi, r12
0x1d7ba:      call    0x4c20
0x1d7bf:      test    rax, rax
0x1d7c2:      je      0x1d7d9
0x1d7c4:      add     rsp, 8
0x1d7c8:      mov     rdx, r12
0x1d7cb:      mov     rsi, rbp
0x1d7ce:      mov     rdi, rax
0x1d7d1:      pop     rbp
0x1d7d2:      pop     r12
0x1d7d4:      jmp     0x4bb0
0x1d7d9:      call    0x18cf0
0x1d7de:      nop
0x1d7e0:      endbr64
0x1d7e4:      push    r15
0x1d7e6:      mov     r15, rsi
0x1d7e9:      push    r14
0x1d7eb:      mov     r14, rcx
0x1d7ee:      push    r13
0x1d7f0:      mov     r13d, edx
0x1d7f3:      push    r12
0x1d7f5:      mov     r12, r8
0x1d7f8:      push    rbp
0x1d7f9:      mov     rbp, rdi
0x1d7fc:      push    rbx
0x1d7fd:      sub     rsp, 0x28
0x1d801:      movzx   ebx, byte ptr [rdi]
0x1d804:      mov     rax, qword ptr fs:[0x28]
0x1d80d:      mov     qword ptr [rsp + 0x18], rax

```

```

0x1d812:    xor     eax, eax
0x1d814:    lea     rax, [rsp + 0x10]
0x1d819:    test    rsi, rsi
0x1d81c:    cmov     r15, rax
0x1d820:    call    0x4f00
0x1d825:    mov     rcx, qword ptr [rax]
0x1d828:    mov     rax, rbp
0x1d82b:    jmp     0x1d838
0x1d82d:    nop     dword ptr [rax]
0x1d830:    movzx   ebx, byte ptr [rax + 1]
0x1d834:    add     rax, 1
0x1d838:    movzx   edx, bl
0x1d83b:    test    byte ptr [rcx + rdx*2 + 1], 0x20
0x1d840:    jne     0x1d830
0x1d842:    cmp     bl, 0x2d
0x1d845:    je      0x1d8ca
0x1d84b:    call    0x4860
0x1d850:    mov     edx, r13d
0x1d853:    mov     rsi, r15
0x1d856:    mov     rdi, rbp
0x1d859:    mov     dword ptr [rax], 0
0x1d85f:    mov     qword ptr [rsp + 8], rax
0x1d864:    call    0x4dd0
0x1d869:    mov     r13, qword ptr [r15]
0x1d86c:    mov     rcx, qword ptr [rsp + 8]
0x1d871:    mov     rbx, rax
0x1d874:    cmp     r13, rbp
0x1d877:    je      0x1d8d8
0x1d879:    mov     eax, dword ptr [rcx]
0x1d87b:    test    eax, eax
0x1d87d:    jne     0x1d8c0
0x1d87f:    xor     ebp, ebp
0x1d881:    test    r12, r12
0x1d884:    je      0x1d893
0x1d886:    movzx   edx, byte ptr [r13]
0x1d88b:    test    dl, dl
0x1d88d:    jne     0x1d958
0x1d893:    mov     qword ptr [r14], rbx
0x1d896:    mov     rax, qword ptr [rsp + 0x18]
0x1d89b:    sub     rax, qword ptr fs:[0x28]
0x1d8a4:    jne     0x1dbf9
0x1d8aa:    add     rsp, 0x28
0x1d8ae:    mov     eax, ebp
0x1d8b0:    pop     rbx
0x1d8b1:    pop     rbp
0x1d8b2:    pop     r12
0x1d8b4:    pop     r13
0x1d8b6:    pop     r14

```

```

0x1d8b8:      pop     r15
0x1d8ba:      ret
0x1d8bb:      nop     dword ptr [rax + rax]
0x1d8c0:      mov     ebp, 1
0x1d8c5:      cmp     eax, 0x22
0x1d8c8:      je      0x1d881
0x1d8ca:      mov     ebp, 4
0x1d8cf:      jmp     0x1d896
0x1d8d1:      nop     dword ptr [rax]
0x1d8d8:      test    r12, r12
0x1d8db:      je      0x1d8ca
0x1d8dd:      movzx    edx, byte ptr [rbp]
0x1d8e1:      test    dl, dl
0x1d8e3:      je      0x1d8ca
0x1d8e5:      movsx    esi, dl
0x1d8e8:      mov     rdi, r12
0x1d8eb:      mov     byte ptr [rsp + 8], dl
0x1d8ef:      xor     ebp, ebp
0x1d8f1:      call    0x4a00
0x1d8f6:      mov     ebx, 1
0x1d8fb:      movzx    edx, byte ptr [rsp + 8]
0x1d900:      test    rax, rax
0x1d903:      je      0x1d8ca
0x1d905:      mov     esi, 0x30
0x1d90a:      mov     rdi, r12
0x1d90d:      mov     byte ptr [rsp + 8], dl
0x1d911:      call    0x4a00
0x1d916:      movzx    edx, byte ptr [rsp + 8]
0x1d91b:      test    rax, rax
0x1d91e:      je      0x1d97c
0x1d920:      movzx    eax, byte ptr [r13 + 1]
0x1d925:      cmp     al, 0x44
0x1d927:      je      0x1d9a2
0x1d929:      cmp     al, 0x69
0x1d92b:      je      0x1dbe0
0x1d931:      cmp     al, 0x42
0x1d933:      je      0x1d9a2
0x1d935:      sub     edx, 0x42
0x1d938:      cmp     dl, 0x35
0x1d93b:      ja      0x1d971
0x1d93d:      lea     rcx, [rip + 0x4c0c]
0x1d944:      movzx    edx, dl
0x1d947:      movsxd   rax, dword ptr [rcx + rdx*4]
0x1d94b:      add     rax, rcx
0x1d94e:      jmp     rax
0x1d951:      -----

```

```

0x1d956:      -----
0x1d957:      -----
0x1d958:      movsx    esi, dl
0x1d95b:      mov      rdi, r12
0x1d95e:      mov      byte ptr [rsp + 8], dl
0x1d962:      call     0x4a00
0x1d967:      movzx    edx, byte ptr [rsp + 8]
0x1d96c:      test     rax, rax
0x1d96f:      jne       0x1d905
0x1d971:      mov      qword ptr [r14], rbx
0x1d974:      or        ebp, 2
0x1d977:      jmp       0x1d896
0x1d97c:      mov      esi, 1
0x1d981:      mov      ecx, 0x400
0x1d986:      sub      edx, 0x42
0x1d989:      cmp      dl, 0x35
0x1d98c:      ja        0x1d971
0x1d98e:      lea      rdi, [rip + 0x4c93]
0x1d995:      movzx    edx, dl
0x1d998:      movsxd   rax, dword ptr [rdi + rdx*4]
0x1d99c:      add      rax, rdi
0x1d99f:      jmp      rax
0x1d9a2:      mov      esi, 2
0x1d9a7:      mov      ecx, 0x3e8
0x1d9ac:      jmp       0x1d986
0x1d9ae:      movsxd   rsi, esi
0x1d9b1:      mov      edi, 4
0x1d9b6:      xor      r8d, r8d
0x1d9b9:      nop      dword ptr [rax]
0x1d9c0:      mov      rax, rbx
0x1d9c3:      mul      rcx
0x1d9c6:      mov      rbx, rax
0x1d9c9:      jo        0x1dc0e
0x1d9cf:      sub      edi, 1
0x1d9d2:      jne       0x1d9c0
0x1d9d4:      or        ebp, r8d
0x1d9d7:      add      rsi, r13
0x1d9da:      mov      eax, ebp
0x1d9dc:      or        eax, 2
0x1d9df:      mov      qword ptr [r15], rsi
0x1d9e2:      cmp      byte ptr [rsi], 0
0x1d9e5:      cmovne   ebp, eax
0x1d9e8:      jmp       0x1d893
0x1d9ed:      movsxd   rsi, esi
0x1d9f0:      mov      rax, rbx
0x1d9f3:      mul      rcx
0x1d9f6:      jo        0x1da13

```

```

0x1d9f8:      mul     rcx
0x1d9fb:      jo      0x1da13
0x1d9fd:      mov     rbx, rax
0x1da00:      jmp     0x1d9d7
0x1da02:      movsxd  rdx, ecx
0x1da05:      movsxd  rsi, esi
0x1da08:      mov     rax, rbx
0x1da0b:      mul     rdx
0x1da0e:      mov     rbx, rax
0x1da11:      jno     0x1d9d7
0x1da13:      mov     ebp, 1
0x1da18:      or      rbx, 0xffffffffffffffff
0x1da1c:      jmp     0x1d9d7
0x1da1e:      nop
0x1da20:      movsxd  rsi, esi
0x1da23:      mov     rax, rbx
0x1da26:      mul     rcx
0x1da29:      jo      0x1dbfe
0x1da2f:      mul     rcx
0x1da32:      jo      0x1dbfe
0x1da38:      mul     rcx
0x1da3b:      jo      0x1dbfe
0x1da41:      mov     rbx, rax
0x1da44:      xor     eax, eax
0x1da46:      or      ebp, eax
0x1da48:      jmp     0x1d9d7
0x1da4a:      movsxd  rsi, esi
0x1da4d:      mov     edi, 5
0x1da52:      xor     r8d, r8d
0x1da55:      nop     dword ptr [rax]
0x1da58:      mov     rax, rbx
0x1da5b:      mul     rcx
0x1da5e:      mov     rbx, rax
0x1da61:      jo      0x1dc1d
0x1da67:      sub     edi, 1
0x1da6a:      jne     0x1da58
0x1da6c:      jmp     0x1d9d4
0x1da71:      movsxd  rsi, esi
0x1da74:      mov     edi, 6
0x1da79:      xor     r8d, r8d
0x1da7c:      nop     dword ptr [rax]
0x1da80:      mov     rax, rbx
0x1da83:      mul     rcx
0x1da86:      mov     rbx, rax
0x1da89:      jo      0x1dc2c
0x1da8f:      sub     edi, 1
0x1da92:      jne     0x1da80
0x1da94:      jmp     0x1d9d4

```

```

0x1da99:      movsxd  rsi, esi
0x1da9c:      mov     rax, rbx
0x1da9f:      shr     rax, 0x36
0x1daa3:      jne     0x1da13
0x1daa9:      shl     rbx, 0xa
0x1daad:      jmp     0x1d9d7
0x1dab2:      movsxd  rsi, esi
0x1dab5:      mov     edi, 8
0x1daba:      xor     r8d, r8d
0x1dabd:      nop     dword ptr [rax]
0x1dac0:      mov     rax, rbx
0x1dac3:      mul     rcx
0x1dac6:      mov     rbx, rax
0x1dac9:      jo      0x1dc3b
0x1dacf:      sub     edi, 1
0x1dad2:      jne     0x1dac0
0x1dad4:      jmp     0x1d9d4
0x1dad9:      movsxd  rsi, esi
0x1dadc:      test    rbx, rbx
0x1dada:      js      0x1da13
0x1dae5:      add     rbx, rbx
0x1dae8:      jmp     0x1d9d7
0x1daed:      movsxd  rsi, esi
0x1daf0:      mov     rax, rbx
0x1daf3:      shr     rax, 0x37
0x1daf7:      jne     0x1da13
0x1dafd:      shl     rbx, 9
0x1db01:      jmp     0x1d9d7
0x1db06:      movsxd  rsi, esi
0x1db09:      mov     edi, 7
0x1db0e:      xor     r8d, r8d
0x1db11:      nop     dword ptr [rax]
0x1db18:      mov     rax, rbx
0x1db1b:      mul     rcx
0x1db1e:      mov     rbx, rax
0x1db21:      jo      0x1dc4a
0x1db27:      sub     edi, 1
0x1db2a:      jne     0x1db18
0x1db2c:      jmp     0x1d9d4
0x1db31:      movsxd  rsi, esi
0x1db34:      jmp     0x1d9d7
0x1db39:      mov     esi, 1
0x1db3e:      mov     ecx, 0x400
0x1db43:      jmp     0x1d9b1
0x1db48:      mov     esi, 1
0x1db4d:      mov     ecx, 0x400
0x1db52:      jmp     0x1d9f0
0x1db57:      mov     esi, 1

```



```

0x1db5c:      mov     edx, 0x400
0x1db61:      jmp     0x1da08
0x1db66:      mov     esi, 1
0x1db6b:      mov     ecx, 0x400
0x1db70:      jmp     0x1da23
0x1db75:      mov     esi, 1
0x1db7a:      mov     ecx, 0x400
0x1db7f:      jmp     0x1dab5
0x1db84:      mov     esi, 1
0x1db89:      mov     ecx, 0x400
0x1db8e:      jmp     0x1da4d
0x1db93:      mov     esi, 1
0x1db98:      mov     ecx, 0x400
0x1db9d:      jmp     0x1da74
0x1dba2:      mov     esi, 1
0x1dba7:      jmp     0x1da9c
0x1dbac:      mov     esi, 1
0x1dbb1:      jmp     0x1dadc
0x1dbb6:      mov     esi, 1
0x1dbbb:      jmp     0x1d9d7
0x1dbc0:      mov     esi, 1
0x1dbc5:      jmp     0x1daf0
0x1dbca:      mov     esi, 1
0x1dbcf:      mov     ecx, 0x400
0x1dbd4:      jmp     0x1db09
0x1dbd9:      nop     dword ptr [rax]
0x1dbe0:      xor     esi, esi
0x1dbe2:      cmp     byte ptr [r13 + 2], 0x42
0x1dbe7:      mov     ecx, 0x400
0x1dbec:      sete    sil
0x1dbf0:      lea     esi, [rsi + rsi + 1]
0x1dbf4:      jmp     0x1d986
0x1dbf9:      call    0x49d0
0x1dbfe:      mov     eax, 1
0x1dc03:      or      rbx, 0xffffffffffffffff
0x1dc07:      or      ebp, eax
0x1dc09:      jmp     0x1d9d7
0x1dc0e:      mov     r8d, 1
0x1dc14:      or      rbx, 0xffffffffffffffff
0x1dc18:      jmp     0x1d9cf
0x1dc1d:      mov     r8d, 1
0x1dc23:      or      rbx, 0xffffffffffffffff
0x1dc27:      jmp     0x1da67
0x1dc2c:      mov     r8d, 1
0x1dc32:      or      rbx, 0xffffffffffffffff
0x1dc36:      jmp     0x1da8f
0x1dc3b:      mov     r8d, 1
0x1dc41:      or      rbx, 0xffffffffffffffff

```

```

0x1dc45:      jmp      0x1dacf
0x1dc4a:      mov      r8d, 1
0x1dc50:      or       rbx, 0xffffffffffffffff
0x1dc54:      jmp      0x1db27
0x1dc59:      nop      dword ptr [rax]
0x1dc60:      endbr64
0x1dc64:      push     rax
0x1dc65:      pop      rax
0x1dc66:      sub      rsp, 8
0x1dc6a:      mov      edi, dword ptr [rip + 0xcb90]
0x1dc70:      call     0x4e20
0x1dc75:      nop      word ptr cs:[rax + rax]
0x1dc80:      endbr64
0x1dc84:      push     r15
0x1dc86:      push     r14
0x1dc88:      mov      r14, rdi
0x1dc8b:      push     r13
0x1dc8d:      mov      r13, rcx
0x1dc90:      push     r12
0x1dc92:      push     rbp
0x1dc93:      mov      rbp, rdx
0x1dc96:      push     rbx
0x1dc97:      mov      rbx, rsi
0x1dc9a:      sub      rsp, 0x28
0x1dc9e:      mov      qword ptr [rsp + 8], rsi
0x1dca3:      mov      qword ptr [rsp + 0x18], rdx
0x1dca8:      call     0x49b0
0x1dcad:      mov      r15, qword ptr [rbx]
0x1dcb0:      test     r15, r15
0x1dcb3:      je       0x1dd86
0x1dcb9:      mov      byte ptr [rsp + 0x17], 0
0x1dcbe:      mov      r12, rax
0x1dcc1:      xor      ebx, ebx
0x1dcc3:      mov      qword ptr [rsp], -1
0x1dccb:      jmp      0x1dd18
0x1dccd:      nop      dword ptr [rax]
0x1dcd0:      mov      rax, qword ptr [rsp + 0x18]
0x1dcd5:      test     rax, rax
0x1dcd8:      je       0x1dd48
0x1dcda:      mov      rdi, qword ptr [rsp]
0x1dcde:      mov      rdx, r13
0x1dce1:      mov      rsi, rbp
0x1dce4:      imul     rdi, r13
0x1dce8:      add      rdi, rax
0x1dceb:      call     0x4af0
0x1dcf0:      mov      ecx, 1
0x1dcf5:      test     eax, eax
0x1dcf7:      movzx    eax, byte ptr [rsp + 0x17]

```

```

0x1dcfc:      cmovne  eax, ecx
0x1dcff:      mov     byte ptr [rsp + 0x17], al
0x1dd03:      mov     rax, qword ptr [rsp + 8]
0x1dd08:      add     rbx, 1
0x1dd0c:      add     rbp, r13
0x1dd0f:      mov     r15, qword ptr [rax + rbx*8]
0x1dd13:      test    r15, r15
0x1dd16:      je       0x1dd50
0x1dd18:      mov     rdx, r12
0x1dd1b:      mov     rsi, r14
0x1dd1e:      mov     rdi, r15
0x1dd21:      call    0x4890
0x1dd26:      test    eax, eax
0x1dd28:      jne      0x1dd03
0x1dd2a:      mov     rdi, r15
0x1dd2d:      call    0x49b0
0x1dd32:      cmp     rax, r12
0x1dd35:      je       0x1dd80
0x1dd37:      cmp     qword ptr [rsp], -1
0x1dd3c:      jne      0x1dcd0
0x1dd3e:      mov     qword ptr [rsp], rbx
0x1dd42:      jmp     0x1dd03
0x1dd44:      nop     dword ptr [rax]
0x1dd48:      mov     byte ptr [rsp + 0x17], 1
0x1dd4d:      jmp     0x1dd03
0x1dd4f:      nop
0x1dd50:      cmp     byte ptr [rsp + 0x17], 0
0x1dd55:      mov     rax, -2
0x1dd5c:      cmov     rax, qword ptr [rsp]
0x1dd61:      mov     qword ptr [rsp], rax
0x1dd65:      mov     rax, qword ptr [rsp]
0x1dd69:      add     rsp, 0x28
0x1dd6d:      pop     rbx
0x1dd6e:      pop     rbp
0x1dd6f:      pop     r12
0x1dd71:      pop     r13
0x1dd73:      pop     r14
0x1dd75:      pop     r15
0x1dd77:      ret
0x1dd78:      nop     dword ptr [rax + rax]
0x1dd80:      mov     qword ptr [rsp], rbx
0x1dd84:      jmp     0x1dd65
0x1dd86:      mov     qword ptr [rsp], -1
0x1dd8e:      jmp     0x1dd65
0x1dd90:      endbr64
0x1dd94:      push    r13
0x1dd96:      cmp     rdx, -1
0x1dd9a:      mov     r13, rsi

```

```

0x1dd9d:    lea     rax, [rip + 0x4977]
0x1dda4:    push    r12
0x1dda6:    mov     rsi, rdi
0x1dda9:    lea     r12, [rip + 0x4950]
0x1ddb0:    mov     edi, 1
0x1ddb5:    push    rbx
0x1ddb6:    cmovne  r12, rax
0x1ddba:    call    0x1e000
0x1ddbf:    mov     rdx, r13
0x1ddc2:    mov     esi, 5
0x1ddc7:    xor     edi, edi
0x1ddc9:    mov     rbx, rax
0x1ddcc:    call    0x1ee20
0x1ddd1:    mov     r8, rbx
0x1ddd4:    mov     rdx, r12
0x1ddd7:    pop     rbx
0x1ddd8:    mov     rcx, rax
0x1dddb:    pop     r12
0x1dddd:    xor     esi, esi
0x1dddf:    xor     edi, edi
0x1dde1:    xor     eax, eax
0x1dde3:    pop     r13
0x1dde5:    jmp     0x4d50
0x1ddea:    nop     word ptr [rax + rax]
0x1ddf0:    endbr64
0x1ddf4:    push    r15
0x1ddf6:    push    r14
0x1ddf8:    lea     r14, [rip + 0x494e]
0x1ddff:    push    r13
0x1de01:    xor     r13d, r13d
0x1de04:    push    r12
0x1de06:    mov     r12, rdx
0x1de09:    mov     edx, 0x14
0x1de0e:    push    rbp
0x1de0f:    mov     rbp, rsi
0x1de12:    mov     esi, 1
0x1de17:    push    rbx
0x1de18:    mov     rbx, rdi
0x1de1b:    sub     rsp, 0x18
0x1de1f:    mov     rcx, qword ptr [rip + 0xca5a]
0x1de26:    mov     qword ptr [rsp + 8], rdi
0x1de2b:    lea     rdi, [rip + 0x4906]
0x1de32:    call    0x4e40
0x1de37:    mov     r15, qword ptr [rbx]
0x1de3a:    xor     ebx, ebx
0x1de3c:    test    r15, r15
0x1de3f:    jne     0x1de75
0x1de41:    jmp     0x1dec0

```

```

0x1de43:      nop      dword ptr [rax + rax]
0x1de48:      mov      rcx, r15
0x1de4b:      mov      rdx, r14
0x1de4e:      mov      esi, 1
0x1de53:      mov      rdi, r8
0x1de56:      xor      eax, eax
0x1de58:      mov      r13, rbp
0x1de5b:      call     0x4e60
0x1de60:      mov      rax, qword ptr [rsp + 8]
0x1de65:      add      rbx, 1
0x1de69:      add      rbp, r12
0x1de6c:      mov      r15, qword ptr [rax + rbx*8]
0x1de70:      test     r15, r15
0x1de73:      je       0x1dec0
0x1de75:      mov      r8, qword ptr [rip + 0xca04]
0x1de7c:      test     rbx, rbx
0x1de7f:      je       0x1de48
0x1de81:      mov      rdx, r12
0x1de84:      mov      rsi, rbp
0x1de87:      mov      rdi, r13
0x1de8a:      mov      qword ptr [rsp], r8
0x1de8e:      call     0x4af0
0x1de93:      mov      r8, qword ptr [rsp]
0x1de97:      test     eax, eax
0x1de99:      jne      0x1de48
0x1de9b:      mov      rcx, r15
0x1de9e:      lea      rdx, [rip + 0x48b2]
0x1dea5:      mov      rdi, r8
0x1dea8:      xor      eax, eax
0x1deaa:      mov      esi, 1
0x1deaf:      call     0x4e60
0x1deb4:      jmp      0x1de60
0x1deb6:      nop      word ptr cs:[rax + rax]
0x1dec0:      mov      rsi, qword ptr [rip + 0xc9b9]
0x1dec7:      add      rsp, 0x18
0x1decb:      mov      edi, 0xa
0x1ded0:      pop      rbx
0x1ded1:      pop      rbp
0x1ded2:      pop      r12
0x1ded4:      pop      r13
0x1ded6:      pop      r14
0x1ded8:      pop      r15
0x1deda:      jmp      0x4b40
0x1dedf:      nop
0x1dee0:      endbr64
0x1dee4:      push     r15
0x1dee6:      mov      r15, rsi
0x1dee9:      push     r14

```

```

0x1deeb:    mov     r14, r8
0x1deee:    push    r13
0x1def0:    mov     r13, rdi
0x1def3:    mov     rdi, r15
0x1def6:    push    r12
0x1def8:    mov     r12, rcx
0x1defb:    mov     rcx, r8
0x1defe:    push    rbp
0x1deff:    mov     rbp, rdx
0x1df02:    mov     rdx, r12
0x1df05:    push    rbx
0x1df06:    mov     rsi, rbp
0x1df09:    mov     rbx, r9
0x1df0c:    sub     rsp, 0x18
0x1df10:    call    0x1dc80
0x1df15:    test    rax, rax
0x1df18:    js      0x1df30
0x1df1a:    add     rsp, 0x18
0x1df1e:    pop     rbx
0x1df1f:    pop     rbp
0x1df20:    pop     r12
0x1df22:    pop     r13
0x1df24:    pop     r14
0x1df26:    pop     r15
0x1df28:    ret
0x1df29:    nop     dword ptr [rax]
0x1df30:    cmp     rax, -1
0x1df34:    lea     r9, [rip + 0x47c5]
0x1df3b:    mov     rsi, r13
0x1df3e:    mov     edi, 1
0x1df43:    lea     rax, [rip + 0x47d1]
0x1df4a:    cmovne  r9, rax
0x1df4e:    mov     qword ptr [rsp + 8], r9
0x1df53:    call    0x1e000
0x1df58:    mov     rdx, r15
0x1df5b:    mov     esi, 5
0x1df60:    xor     edi, edi
0x1df62:    mov     r13, rax
0x1df65:    call    0x1ee20
0x1df6a:    mov     rdx, qword ptr [rsp + 8]
0x1df6f:    mov     r8, r13
0x1df72:    xor     esi, esi
0x1df74:    mov     rcx, rax
0x1df77:    xor     edi, edi
0x1df79:    xor     eax, eax
0x1df7b:    call    0x4d50
0x1df80:    mov     rdx, r14
0x1df83:    mov     rsi, r12

```

```

0x1df86:      mov     rdi, rbp
0x1df89:      call   0x1ddf0
0x1df8e:      call   rbx
0x1df90:      mov     rax, -1
0x1df97:      jmp     0x1df1a
0x1df99:      nop     dword ptr [rax]
0x1dfa0:      endbr64
0x1dfa4:      push    r14
0x1dfa6:      push    r13
0x1dfa8:      push    r12
0x1dfaa:      push    rbp
0x1dfab:      push    rbx
0x1dfac:      mov     r14, qword ptr [rsi]
0x1dfaf:      test    r14, r14
0x1dfb2:      je      0x1dfe9
0x1dfb4:      mov     r12, rdi
0x1dfb7:      mov     r13, rcx
0x1dfba:      lea     rbx, [rsi + 8]
0x1dfbe:      mov     rbp, rdx
0x1dfc1:      jmp     0x1dfd7
0x1dfc3:      nop     dword ptr [rax + rax]
0x1dfc8:      mov     r14, qword ptr [rbx]
0x1dfcb:      add     rbp, r13
0x1dfce:      add     rbx, 8
0x1dfd2:      test    r14, r14
0x1dfd5:      je      0x1dfe9
0x1dfd7:      mov     rdx, r13
0x1dfda:      mov     rsi, rbp
0x1dfdd:      mov     rdi, r12
0x1dfe0:      call   0x4af0
0x1dfe5:      test    eax, eax
0x1dfe7:      jne     0x1dfc8
0x1dfe9:      pop     rbx
0x1dfea:      mov     rax, r14
0x1dfed:      pop     rbp
0x1dfee:      pop     r12
0x1dff0:      pop     r13
0x1dff2:      pop     r14
0x1dff4:      ret
0x1dff5:      nop     word ptr cs:[rax + rax]
0x1dfff:      nop
0x1e000:      endbr64
0x1e004:      mov     rdx, rsi
0x1e007:      mov     esi, 5
0x1e00c:      jmp     0x1ee20
0x1e011:      nop     word ptr cs:[rax + rax]
0x1e01c:      nop     dword ptr [rax]
0x1e020:      endbr64

```

```

0x1e024:    mov     rdx, rdi
0x1e027:    mov     esi, 5
0x1e02c:    xor     edi, edi
0x1e02e:    jmp     0x1ee20
0x1e033:    nop     word ptr cs:[rax + rax]
0x1e03d:    nop     dword ptr [rax]
0x1e040:    push    r15
0x1e042:    push    r14
0x1e044:    push    r13
0x1e046:    mov     r13, rcx
0x1e049:    push    r12
0x1e04b:    push    rbp
0x1e04c:    push    rbx
0x1e04d:    mov     ebx, r8d
0x1e050:    sub     rsp, 0x88
0x1e057:    mov     qword ptr [rsp + 8], rdi
0x1e05c:    mov     qword ptr [rsp + 0x30], rsi
0x1e061:    mov     qword ptr [rsp + 0x20], rdx
0x1e066:    mov     dword ptr [rsp + 0x10], r8d
0x1e06b:    mov     qword ptr [rsp + 0x18], r9
0x1e070:    mov     rax, qword ptr fs:[0x28]
0x1e079:    mov     qword ptr [rsp + 0x78], rax
0x1e07e:    xor     eax, eax
0x1e080:    call    0x49a0
0x1e085:    cmp     ebx, 4
0x1e088:    mov     r11, qword ptr [rsp + 0x20]
0x1e08d:    mov     r10, qword ptr [rsp + 0x30]
0x1e092:    mov     qword ptr [rsp + 0x28], rax
0x1e097:    jne     0x1e832
0x1e09d:    mov     qword ptr [rsp + 0x20], 0
0x1e0a6:    mov     r8d, 1
0x1e0ac:    xor     r9d, r9d
0x1e0af:    nop
0x1e0b0:    mov     rbx, r9
0x1e0b3:    xor     r12d, r12d
0x1e0b6:    nop     word ptr cs:[rax + rax]
0x1e0c0:    cmp     r13, r12
0x1e0c3:    setne   al
0x1e0c6:    cmp     r13, -1
0x1e0ca:    jne     0x1e0d4
0x1e0cc:    cmp     byte ptr [r11 + r12], 0
0x1e0d1:    setne   al
0x1e0d4:    test    al, al
0x1e0d6:    je      0x1e7ae
0x1e0dc:    lea     rsi, [r11 + r12]
0x1e0e0:    movzx   edx, byte ptr [rsi]
0x1e0e3:    test    r9, r9
0x1e0e6:    je      0x1e4b0

```



```

0x1e0ec:      test    r8b, r8b
0x1e0ef:      je      0x1e4b0
0x1e0f5:      lea     r14, [r12 + 1]
0x1e0fa:      mov     r15, r14
0x1e0fd:      cmp     r14, r13
0x1e100:      ja      0x1e120
0x1e102:      mov     rax, qword ptr [rsp + 0x20]
0x1e107:      cmp     byte ptr [rax], dl
0x1e109:      jne     0x1e120
0x1e10b:      cmp     rbx, r10
0x1e10e:      jae     0x1e11c
0x1e110:      mov     rax, qword ptr [rsp + 8]
0x1e115:      mov     byte ptr [rax + rbx], 0x5c
0x1e119:      movzx   edx, byte ptr [rsi]
0x1e11c:      add     rbx, 1
0x1e120:      mov     ebp, edx
0x1e122:      cmp     dl, 0x3f
0x1e125:      jg      0x1e458
0x1e12b:      test    dl, dl
0x1e12d:      js      0x1e240
0x1e133:      cmp     dl, 0x3f
0x1e136:      ja      0x1e240
0x1e13c:      lea     rcx, [rip + 0x464d]
0x1e143:      movzx   eax, dl
0x1e146:      movsxd  rax, dword ptr [rcx + rax*4]
0x1e14a:      add     rax, rcx
0x1e14d:      jmp     rax
0x1e150:      -----

0x1e1a2:      -----
0x1e1a3:      -----
0x1e1a4:      test    r8b, r8b
0x1e1a7:      je      0x1e9cf
0x1e1ad:      nop     dword ptr [rax]
0x1e1b0:      mov     edx, ebp
0x1e1b2:      mov     eax, 1
0x1e1b7:      mov     ecx, ebp
0x1e1b9:      mov     r12, r15
0x1e1bc:      shr     dl, 5
0x1e1bf:      shl     eax, cl
0x1e1c1:      movzx   edx, dl
0x1e1c4:      mov     rsi, qword ptr [rsp + 0x18]
0x1e1c9:      and     eax, dword ptr [rsi + rdx*4 + 4]
0x1e1cd:      je      0x1e1e1
0x1e1cf:      cmp     rbx, r10
0x1e1d2:      jae     0x1e1dd

```

```

0x1e1d4:    mov     rax, qword ptr [rsp + 8]
0x1e1d9:    mov     byte ptr [rax + rbx], 0x5c
0x1e1dd:    add     rbx, 1
0x1e1e1:    cmp     rbx, r10
0x1e1e4:    jae     0x1e1ef
0x1e1e6:    mov     rax, qword ptr [rsp + 8]
0x1e1eb:    mov     byte ptr [rax + rbx], bpl
0x1e1ef:    add     rbx, 1
0x1e1f3:    jmp     0x1e0c0
0x1e1f8:    cmp     dl, 0x7b
0x1e1fb:    je      0x1e72c
0x1e201:    cmp     dl, 0x7c
0x1e204:    jne     0x1e500
0x1e20a:    cmp     dword ptr [rsp + 0x10], 1
0x1e20f:    je      0x1e380
0x1e215:    lea     r15, [r12 + 1]
0x1e21a:    jmp     0x1e1a4
0x1e21c:    nop     dword ptr [rax]
0x1e220:    cmp     dl, 0x7d
0x1e223:    je      0x1e72c
0x1e229:    jle     0x1e727
0x1e22f:    cmp     dl, 0x7e
0x1e232:    je      0x1e365
0x1e238:    nop     dword ptr [rax + rax]
0x1e240:    cmp     qword ptr [rsp + 0x28], 1
0x1e246:    mov     byte ptr [rsp + 0x30], dl
0x1e24a:    jne     0x1e518
0x1e250:    mov     qword ptr [rsp + 0x50], r11
0x1e255:    mov     qword ptr [rsp + 0x48], r10
0x1e25a:    mov     byte ptr [rsp + 0x40], r8b
0x1e25f:    mov     qword ptr [rsp + 0x38], r9
0x1e264:    call    0x4f00
0x1e269:    movzx   edx, byte ptr [rsp + 0x30]
0x1e26e:    movzx   r8d, byte ptr [rsp + 0x40]
0x1e274:    mov     ecx, 1
0x1e279:    mov     rax, qword ptr [rax]
0x1e27c:    mov     r9, qword ptr [rsp + 0x38]
0x1e281:    mov     r10, qword ptr [rsp + 0x48]
0x1e286:    mov     r11, qword ptr [rsp + 0x50]
0x1e28b:    movzx   eax, word ptr [rax + rdx*2]
0x1e28f:    shr     ax, 0xe
0x1e293:    xor     eax, 1
0x1e296:    and     eax, r8d
0x1e299:    test    al, al
0x1e29b:    je      0x1e1a4
0x1e2a1:    mov     eax, r8d
0x1e2a4:    mov     rsi, qword ptr [rsp + 8]
0x1e2a9:    add     rcx, r12

```

```

0x1e2ac:      jmp      0x1e2c6
0x1e2ae:      nop
0x1e2b0:      cmp      rbx, r10
0x1e2b3:      jae      0x1e2b9
0x1e2b5:      mov      byte ptr [rsi + rbx], bpl
0x1e2b9:      movzx    ebp, byte ptr [r11 + r14]
0x1e2be:      add      rbx, 1
0x1e2c2:      add      r14, 1
0x1e2c6:      test     al, al
0x1e2c8:      je       0x1e30a
0x1e2ca:      cmp      rbx, r10
0x1e2cd:      jae      0x1e2d3
0x1e2cf:      mov      byte ptr [rsi + rbx], 0x5c
0x1e2d3:      lea      rdx, [rbx + 1]
0x1e2d7:      cmp      r10, rdx
0x1e2da:      jbe      0x1e2e8
0x1e2dc:      mov      edx, ebp
0x1e2de:      shr      dl, 6
0x1e2e1:      add      edx, 0x30
0x1e2e4:      mov      byte ptr [rsi + rbx + 1], dl
0x1e2e8:      lea      rdx, [rbx + 2]
0x1e2ec:      cmp      r10, rdx
0x1e2ef:      jbe      0x1e300
0x1e2f1:      mov      edx, ebp
0x1e2f3:      shr      dl, 3
0x1e2f6:      and      edx, 7
0x1e2f9:      add      edx, 0x30
0x1e2fc:      mov      byte ptr [rsi + rbx + 2], dl
0x1e300:      and      ebp, 7
0x1e303:      add      rbx, 3
0x1e307:      add      ebp, 0x30
0x1e30a:      mov      r12, r14
0x1e30d:      cmp      r14, rcx
0x1e310:      jb       0x1e2b0
0x1e312:      jmp      0x1e1e1
0x1e317:      nop      word ptr [rax + rax]
0x1e320:      mov      eax, dword ptr [rsp + 0x10]
0x1e324:      cmp      eax, 1
0x1e327:      je       0x1e380
0x1e329:      cmp      eax, 3
0x1e32c:      je       0x1e760
0x1e332:      lea      r15, [r12 + 1]
0x1e337:      mov      ebp, 0x3f
0x1e33c:      jmp      0x1e1a4
0x1e341:      mov      ebp, 0x72
0x1e346:      cmp      dword ptr [rsp + 0x10], 1
0x1e34b:      je       0x1e380
0x1e34d:      mov      r12, r14

```

```

0x1e350:    jmp     0x1e1cf
0x1e355:    mov     ebp, 0x62
0x1e35a:    jmp     0x1e34d
0x1e35c:    nop     dword ptr [rax]
0x1e360:    mov     ebp, 0x23
0x1e365:    test    r12, r12
0x1e368:    jne     0x1e215
0x1e36e:    cmp     dword ptr [rsp + 0x10], 1
0x1e373:    jne     0x1e215
0x1e379:    nop     dword ptr [rax]
0x1e380:    mov     qword ptr [rsp + 0x20], r11
0x1e385:    mov     qword ptr [rsp + 0x10], r10
0x1e38a:    call    0x49a0
0x1e38f:    mov     r10, qword ptr [rsp + 0x10]
0x1e394:    mov     r11, qword ptr [rsp + 0x20]
0x1e399:    mov     qword ptr [rsp + 0x28], rax
0x1e39e:    lea     rax, [rip + 0x43b7]
0x1e3a5:    mov     dword ptr [rsp + 0x10], 2
0x1e3ad:    mov     r9d, 1
0x1e3b3:    xor     r8d, r8d
0x1e3b6:    mov     qword ptr [rsp + 0x20], rax
0x1e3bb:    test    r10, r10
0x1e3be:    je      0x1e0b0
0x1e3c4:    mov     rax, qword ptr [rsp + 8]
0x1e3c9:    mov     byte ptr [rax], 0x27
0x1e3cc:    jmp     0x1e0b0
0x1e3d1:    mov     ebp, 0x74
0x1e3d6:    jmp     0x1e346
0x1e3db:    mov     ebp, 0x66
0x1e3e0:    jmp     0x1e34d
0x1e3e5:    mov     r12, r14
0x1e3e8:    mov     ebp, 0x76
0x1e3ed:    jmp     0x1e1cf
0x1e3f2:    mov     r12, r14
0x1e3f5:    cmp     r10, rbx
0x1e3f8:    jbe     0x1e403
0x1e3fa:    mov     rax, qword ptr [rsp + 8]
0x1e3ff:    mov     byte ptr [rax + rbx], 0x5c
0x1e403:    lea     rax, [rbx + 1]
0x1e407:    cmp     r10, rax
0x1e40a:    jbe     0x1e416
0x1e40c:    mov     rax, qword ptr [rsp + 8]
0x1e411:    mov     byte ptr [rax + rbx + 1], 0x30
0x1e416:    lea     rax, [rbx + 2]
0x1e41a:    cmp     r10, rax
0x1e41d:    jbe     0x1e429
0x1e41f:    mov     rax, qword ptr [rsp + 8]
0x1e424:    mov     byte ptr [rax + rbx + 2], 0x30

```

```

0x1e429:      add     rbx, 3
0x1e42d:      mov     eax, 0x10000
0x1e432:      mov     edx, 1
0x1e437:      mov     ebp, 0x30
0x1e43c:      jmp     0x1e1c4
0x1e441:      mov     ebp, 0x61
0x1e446:      jmp     0x1e34d
0x1e44b:      mov     ebp, 0x6e
0x1e450:      jmp     0x1e346
0x1e455:      nop     dword ptr [rax]
0x1e458:      cmp     dl, 0x7a
0x1e45b:      jg      0x1e220
0x1e461:      cmp     dl, 0x40
0x1e464:      je      0x1e240
0x1e46a:      lea     ecx, [rbp - 0x41]
0x1e46d:      mov     eax, 1
0x1e472:      movabs  rdi, 0x3fffffff53ffffff
0x1e47c:      shl     rax, cl
0x1e47f:      test    rax, rdi
0x1e482:      jne     0x1e1b0
0x1e488:      test    eax, 0xa4000000
0x1e48d:      jne     0x1e20a
0x1e493:      cmp     bpl, 0x5c
0x1e497:      jne     0x1e240
0x1e49d:      cmp     dword ptr [rsp + 0x10], 1
0x1e4a2:      je      0x1e380
0x1e4a8:      jmp     0x1e34d
0x1e4ad:      nop     dword ptr [rax]
0x1e4b0:      mov     ebp, edx
0x1e4b2:      cmp     dl, 0x3f
0x1e4b5:      jg      0x1e6d8
0x1e4bb:      test    dl, dl
0x1e4bd:      js      0x1e500
0x1e4bf:      cmp     dl, 0x3f
0x1e4c2:      ja      0x1e500
0x1e4c4:      lea     rdi, [rip + 0x43c5]
0x1e4cb:      movzx   eax, dl
0x1e4ce:      movsxd  rax, dword ptr [rdi + rax*4]
0x1e4d2:      add     rax, rdi
0x1e4d5:      jmp     rax
0x1e4d8:      -----

0x1e4de:      -----
0x1e4df:      -----
0x1e4e0:      cmp     dl, 0x7d
0x1e4e3:      je      0x1e72c

```

```

0x1e4e9:    jle     0x1e1f8
0x1e4ef:    cmp     dl, 0x7e
0x1e4f2:    je      0x1e365
0x1e4f8:    nop     dword ptr [rax + rax]
0x1e500:    lea     r14, [r12 + 1]
0x1e505:    cmp     qword ptr [rsp + 0x28], 1
0x1e50b:    mov     byte ptr [rsp + 0x30], dl
0x1e50f:    mov     r15, r14
0x1e512:    je      0x1e250
0x1e518:    mov     qword ptr [rsp + 0x70], 0
0x1e521:    lea     rcx, [rsp + 0x70]
0x1e526:    cmp     r13, -1
0x1e52a:    jne     0x1e574
0x1e52c:    mov     rdi, r11
0x1e52f:    mov     qword ptr [rsp + 0x58], r10
0x1e534:    mov     byte ptr [rsp + 0x50], r8b
0x1e539:    mov     qword ptr [rsp + 0x48], r9
0x1e53e:    mov     qword ptr [rsp + 0x40], rsi
0x1e543:    mov     qword ptr [rsp + 0x38], rcx
0x1e548:    mov     qword ptr [rsp + 0x30], r11
0x1e54d:    call    0x49b0
0x1e552:    mov     r10, qword ptr [rsp + 0x58]
0x1e557:    movzx   r8d, byte ptr [rsp + 0x50]
0x1e55d:    mov     r9, qword ptr [rsp + 0x48]
0x1e562:    mov     rsi, qword ptr [rsp + 0x40]
0x1e567:    mov     r13, rax
0x1e56a:    mov     rcx, qword ptr [rsp + 0x38]
0x1e56f:    mov     r11, qword ptr [rsp + 0x30]
0x1e574:    mov     rdx, r13
0x1e577:    lea     rdi, [rsp + 0x6c]
0x1e57c:    mov     qword ptr [rsp + 0x48], r11
0x1e581:    sub     rdx, r12
0x1e584:    mov     qword ptr [rsp + 0x40], r10
0x1e589:    mov     byte ptr [rsp + 0x38], r8b
0x1e58e:    mov     qword ptr [rsp + 0x30], r9
0x1e593:    mov     qword ptr [rsp + 0x58], rdx
0x1e598:    mov     qword ptr [rsp + 0x50], rsi
0x1e59d:    call    0x49f0
0x1e5a2:    mov     r9, qword ptr [rsp + 0x30]
0x1e5a7:    movzx   r8d, byte ptr [rsp + 0x38]
0x1e5ad:    test    rax, rax
0x1e5b0:    mov     r10, qword ptr [rsp + 0x40]
0x1e5b5:    mov     r11, qword ptr [rsp + 0x48]
0x1e5ba:    mov     rcx, rax
0x1e5bd:    je      0x1e1a4
0x1e5c3:    cmp     rax, -1
0x1e5c7:    je      0x1e828
0x1e5cd:    cmp     rax, -2

```

```

0x1e5d1:    mov     rsi, qword ptr [rsp + 0x50]
0x1e5d6:    mov     rdx, qword ptr [rsp + 0x58]
0x1e5db:    je      0x1e93b
0x1e5e1:    cmp     dword ptr [rsp + 0x10], 1
0x1e5e6:    je      0x1e95f
0x1e5ec:    mov     edi, dword ptr [rsp + 0x6c]
0x1e5f0:    mov     qword ptr [rsp + 0x50], r11
0x1e5f5:    mov     qword ptr [rsp + 0x48], r10
0x1e5fa:    mov     byte ptr [rsp + 0x40], r8b
0x1e5ff:    mov     qword ptr [rsp + 0x38], r9
0x1e604:    mov     qword ptr [rsp + 0x30], rcx
0x1e609:    call    0x4e80
0x1e60e:    movzx   r8d, byte ptr [rsp + 0x40]
0x1e614:    mov     rcx, qword ptr [rsp + 0x30]
0x1e619:    test    eax, eax
0x1e61b:    mov     r9, qword ptr [rsp + 0x38]
0x1e620:    mov     r10, qword ptr [rsp + 0x48]
0x1e625:    sete    al
0x1e628:    mov     r11, qword ptr [rsp + 0x50]
0x1e62d:    and     eax, r8d
0x1e630:    cmp     rcx, 1
0x1e634:    ja      0x1e2a4
0x1e63a:    jmp     0x1e299
0x1e63f:    mov     ebp, 0xc
0x1e644:    mov     eax, 0x66
0x1e649:    nop     dword ptr [rax]
0x1e650:    add     r12, 1
0x1e654:    test    r8b, r8b
0x1e657:    je      0x1e1e1
0x1e65d:    mov     ebp, eax
0x1e65f:    jmp     0x1e1cf
0x1e664:    nop     dword ptr [rax]
0x1e668:    mov     ebp, 0xb
0x1e66d:    mov     eax, 0x76
0x1e672:    jmp     0x1e650
0x1e674:    mov     ebp, 9
0x1e679:    mov     eax, 0x74
0x1e67e:    cmp     dword ptr [rsp + 0x10], 1
0x1e683:    jne     0x1e650
0x1e685:    jmp     0x1e380
0x1e68a:    nop     word ptr [rax + rax]
0x1e690:    add     r12, 1
0x1e694:    test    r8b, r8b
0x1e697:    jne     0x1e3f5
0x1e69d:    xor     ebp, ebp
0x1e69f:    jmp     0x1e1e1
0x1e6a4:    mov     ebp, 8
0x1e6a9:    mov     eax, 0x62

```

```

0x1e6ae:      jmp      0x1e650
0x1e6b0:      mov      ebp, 0xa
0x1e6b5:      mov      eax, 0x6e
0x1e6ba:      jmp      0x1e67e
0x1e6bc:      mov      ebp, 7
0x1e6c1:      mov      eax, 0x61
0x1e6c6:      jmp      0x1e650
0x1e6c8:      mov      ebp, 0xd
0x1e6cd:      mov      eax, 0x72
0x1e6d2:      jmp      0x1e67e
0x1e6d4:      nop      dword ptr [rax]
0x1e6d8:      cmp      dl, 0x7a
0x1e6db:      jg       0x1e4e0
0x1e6e1:      cmp      dl, 0x40
0x1e6e4:      je       0x1e500
0x1e6ea:      lea      ecx, [rbp - 0x41]
0x1e6ed:      mov      eax, 1
0x1e6f2:      movabs   rdi, 0x3fffffff53ffffff
0x1e6fc:      shl      rax, cl
0x1e6ff:      test     rax, rdi
0x1e702:      jne      0x1e215
0x1e708:      test     eax, 0xa4000000
0x1e70d:      jne      0x1e20a
0x1e713:      cmp      bpl, 0x5c
0x1e717:      jne      0x1e500
0x1e71d:      mov      eax, 0x5c
0x1e722:      jmp      0x1e67e
0x1e727:      cmp      dl, 0x7b
0x1e72a:      jne      0x1e750
0x1e72c:      cmp      r13, 1
0x1e730:      setne    al
0x1e733:      cmp      r13, -1
0x1e737:      jne      0x1e741
0x1e739:      cmp      byte ptr [r11 + 1], 0
0x1e73e:      setne    al
0x1e741:      test     al, al
0x1e743:      jne      0x1e215
0x1e749:      jmp      0x1e365
0x1e74e:      nop
0x1e750:      cmp      dl, 0x7c
0x1e753:      je       0x1e20a
0x1e759:      jmp      0x1e240
0x1e75e:      nop
0x1e760:      lea      rax, [r12 + 2]
0x1e765:      lea      r15, [r12 + 1]
0x1e76a:      mov      ebp, 0x3f
0x1e76f:      cmp      rax, r13
0x1e772:      jae      0x1e1a4

```



```

0x1e778:      cmp     byte ptr [r11 + r12 + 1], 0x3f
0x1e77e:      jne     0x1e1a4
0x1e784:      movzx   ebp, byte ptr [r11 + r12 + 2]
0x1e78a:      cmp     bpl, 0x3e
0x1e78e:      ja      0x1e7a4
0x1e790:      movabs   rax, 0x7000a38200000000
0x1e79a:      bt      rax, rbp
0x1e79e:      jb      0x1e8f9
0x1e7a4:      mov     ebp, 0x3f
0x1e7a9:      jmp     0x1e1a4
0x1e7ae:      test    r12, r12
0x1e7b1:      jne     0x1e7be
0x1e7b3:      cmp     dword ptr [rsp + 0x10], 1
0x1e7b8:      je      0x1e380
0x1e7be:      mov     rsi, qword ptr [rsp + 0x20]
0x1e7c3:      mov     rax, rbx
0x1e7c6:      test    rsi, rsi
0x1e7c9:      je      0x1e7f4
0x1e7cb:      movzx   edx, byte ptr [rsi]
0x1e7ce:      test    dl, dl
0x1e7d0:      je      0x1e7f4
0x1e7d2:      sub     rsi, rbx
0x1e7d5:      mov     rcx, rsi
0x1e7d8:      mov     rsi, qword ptr [rsp + 8]
0x1e7dd:      nop     dword ptr [rax]
0x1e7e0:      cmp     r10, rax
0x1e7e3:      jbe     0x1e7e8
0x1e7e5:      mov     byte ptr [rsi + rax], dl
0x1e7e8:      add     rax, 1
0x1e7ec:      movzx   edx, byte ptr [rcx + rax]
0x1e7f0:      test    dl, dl
0x1e7f2:      jne     0x1e7e0
0x1e7f4:      cmp     rax, r10
0x1e7f7:      jae     0x1e802
0x1e7f9:      mov     rsi, qword ptr [rsp + 8]
0x1e7fe:      mov     byte ptr [rsi + rax], 0
0x1e802:      mov     rdx, qword ptr [rsp + 0x78]
0x1e807:      sub     rdx, qword ptr fs:[0x28]
0x1e810:      jne     0x1e9ca
0x1e816:      add     rsp, 0x88
0x1e81d:      pop     rbx
0x1e81e:      pop     rbp
0x1e81f:      pop     r12
0x1e821:      pop     r13
0x1e823:      pop     r14
0x1e825:      pop     r15
0x1e827:      ret
0x1e828:      mov     eax, r8d

```

```

0x1e82b:    xor     ecx, ecx
0x1e82d:    jmp     0x1e299
0x1e832:    mov     eax, dword ptr [rsp + 0x10]
0x1e836:    jbe     0x1e88a
0x1e838:    sub     eax, 5
0x1e83b:    cmp     eax, 1
0x1e83e:    ja      0x1e8c8
0x1e844:    cmp     dword ptr [rsp + 0x10], 6
0x1e849:    mov     edx, 0x60
0x1e84e:    mov     eax, 0x22
0x1e853:    cmovne  eax, edx
0x1e856:    lea     rcx, [rip + 0x2a3b]
0x1e85d:    lea     rdx, [rip + 0x3ef8]
0x1e864:    cmov     rdx, rcx
0x1e868:    mov     qword ptr [rsp + 0x20], rdx
0x1e86d:    test    r10, r10
0x1e870:    je      0x1e879
0x1e872:    mov     rsi, qword ptr [rsp + 8]
0x1e877:    mov     byte ptr [rsi], al
0x1e879:    mov     r8d, 1
0x1e87f:    mov     r9d, 1
0x1e885:    jmp     0x1e0b0
0x1e88a:    cmp     eax, 2
0x1e88d:    je      0x1e39e
0x1e893:    cmp     eax, 3
0x1e896:    jne     0x1e8c8
0x1e898:    test    r10, r10
0x1e89b:    je      0x1e8dc
0x1e89d:    mov     rax, qword ptr [rsp + 8]
0x1e8a2:    mov     r8d, 1
0x1e8a8:    mov     r9d, 1
0x1e8ae:    mov     byte ptr [rax], 0x22
0x1e8b1:    lea     rax, [rip + 0x29e0]
0x1e8b8:    mov     qword ptr [rsp + 0x20], rax
0x1e8bd:    jmp     0x1e0b0
0x1e8c2:    nop     word ptr [rax + rax]
0x1e8c8:    mov     qword ptr [rsp + 0x20], 0
0x1e8d1:    xor     r8d, r8d
0x1e8d4:    xor     r9d, r9d
0x1e8d7:    jmp     0x1e0b0
0x1e8dc:    lea     rax, [rip + 0x29b5]
0x1e8e3:    mov     r8d, 1
0x1e8e9:    mov     r9d, 1
0x1e8ef:    mov     qword ptr [rsp + 0x20], rax
0x1e8f4:    jmp     0x1e0b0
0x1e8f9:    cmp     r10, rbx
0x1e8fc:    jbe     0x1e907
0x1e8fe:    mov     rax, qword ptr [rsp + 8]

```

```

0x1e903:    mov     byte ptr [rax + rbx], 0x3f
0x1e907:    lea     rax, [rbx + 1]
0x1e90b:    cmp     r10, rax
0x1e90e:    jbe     0x1e91a
0x1e910:    mov     rax, qword ptr [rsp + 8]
0x1e915:    mov     byte ptr [rax + rbx + 1], 0x5c
0x1e91a:    lea     rax, [rbx + 2]
0x1e91e:    cmp     r10, rax
0x1e921:    jbe     0x1e92d
0x1e923:    mov     rax, qword ptr [rsp + 8]
0x1e928:    mov     byte ptr [rax + rbx + 2], 0x3f
0x1e92d:    add     rbx, 3
0x1e931:    lea     r15, [r12 + 3]
0x1e936:    jmp     0x1e1a4
0x1e93b:    xor     ecx, ecx
0x1e93d:    cmp     r13, r12
0x1e940:    ja      0x1e951
0x1e942:    jmp     0x1e9c2
0x1e944:    nop     dword ptr [rax]
0x1e948:    add     rcx, 1
0x1e94c:    cmp     rdx, rcx
0x1e94f:    je      0x1e957
0x1e951:    cmp     byte ptr [rsi + rcx], 0
0x1e955:    jne     0x1e948
0x1e957:    mov     eax, r8d
0x1e95a:    jmp     0x1e630
0x1e95f:    cmp     rax, 1
0x1e963:    je      0x1e5ec
0x1e969:    lea     rdx, [r11 + r14]
0x1e96d:    add     rsi, rax
0x1e970:    jmp     0x1e985
0x1e972:    nop     word ptr [rax + rax]
0x1e978:    add     rdx, 1
0x1e97c:    cmp     rsi, rdx
0x1e97f:    je      0x1e5ec
0x1e985:    movzx   eax, byte ptr [rdx]
0x1e988:    sub     eax, 0x5b
0x1e98b:    cmp     al, 0x21
0x1e98d:    ja      0x1e978
0x1e98f:    movabs  rdi, 0x20000002b
0x1e999:    bt      rdi, rax
0x1e99d:    jae     0x1e978
0x1e99f:    mov     qword ptr [rsp + 0x20], r10
0x1e9a4:    mov     qword ptr [rsp + 0x10], r11
0x1e9a9:    call    0x49a0
0x1e9ae:    mov     r11, qword ptr [rsp + 0x10]
0x1e9b3:    mov     r10, qword ptr [rsp + 0x20]
0x1e9b8:    mov     qword ptr [rsp + 0x28], rax

```

```

0x1e9bd:      jmp      0x1e39e
0x1e9c2:      mov     eax, r8d
0x1e9c5:      jmp     0x1e299
0x1e9ca:      call    0x49d0
0x1e9cf:      mov     r12, r15
0x1e9d2:      jmp     0x1e1e1
0x1e9d7:      nop     word ptr [rax + rax]
0x1e9e0:      push    r15
0x1e9e2:      mov     r15d, edi
0x1e9e5:      push    r14
0x1e9e7:      push    r13
0x1e9e9:      mov     r13, rdx
0x1e9ec:      push    r12
0x1e9ee:      mov     r12, rsi
0x1e9f1:      push    rbp
0x1e9f2:      mov     rbp, rcx
0x1e9f5:      push    rbx
0x1e9f6:      sub     rsp, 0x28
0x1e9fa:      call    0x4860
0x1e9ff:      mov     r14, qword ptr [rip + 0xbe0a]
0x1ea06:      mov     rbx, rax
0x1ea09:      mov     eax, dword ptr [rax]
0x1ea0b:      mov     dword ptr [rsp + 0xc], eax
0x1ea0f:      test    r15d, r15d
0x1ea12:      js      0x4f2a
0x1ea18:      cmp     dword ptr [rip + 0xbde5], r15d
0x1ea1f:      mov     r10d, dword ptr [rsp + 0xc]
0x1ea24:      ja      0x1ea89
0x1ea26:      lea     edx, [r15 + 1]
0x1ea2a:      lea     rax, [rip + 0xbdef]
0x1ea31:      mov     rsi, rdx
0x1ea34:      mov     dword ptr [rsp + 0x10], edx
0x1ea38:      shl     rsi, 4
0x1ea3c:      cmp     r14, rax
0x1ea3f:      je      0x1eb40
0x1ea45:      mov     rdi, r14
0x1ea48:      mov     qword ptr [rsp + 0x18], rdx
0x1ea4d:      call    0x1d690
0x1ea52:      mov     rdx, qword ptr [rsp + 0x18]
0x1ea57:      mov     qword ptr [rip + 0xbdb2], rax
0x1ea5e:      mov     r14, rax
0x1ea61:      mov     edi, dword ptr [rip + 0xbd9d]
0x1ea67:      xor     esi, esi
0x1ea69:      sub     rdx, rdi
0x1ea6c:      shl     rdi, 4
0x1ea70:      shl     rdx, 4
0x1ea74:      add     rdi, r14
0x1ea77:      call    0x4a50

```

```

0x1ea7c:    mov     eax, dword ptr [rsp + 0x10]
0x1ea80:    mov     r10d, dword ptr [rbx]
0x1ea83:    mov     dword ptr [rip + 0xbd7b], eax
0x1ea89:    movsxd  r8, r15d
0x1ea8c:    mov     r9, rbp
0x1ea8f:    mov     rcx, r13
0x1ea92:    mov     rdx, r12
0x1ea95:    shl     r8, 4
0x1ea99:    mov     dword ptr [rsp + 0x18], r10d
0x1ea9e:    lea     r15, [r14 + r8]
0x1eaa2:    mov     r8d, dword ptr [rbp]
0x1eaa6:    mov     r14, qword ptr [r15 + 8]
0x1eaaa:    mov     rsi, qword ptr [r15]
0x1eaad:    mov     rdi, r14
0x1eab0:    mov     qword ptr [rsp + 0x10], rsi
0x1eab5:    call    0x1e040
0x1eaba:    mov     r10d, dword ptr [rsp + 0x18]
0x1eabf:    mov     rsi, qword ptr [rsp + 0x10]
0x1eac4:    mov     dword ptr [rbx], r10d
0x1eac7:    cmp     rax, rsi
0x1eaca:    jnb     0x1eb1f
0x1eacc:    lea     rsi, [rax + 1]
0x1ead0:    lea     rax, [rip + 0xd089]
0x1ead7:    mov     qword ptr [r15], rsi
0x1eada:    cmp     r14, rax
0x1eadd:    je      0x1eaf1
0x1eadf:    mov     rdi, r14
0x1eae2:    mov     qword ptr [rsp + 0x10], rsi
0x1eae7:    call    0x4800
0x1eaec:    mov     rsi, qword ptr [rsp + 0x10]
0x1eaf1:    mov     rdi, rsi
0x1eaf4:    mov     qword ptr [rsp + 0x10], rsi
0x1eaf9:    call    0x1d670
0x1eafe:    mov     r8d, dword ptr [rbp]
0x1eb02:    mov     r9, rbp
0x1eb05:    mov     rcx, r13
0x1eb08:    mov     qword ptr [r15 + 8], rax
0x1eb0c:    mov     rsi, qword ptr [rsp + 0x10]
0x1eb11:    mov     rdx, r12
0x1eb14:    mov     rdi, rax
0x1eb17:    mov     r14, rax
0x1eb1a:    call    0x1e040
0x1eb1f:    mov     eax, dword ptr [rsp + 0xc]
0x1eb23:    mov     dword ptr [rbx], eax
0x1eb25:    add     rsp, 0x28
0x1eb29:    mov     rax, r14
0x1eb2c:    pop     rbx
0x1eb2d:    pop     rbp

```

```

0x1eb2e:    pop     r12
0x1eb30:    pop     r13
0x1eb32:    pop     r14
0x1eb34:    pop     r15
0x1eb36:    ret
0x1eb37:    nop     word ptr [rax + rax]
0x1eb40:    xor     edi, edi
0x1eb42:    mov     qword ptr [rsp + 0x18], rdx
0x1eb47:    call    0x1d690
0x1eb4c:    movdqa  xmm0, xmmword ptr [rip + 0xbccc]
0x1eb54:    mov     rdx, qword ptr [rsp + 0x18]
0x1eb59:    mov     qword ptr [rip + 0xbcb0], rax
0x1eb60:    mov     r14, rax
0x1eb63:    movups  xmmword ptr [rax], xmm0
0x1eb66:    jmp     0x1ea61
0x1eb6b:    nop     dword ptr [rax + rax]
0x1eb70:    endbr64
0x1eb74:    push    r12
0x1eb76:    push    rbp
0x1eb77:    mov     rbp, rdi
0x1eb7a:    push    rbx
0x1eb7b:    call    0x4860
0x1eb80:    test    rbp, rbp
0x1eb83:    mov     esi, 0x24
0x1eb88:    mov     r12d, dword ptr [rax]
0x1eb8b:    mov     rbx, rax
0x1eb8e:    lea     rax, [rip + 0xd0cb]
0x1eb95:    cmov     rbp, rax
0x1eb99:    mov     rdi, rbp
0x1eb9c:    call    0x1d760
0x1eba1:    mov     dword ptr [rbx], r12d
0x1eba4:    pop     rbx
0x1eba5:    pop     rbp
0x1eba6:    pop     r12
0x1eba8:    ret
0x1eba9:    nop     dword ptr [rax]
0x1ebb0:    endbr64
0x1ebb4:    test    rdi, rdi
0x1ebb7:    lea     rax, [rip + 0xd0a2]
0x1ebbe:    cmov     rdi, rax
0x1ebc2:    mov     eax, dword ptr [rdi]
0x1ebc4:    ret
0x1ebc5:    nop     word ptr cs:[rax + rax]
0x1ebd0:    endbr64
0x1ebd4:    test    rdi, rdi
0x1ebd7:    lea     rax, [rip + 0xd082]
0x1ebde:    cmov     rdi, rax
0x1ebe2:    mov     dword ptr [rdi], esi

```

```

0x1ebe4:      ret
0x1ebe5:      nop      word ptr cs:[rax + rax]
0x1ebf0:      endbr64
0x1ebf4:      test     rdi, rdi
0x1ebf7:      lea      rax, [rip + 0xd062]
0x1ebfe:      mov      ecx, esi
0x1ec00:      cmov     rdi, rax
0x1ec04:      mov      eax, esi
0x1ec06:      and      ecx, 0x1f
0x1ec09:      shr      al, 5
0x1ec0c:      movzx    eax, al
0x1ec0f:      lea      rsi, [rdi + rax*4 + 4]
0x1ec14:      mov      edi, dword ptr [rsi]
0x1ec16:      mov      eax, edi
0x1ec18:      shr      eax, cl
0x1ec1a:      xor      edx, eax
0x1ec1c:      and      eax, 1
0x1ec1f:      and      edx, 1
0x1ec22:      shl      edx, cl
0x1ec24:      xor      edx, edi
0x1ec26:      mov      dword ptr [rsi], edx
0x1ec28:      ret
0x1ec29:      nop      dword ptr [rax]
0x1ec30:      endbr64
0x1ec34:      push     r15
0x1ec36:      lea      rax, [rip + 0xd023]
0x1ec3d:      push     r14
0x1ec3f:      mov      r14, rdx
0x1ec42:      push     r13
0x1ec44:      mov      r13, rsi
0x1ec47:      push     r12
0x1ec49:      mov      r12, rdi
0x1ec4c:      push     rbp
0x1ec4d:      push     rbx
0x1ec4e:      mov      rbx, r8
0x1ec51:      sub      rsp, 0x18
0x1ec55:      test     r8, r8
0x1ec58:      cmov     rbx, rax
0x1ec5c:      mov      qword ptr [rsp + 8], rcx
0x1ec61:      call     0x4860
0x1ec66:      mov      rcx, qword ptr [rsp + 8]
0x1ec6b:      mov      rdx, r14
0x1ec6e:      mov      rsi, r13
0x1ec71:      mov      r15d, dword ptr [rax]
0x1ec74:      mov      r8d, dword ptr [rbx]
0x1ec77:      mov      rbp, rax
0x1ec7a:      mov      r9, rbx
0x1ec7d:      mov      rdi, r12

```

```

0x1ec80:      call    0x1e040
0x1ec85:      mov     dword ptr [rbp], r15d
0x1ec89:      add     rsp, 0x18
0x1ec8d:      pop     rbx
0x1ec8e:      pop     rbp
0x1ec8f:      pop     r12
0x1ec91:      pop     r13
0x1ec93:      pop     r14
0x1ec95:      pop     r15
0x1ec97:      ret
0x1ec98:      nop     dword ptr [rax + rax]
0x1eca0:      endbr64
0x1eca4:      push    r15
0x1eca6:      mov     r15, rdi
0x1eca9:      push    r14
0x1ecab:      mov     r14, rsi
0x1ecae:      push    r13
0x1ecb0:      push    r12
0x1ecb2:      push    rbp
0x1ecb3:      push    rbx
0x1ecb4:      mov     rbx, rdx
0x1ecb7:      sub     rsp, 0x18
0x1ecbb:      call    0x4860
0x1ecc0:      test    rbx, rbx
0x1ecc3:      mov     rcx, r14
0x1ecc6:      mov     rdx, r15
0x1ecc9:      mov     r12d, dword ptr [rax]
0x1eccc:      mov     rbp, rax
0x1eccf:      lea     rax, [rip + 0xcf8a]
0x1ecd6:      cmov     rbx, rax
0x1ecda:      xor     esi, esi
0x1ecdc:      xor     edi, edi
0x1ecde:      mov     r8d, dword ptr [rbx]
0x1ece1:      mov     r9, rbx
0x1ece4:      call    0x1e040
0x1ece9:      mov     dword ptr [rbp], r12d
0x1eced:      lea     r13, [rax + 1]
0x1ecf1:      mov     dword ptr [rsp + 0xc], r12d
0x1ecf6:      mov     rdi, r13
0x1ecf9:      call    0x1d670
0x1ecfe:      mov     r8d, dword ptr [rbx]
0x1ed01:      mov     r9, rbx
0x1ed04:      mov     rcx, r14
0x1ed07:      mov     rdx, r15
0x1ed0a:      mov     rsi, r13
0x1ed0d:      mov     rdi, rax
0x1ed10:      mov     r12, rax
0x1ed13:      call    0x1e040

```



```

0x1ed18:    mov     eax, dword ptr [rsp + 0xc]
0x1ed1c:    mov     dword ptr [rbp], eax
0x1ed1f:    add     rsp, 0x18
0x1ed23:    mov     rax, r12
0x1ed26:    pop     rbx
0x1ed27:    pop     rbp
0x1ed28:    pop     r12
0x1ed2a:    pop     r13
0x1ed2c:    pop     r14
0x1ed2e:    pop     r15
0x1ed30:    ret
0x1ed31:    nop     word ptr cs:[rax + rax]
0x1ed3c:    nop     dword ptr [rax]
0x1ed40:    endbr64
0x1ed44:    mov     eax, dword ptr [rip + 0xbaba]
0x1ed4a:    push    r12
0x1ed4c:    mov     r12, qword ptr [rip + 0xbabd]
0x1ed53:    push    rbp
0x1ed54:    push    rbx
0x1ed55:    cmp     eax, 1
0x1ed58:    jbe     0x1ed81
0x1ed5a:    sub     eax, 2
0x1ed5d:    lea     rbx, [r12 + 0x18]
0x1ed62:    shl     rax, 4
0x1ed66:    lea     rbp, [r12 + rax + 0x28]
0x1ed6b:    nop     dword ptr [rax + rax]
0x1ed70:    mov     rdi, qword ptr [rbx]
0x1ed73:    add     rbx, 0x10
0x1ed77:    call    0x4800
0x1ed7c:    cmp     rbx, rbp
0x1ed7f:    jne     0x1ed70
0x1ed81:    mov     rdi, qword ptr [r12 + 8]
0x1ed86:    lea     rbx, [rip + 0xcdd3]
0x1ed8d:    cmp     rdi, rbx
0x1ed90:    je      0x1eda9
0x1ed92:    call    0x4800
0x1ed97:    mov     qword ptr [rip + 0xba8a], rbx
0x1ed9e:    mov     qword ptr [rip + 0xba77], 0x100
0x1eda9:    lea     rbx, [rip + 0xba70]
0x1edb0:    cmp     r12, rbx
0x1edb3:    je      0x1edc4
0x1edb5:    mov     rdi, r12
0x1edb8:    call    0x4800
0x1edbd:    mov     qword ptr [rip + 0xba4c], rbx
0x1edc4:    mov     dword ptr [rip + 0xba36], 1
0x1edce:    pop     rbx
0x1edcf:    pop     rbp
0x1edd0:    pop     r12

```

```

0x1edd2:      ret
0x1edd3:      nop      word ptr cs:[rax + rax]
0x1edde:      nop
0x1ede0:      endbr64
0x1ede4:      lea      rcx, [rip + 0xce75]
0x1edeb:      mov      rdx, -1
0x1edf2:      jmp      0x1e9e0
0x1edf7:      nop      word ptr [rax + rax]
0x1ee00:      endbr64
0x1ee04:      mov      rsi, rdi
0x1ee07:      lea      rcx, [rip + 0xce52]
0x1ee0e:      mov      rdx, -1
0x1ee15:      xor      edi, edi
0x1ee17:      jmp      0x1e9e0
0x1ee1c:      nop      dword ptr [rax]
0x1ee20:      endbr64
0x1ee24:      sub      rsp, 0x68
0x1ee28:      mov      r8d, esi
0x1ee2b:      pxor     xmm0, xmm0
0x1ee2f:      mov      rsi, rdx
0x1ee32:      mov      rax, qword ptr fs:[0x28]
0x1ee3b:      mov      qword ptr [rsp + 0x58], rax
0x1ee40:      xor      eax, eax
0x1ee42:      movups   xmmword ptr [rsp + 0x34], xmm0
0x1ee47:      mov      rdx, -1
0x1ee4e:      mov      rcx, rsp
0x1ee51:      movups   xmmword ptr [rsp + 0x44], xmm0
0x1ee56:      movdqa   xmm2, xmmword ptr [rsp + 0x40]
0x1ee5c:      mov      eax, dword ptr [rsp + 0x50]
0x1ee60:      mov      dword ptr [rsp + 0x30], r8d
0x1ee65:      movdqa   xmm1, xmmword ptr [rsp + 0x30]
0x1ee6b:      mov      dword ptr [rsp + 0x20], eax
0x1ee6f:      movaps   xmmword ptr [rsp], xmm1
0x1ee73:      movaps   xmmword ptr [rsp + 0x10], xmm2
0x1ee78:      call     0x1e9e0
0x1ee7d:      mov      rdx, qword ptr [rsp + 0x58]
0x1ee82:      sub      rdx, qword ptr fs:[0x28]
0x1ee8b:      jne      0x1ee92
0x1ee8d:      add      rsp, 0x68
0x1ee91:      ret
0x1ee92:      call     0x49d0
0x1ee97:      nop      word ptr [rax + rax]
0x1eea0:      endbr64
0x1eea4:      sub      rsp, 0x68
0x1eea8:      mov      r8d, esi
0x1eeab:      pxor     xmm0, xmm0
0x1eeaf:      mov      rsi, rdx
0x1eeb2:      mov      rax, qword ptr fs:[0x28]

```

```

0x1eebb:    mov     qword ptr [rsp + 0x58], rax
0x1eec0:    xor     eax, eax
0x1eec2:    movups  xmmword ptr [rsp + 0x34], xmm0
0x1eec7:    mov     rdx, rcx
0x1eeca:    mov     rcx, rsp
0x1eecd:    movups  xmmword ptr [rsp + 0x44], xmm0
0x1eed2:    movdqa  xmm2, xmmword ptr [rsp + 0x40]
0x1eed8:    mov     eax, dword ptr [rsp + 0x50]
0x1eedc:    mov     dword ptr [rsp + 0x30], r8d
0x1eee1:    movdqa  xmm1, xmmword ptr [rsp + 0x30]
0x1eee7:    mov     dword ptr [rsp + 0x20], eax
0x1eeeb:    movaps  xmmword ptr [rsp], xmm1
0x1eeef:    movaps  xmmword ptr [rsp + 0x10], xmm2
0x1eef4:    call    0x1e9e0
0x1eef9:    mov     rdx, qword ptr [rsp + 0x58]
0x1eefe:    sub     rdx, qword ptr fs:[0x28]
0x1ef07:    jne     0x1ef0e
0x1ef09:    add     rsp, 0x68
0x1ef0d:    ret
0x1ef0e:    call    0x49d0
0x1ef13:    nop     word ptr cs:[rax + rax]
0x1ef1e:    nop
0x1ef20:    endbr64
0x1ef24:    sub     rsp, 0x68
0x1ef28:    pxor    xmm0, xmm0
0x1ef2c:    mov     rdx, -1
0x1ef33:    mov     rax, qword ptr fs:[0x28]
0x1ef3c:    mov     qword ptr [rsp + 0x58], rax
0x1ef41:    xor     eax, eax
0x1ef43:    mov     dword ptr [rsp + 0x30], edi
0x1ef47:    mov     rcx, rsp
0x1ef4a:    xor     edi, edi
0x1ef4c:    movups  xmmword ptr [rsp + 0x34], xmm0
0x1ef51:    movdqa  xmm1, xmmword ptr [rsp + 0x30]
0x1ef57:    movups  xmmword ptr [rsp + 0x44], xmm0
0x1ef5c:    movdqa  xmm2, xmmword ptr [rsp + 0x40]
0x1ef62:    mov     eax, dword ptr [rsp + 0x50]
0x1ef66:    movaps  xmmword ptr [rsp], xmm1
0x1ef6a:    mov     dword ptr [rsp + 0x20], eax
0x1ef6e:    movaps  xmmword ptr [rsp + 0x10], xmm2
0x1ef73:    call    0x1e9e0
0x1ef78:    mov     rdx, qword ptr [rsp + 0x58]
0x1ef7d:    sub     rdx, qword ptr fs:[0x28]
0x1ef86:    jne     0x1ef8d
0x1ef88:    add     rsp, 0x68
0x1ef8c:    ret
0x1ef8d:    call    0x49d0
0x1ef92:    nop     word ptr cs:[rax + rax]

```

```

0x1ef9d:      nop      dword ptr [rax]
0x1efa0:      endbr64
0x1efa4:      sub      rsp, 0x38
0x1efa8:      movdqa   xmm0, xmmword ptr [rip + 0xccb0]
0x1efb0:      movdqa   xmm1, xmmword ptr [rip + 0xccb8]
0x1efb8:      mov      ecx, esi
0x1efba:      mov      rax, qword ptr fs:[0x28]
0x1efc3:      mov      qword ptr [rsp + 0x28], rax
0x1efc8:      xor      eax, eax
0x1efca:      mov      eax, dword ptr [rip + 0xccb0]
0x1efd0:      and      ecx, 0x1f
0x1efd3:      mov      r9, rsp
0x1efd6:      movaps   xmmword ptr [rsp], xmm0
0x1efda:      mov      dword ptr [rsp + 0x20], eax
0x1efde:      mov      eax, esi
0x1efe0:      shr      al, 5
0x1efe3:      movaps   xmmword ptr [rsp + 0x10], xmm1
0x1efe8:      movzx    eax, al
0x1efeb:      lea      rdx, [rsp + rax*4 + 4]
0x1eff0:      mov      esi, dword ptr [rdx]
0x1eff2:      mov      eax, esi
0x1eff4:      shr      eax, cl
0x1eff6:      not      eax
0x1eff8:      and      eax, 1
0x1effb:      shl      eax, cl
0x1effd:      mov      rcx, r9
0x1f000:      xor      eax, esi
0x1f002:      mov      rsi, rdi
0x1f005:      xor      edi, edi
0x1f007:      mov      dword ptr [rdx], eax
0x1f009:      mov      rdx, -1
0x1f010:      call     0x1e9e0
0x1f015:      mov      rdx, qword ptr [rsp + 0x28]
0x1f01a:      sub      rdx, qword ptr fs:[0x28]
0x1f023:      jne      0x1f02a
0x1f025:      add      rsp, 0x38
0x1f029:      ret
0x1f02a:      call     0x49d0
0x1f02f:      nop
0x1f030:      endbr64
0x1f034:      mov      esi, 0x3a
0x1f039:      jmp      0x1efa0

```

[10]: *# loading again as angr project for comparing the output dissembled code*

```
import os
```

```

import angr

base_addr = 0x100000
p = angr.Project(filePath, auto_load_libs=False, load_options = {'main_opts':
    ↳ {'base_addr': base_addr}})
cfg = p.analyses.CFGFast()
cfg.normalize()

```

```

ERROR      | 2023-02-26 17:40:05,090 |
angr.analyses.propagator.engine_vex.SimEnginePropagatorVEX | Unsupported Binop
Iop_MullS64.
ERROR      | 2023-02-26 17:40:05,116 |
angr.analyses.propagator.engine_vex.SimEnginePropagatorVEX | Unsupported Binop
Iop_MullS64.
ERROR      | 2023-02-26 17:40:06,783 |
angr.analyses.propagator.engine_vex.SimEnginePropagatorVEX | Unsupported Binop
Iop_F64toI32S.
ERROR      | 2023-02-26 17:40:06,788 |
angr.analyses.propagator.engine_vex.SimEnginePropagatorVEX | Unsupported Binop
Iop_F64toI32S.
ERROR      | 2023-02-26 17:40:06,981 |
angr.analyses.propagator.engine_vex.SimEnginePropagatorVEX | Unsupported Binop
Iop_MullS64.
ERROR      | 2023-02-26 17:40:07,048 |
angr.analyses.propagator.engine_vex.SimEnginePropagatorVEX | Unsupported Binop
Iop_MullS64.
ERROR      | 2023-02-26 17:40:07,653 |
angr.analyses.propagator.engine_vex.SimEnginePropagatorVEX | Unsupported Binop
Iop_64HLto128.

```

[11]: *# using angrs basic graph traversal method*
#<https://stackoverflow.com/questions/74457578/clean-angr-disassemble-output>

```

with open('angr.s', 'w') as angr_file:

    for func_node in cfg.functions.values():
        for block in func_node.blocks:
            c = block.capstone

            for i in c.insns:
                #
                print(i)

```

```

print("0x%x:\t%s\t%s" %(i.address,
                        i.mnemonic, i.op_str))
anгр_file.write("0x%x:\t%s\t%s" %(i.address,
                                i.mnemonic, i.op_str)+'\n')

```

```

0x104000:    endbr64
0x104004:    sub     rsp, 8
0x104008:    mov     rax, qword ptr [rip + 0x24fd9]
0x10400f:    test    rax, rax
0x104012:    je      0x104016
0x104016:    add     rsp, 8
0x10401a:    ret
0x104014:    call    rax
0x104020:    push    qword ptr [rip + 0x24be2]
0x104026:    bnd jmp qword ptr [rip + 0x24be3]
0x10402d:    nop     dword ptr [rax]
0x104030:    endbr64
0x104034:    push    0
0x104039:    bnd jmp 0x104020
0x10403f:    nop
0x104040:    endbr64
0x104044:    push    1
0x104049:    bnd jmp 0x104020
0x10404f:    nop
0x104050:    endbr64
0x104054:    push    2
0x104059:    bnd jmp 0x104020
0x10405f:    nop
0x104060:    endbr64
0x104064:    push    3
0x104069:    bnd jmp 0x104020
0x10406f:    nop
0x104070:    endbr64
0x104074:    push    4
0x104079:    bnd jmp 0x104020
0x10407f:    nop
0x104080:    endbr64
0x104084:    push    5
0x104089:    bnd jmp 0x104020
0x10408f:    nop
0x104090:    endbr64
0x104094:    push    6
0x104099:    bnd jmp 0x104020
0x10409f:    nop
0x1040a0:    endbr64
0x1040a4:    push    7
0x1040a9:    bnd jmp 0x104020

```

```

0x1040af:      nop
0x1040b0:      endbr64
0x1040b4:      push     8
0x1040b9:      bnd jmp 0x104020
0x1040bf:      nop
0x1040c0:      endbr64
0x1040c4:      push     9
0x1040c9:      bnd jmp 0x104020
0x1040cf:      nop
0x1040d0:      endbr64
0x1040d4:      push     0xa
0x1040d9:      bnd jmp 0x104020
0x1040df:      nop
0x1040e0:      endbr64
0x1040e4:      push     0xb
0x1040e9:      bnd jmp 0x104020
0x1040ef:      nop
0x1040f0:      endbr64
0x1040f4:      push     0xc
0x1040f9:      bnd jmp 0x104020
0x1040ff:      nop
0x104100:      endbr64
0x104104:      push     0xd
0x104109:      bnd jmp 0x104020
0x10410f:      nop
0x104110:      endbr64
0x104114:      push     0xe
0x104119:      bnd jmp 0x104020
0x10411f:      nop
0x104120:      endbr64
0x104124:      push     0xf
0x104129:      bnd jmp 0x104020
0x10412f:      nop
0x104130:      endbr64
0x104134:      push     0x10
0x104139:      bnd jmp 0x104020
0x10413f:      nop
0x104140:      endbr64
0x104144:      push     0x11
0x104149:      bnd jmp 0x104020
0x10414f:      nop
0x104150:      endbr64
0x104154:      push     0x12
0x104159:      bnd jmp 0x104020
0x10415f:      nop
0x104160:      endbr64
0x104164:      push     0x13
0x104169:      bnd jmp 0x104020

```

```

0x10416f:      nop
0x104170:      endbr64
0x104174:      push     0x14
0x104179:      bnd jmp  0x104020
0x10417f:      nop
0x104180:      endbr64
0x104184:      push     0x15
0x104189:      bnd jmp  0x104020
0x10418f:      nop
0x104190:      endbr64
0x104194:      push     0x16
0x104199:      bnd jmp  0x104020
0x10419f:      nop
0x1041a0:      endbr64
0x1041a4:      push     0x17
0x1041a9:      bnd jmp  0x104020
0x1041af:      nop
0x1041b0:      endbr64
0x1041b4:      push     0x18
0x1041b9:      bnd jmp  0x104020
0x1041bf:      nop
0x1041c0:      endbr64
0x1041c4:      push     0x19
0x1041c9:      bnd jmp  0x104020
0x1041cf:      nop
0x1041d0:      endbr64
0x1041d4:      push     0x1a
0x1041d9:      bnd jmp  0x104020
0x1041df:      nop
0x1041e0:      endbr64
0x1041e4:      push     0x1b
0x1041e9:      bnd jmp  0x104020
0x1041ef:      nop
0x1041f0:      endbr64
0x1041f4:      push     0x1c
0x1041f9:      bnd jmp  0x104020
0x1041ff:      nop
0x104200:      endbr64
0x104204:      push     0x1d
0x104209:      bnd jmp  0x104020
0x10420f:      nop
0x104210:      endbr64
0x104214:      push     0x1e
0x104219:      bnd jmp  0x104020
0x10421f:      nop
0x104220:      endbr64
0x104224:      push     0x1f
0x104229:      bnd jmp  0x104020

```



```

0x10422f:    nop
0x104230:    endbr64
0x104234:    push     0x20
0x104239:    bnd jmp  0x104020
0x10423f:    nop
0x104240:    endbr64
0x104244:    push     0x21
0x104249:    bnd jmp  0x104020
0x10424f:    nop
0x104250:    endbr64
0x104254:    push     0x22
0x104259:    bnd jmp  0x104020
0x10425f:    nop
0x104260:    endbr64
0x104264:    push     0x23
0x104269:    bnd jmp  0x104020
0x10426f:    nop
0x104270:    endbr64
0x104274:    push     0x24
0x104279:    bnd jmp  0x104020
0x10427f:    nop
0x104280:    endbr64
0x104284:    push     0x25
0x104289:    bnd jmp  0x104020
0x10428f:    nop
0x104290:    endbr64
0x104294:    push     0x26
0x104299:    bnd jmp  0x104020
0x10429f:    nop
0x1042a0:    endbr64
0x1042a4:    push     0x27
0x1042a9:    bnd jmp  0x104020
0x1042af:    nop
0x1042b0:    endbr64
0x1042b4:    push     0x28
0x1042b9:    bnd jmp  0x104020
0x1042bf:    nop
0x1042c0:    endbr64
0x1042c4:    push     0x29
0x1042c9:    bnd jmp  0x104020
0x1042cf:    nop
0x1042d0:    endbr64
0x1042d4:    push     0x2a
0x1042d9:    bnd jmp  0x104020
0x1042df:    nop
0x1042e0:    endbr64
0x1042e4:    push     0x2b
0x1042e9:    bnd jmp  0x104020

```

```

0x1042ef:      nop
0x1042f0:      endbr64
0x1042f4:      push     0x2c
0x1042f9:      bnd jmp 0x104020
0x1042ff:      nop
0x104300:      endbr64
0x104304:      push     0x2d
0x104309:      bnd jmp 0x104020
0x10430f:      nop
0x104310:      endbr64
0x104314:      push     0x2e
0x104319:      bnd jmp 0x104020
0x10431f:      nop
0x104320:      endbr64
0x104324:      push     0x2f
0x104329:      bnd jmp 0x104020
0x10432f:      nop
0x104330:      endbr64
0x104334:      push     0x30
0x104339:      bnd jmp 0x104020
0x10433f:      nop
0x104340:      endbr64
0x104344:      push     0x31
0x104349:      bnd jmp 0x104020
0x10434f:      nop
0x104350:      endbr64
0x104354:      push     0x32
0x104359:      bnd jmp 0x104020
0x10435f:      nop
0x104360:      endbr64
0x104364:      push     0x33
0x104369:      bnd jmp 0x104020
0x10436f:      nop
0x104370:      endbr64
0x104374:      push     0x34
0x104379:      bnd jmp 0x104020
0x10437f:      nop
0x104380:      endbr64
0x104384:      push     0x35
0x104389:      bnd jmp 0x104020
0x10438f:      nop
0x104390:      endbr64
0x104394:      push     0x36
0x104399:      bnd jmp 0x104020
0x10439f:      nop
0x1043a0:      endbr64
0x1043a4:      push     0x37
0x1043a9:      bnd jmp 0x104020

```

```

0x1043af:      nop
0x1043b0:      endbr64
0x1043b4:      push     0x38
0x1043b9:      bnd jmp 0x104020
0x1043bf:      nop
0x1043c0:      endbr64
0x1043c4:      push     0x39
0x1043c9:      bnd jmp 0x104020
0x1043cf:      nop
0x1043d0:      endbr64
0x1043d4:      push     0x3a
0x1043d9:      bnd jmp 0x104020
0x1043df:      nop
0x1043e0:      endbr64
0x1043e4:      push     0x3b
0x1043e9:      bnd jmp 0x104020
0x1043ef:      nop
0x1043f0:      endbr64
0x1043f4:      push     0x3c
0x1043f9:      bnd jmp 0x104020
0x1043ff:      nop
0x104400:      endbr64
0x104404:      push     0x3d
0x104409:      bnd jmp 0x104020
0x10440f:      nop
0x104410:      endbr64
0x104414:      push     0x3e
0x104419:      bnd jmp 0x104020
0x10441f:      nop
0x104420:      endbr64
0x104424:      push     0x3f
0x104429:      bnd jmp 0x104020
0x10442f:      nop
0x104430:      endbr64
0x104434:      push     0x40
0x104439:      bnd jmp 0x104020
0x10443f:      nop
0x104440:      endbr64
0x104444:      push     0x41
0x104449:      bnd jmp 0x104020
0x10444f:      nop
0x104450:      endbr64
0x104454:      push     0x42
0x104459:      bnd jmp 0x104020
0x10445f:      nop
0x104460:      endbr64
0x104464:      push     0x43
0x104469:      bnd jmp 0x104020

```

```

0x10446f:      nop
0x104470:      endbr64
0x104474:      push     0x44
0x104479:      bnd jmp  0x104020
0x10447f:      nop
0x104480:      endbr64
0x104484:      push     0x45
0x104489:      bnd jmp  0x104020
0x10448f:      nop
0x104490:      endbr64
0x104494:      push     0x46
0x104499:      bnd jmp  0x104020
0x10449f:      nop
0x1044a0:      endbr64
0x1044a4:      push     0x47
0x1044a9:      bnd jmp  0x104020
0x1044af:      nop
0x1044b0:      endbr64
0x1044b4:      push     0x48
0x1044b9:      bnd jmp  0x104020
0x1044bf:      nop
0x1044c0:      endbr64
0x1044c4:      push     0x49
0x1044c9:      bnd jmp  0x104020
0x1044cf:      nop
0x1044d0:      endbr64
0x1044d4:      push     0x4a
0x1044d9:      bnd jmp  0x104020
0x1044df:      nop
0x1044e0:      endbr64
0x1044e4:      push     0x4b
0x1044e9:      bnd jmp  0x104020
0x1044ef:      nop
0x1044f0:      endbr64
0x1044f4:      push     0x4c
0x1044f9:      bnd jmp  0x104020
0x1044ff:      nop
0x104500:      endbr64
0x104504:      push     0x4d
0x104509:      bnd jmp  0x104020
0x10450f:      nop
0x104510:      endbr64
0x104514:      push     0x4e
0x104519:      bnd jmp  0x104020
0x10451f:      nop
0x104520:      endbr64
0x104524:      push     0x4f
0x104529:      bnd jmp  0x104020

```

```

0x10452f:      nop
0x104530:      endbr64
0x104534:      push     0x50
0x104539:      bnd jmp  0x104020
0x10453f:      nop
0x104540:      endbr64
0x104544:      push     0x51
0x104549:      bnd jmp  0x104020
0x10454f:      nop
0x104550:      endbr64
0x104554:      push     0x52
0x104559:      bnd jmp  0x104020
0x10455f:      nop
0x104560:      endbr64
0x104564:      push     0x53
0x104569:      bnd jmp  0x104020
0x10456f:      nop
0x104570:      endbr64
0x104574:      push     0x54
0x104579:      bnd jmp  0x104020
0x10457f:      nop
0x104580:      endbr64
0x104584:      push     0x55
0x104589:      bnd jmp  0x104020
0x10458f:      nop
0x104590:      endbr64
0x104594:      push     0x56
0x104599:      bnd jmp  0x104020
0x10459f:      nop
0x1045a0:      endbr64
0x1045a4:      push     0x57
0x1045a9:      bnd jmp  0x104020
0x1045af:      nop
0x1045b0:      endbr64
0x1045b4:      push     0x58
0x1045b9:      bnd jmp  0x104020
0x1045bf:      nop
0x1045c0:      endbr64
0x1045c4:      push     0x59
0x1045c9:      bnd jmp  0x104020
0x1045cf:      nop
0x1045d0:      endbr64
0x1045d4:      push     0x5a
0x1045d9:      bnd jmp  0x104020
0x1045df:      nop
0x1045e0:      endbr64
0x1045e4:      push     0x5b
0x1045e9:      bnd jmp  0x104020

```

```

0x1045ef:    nop
0x1045f0:    endbr64
0x1045f4:    push     0x5c
0x1045f9:    bnd jmp  0x104020
0x1045ff:    nop
0x104600:    endbr64
0x104604:    push     0x5d
0x104609:    bnd jmp  0x104020
0x10460f:    nop
0x104610:    endbr64
0x104614:    push     0x5e
0x104619:    bnd jmp  0x104020
0x10461f:    nop
0x104620:    endbr64
0x104624:    push     0x5f
0x104629:    bnd jmp  0x104020
0x10462f:    nop
0x104630:    endbr64
0x104634:    push     0x60
0x104639:    bnd jmp  0x104020
0x10463f:    nop
0x104640:    endbr64
0x104644:    push     0x61
0x104649:    bnd jmp  0x104020
0x10464f:    nop
0x104650:    endbr64
0x104654:    push     0x62
0x104659:    bnd jmp  0x104020
0x10465f:    nop
0x104660:    endbr64
0x104664:    push     0x63
0x104669:    bnd jmp  0x104020
0x10466f:    nop
0x104670:    endbr64
0x104674:    push     0x64
0x104679:    bnd jmp  0x104020
0x10467f:    nop
0x104680:    endbr64
0x104684:    push     0x65
0x104689:    bnd jmp  0x104020
0x10468f:    nop
0x104690:    endbr64
0x104694:    push     0x66
0x104699:    bnd jmp  0x104020
0x10469f:    nop
0x1046a0:    endbr64
0x1046a4:    push     0x67
0x1046a9:    bnd jmp  0x104020

```

```

0x1046af:      nop
0x1046b0:      endbr64
0x1046b4:      push     0x68
0x1046b9:      bnd jmp 0x104020
0x1046bf:      nop
0x1046c0:      endbr64
0x1046c4:      push     0x69
0x1046c9:      bnd jmp 0x104020
0x1046cf:      nop
0x1046d0:      endbr64
0x1046d4:      push     0x6a
0x1046d9:      bnd jmp 0x104020
0x1046df:      nop
0x1046e0:      endbr64
0x1046e4:      push     0x6b
0x1046e9:      bnd jmp 0x104020
0x1046ef:      nop
0x1046f0:      endbr64
0x1046f4:      push     0x6c
0x1046f9:      bnd jmp 0x104020
0x1046ff:      nop
0x104700:      endbr64
0x104704:      push     0x6d
0x104709:      bnd jmp 0x104020
0x10470f:      nop
0x104710:      endbr64
0x104714:      push     0x6e
0x104719:      bnd jmp 0x104020
0x10471f:      nop
0x104720:      endbr64
0x104724:      push     0x6f
0x104729:      bnd jmp 0x104020
0x10472f:      nop
0x104730:      endbr64
0x104734:      push     0x70
0x104739:      bnd jmp 0x104020
0x10473f:      nop
0x104740:      endbr64
0x104744:      push     0x71
0x104749:      bnd jmp 0x104020
0x10474f:      nop
0x104750:      endbr64
0x104754:      push     0x72
0x104759:      bnd jmp 0x104020
0x10475f:      nop
0x104760:      endbr64
0x104764:      push     0x73
0x104769:      bnd jmp 0x104020

```

```

0x10476f:      nop
0x104770:      endbr64
0x104774:      push     0x74
0x104779:      bnd jmp  0x104020
0x10477f:      nop
0x104780:      endbr64
0x104784:      push     0x75
0x104789:      bnd jmp  0x104020
0x10478f:      nop
0x104790:      endbr64
0x104794:      bnd jmp  qword ptr [rip + 0x2483d]
0x1047a0:      endbr64
0x1047a4:      bnd jmp  qword ptr [rip + 0x24835]
0x1047b0:      endbr64
0x1047b4:      bnd jmp  qword ptr [rip + 0x2483d]
0x1047c0:      endbr64
0x1047c4:      bnd jmp  qword ptr [rip + 0x2444d]
0x1047d0:      endbr64
0x1047d4:      bnd jmp  qword ptr [rip + 0x24445]
0x1047e0:      endbr64
0x1047e4:      bnd jmp  qword ptr [rip + 0x2443d]
0x1047f0:      endbr64
0x1047f4:      bnd jmp  qword ptr [rip + 0x24435]
0x104800:      endbr64
0x104804:      bnd jmp  qword ptr [rip + 0x2442d]
0x104810:      endbr64
0x104814:      bnd jmp  qword ptr [rip + 0x24425]
0x104820:      endbr64
0x104824:      bnd jmp  qword ptr [rip + 0x2441d]
0x104830:      endbr64
0x104834:      bnd jmp  qword ptr [rip + 0x24415]
0x104840:      endbr64
0x104844:      bnd jmp  qword ptr [rip + 0x2440d]
0x104850:      endbr64
0x104854:      bnd jmp  qword ptr [rip + 0x24405]
0x104860:      endbr64
0x104864:      bnd jmp  qword ptr [rip + 0x243fd]
0x104870:      endbr64
0x104874:      bnd jmp  qword ptr [rip + 0x243f5]
0x104880:      endbr64
0x104884:      bnd jmp  qword ptr [rip + 0x243ed]
0x104890:      endbr64
0x104894:      bnd jmp  qword ptr [rip + 0x243e5]
0x1048a0:      endbr64
0x1048a4:      bnd jmp  qword ptr [rip + 0x243dd]
0x1048b0:      endbr64
0x1048b4:      bnd jmp  qword ptr [rip + 0x243d5]
0x1048c0:      endbr64

```



```

0x1048c4:      bnd jmp qword ptr [rip + 0x243cd]
0x1048d0:      endbr64
0x1048d4:      bnd jmp qword ptr [rip + 0x243c5]
0x1048e0:      endbr64
0x1048e4:      bnd jmp qword ptr [rip + 0x243bd]
0x1048f0:      endbr64
0x1048f4:      bnd jmp qword ptr [rip + 0x243b5]
0x104900:      endbr64
0x104904:      bnd jmp qword ptr [rip + 0x243ad]
0x104910:      endbr64
0x104914:      bnd jmp qword ptr [rip + 0x243a5]
0x104920:      endbr64
0x104924:      bnd jmp qword ptr [rip + 0x2439d]
0x104930:      endbr64
0x104934:      bnd jmp qword ptr [rip + 0x24395]
0x104940:      endbr64
0x104944:      bnd jmp qword ptr [rip + 0x2438d]
0x104950:      endbr64
0x104954:      bnd jmp qword ptr [rip + 0x24385]
0x104960:      endbr64
0x104964:      bnd jmp qword ptr [rip + 0x2437d]
0x104970:      endbr64
0x104974:      bnd jmp qword ptr [rip + 0x24375]
0x104980:      endbr64
0x104984:      bnd jmp qword ptr [rip + 0x2436d]
0x104990:      endbr64
0x104994:      bnd jmp qword ptr [rip + 0x24365]
0x1049a0:      endbr64
0x1049a4:      bnd jmp qword ptr [rip + 0x2435d]
0x1049b0:      endbr64
0x1049b4:      bnd jmp qword ptr [rip + 0x24355]
0x1049c0:      endbr64
0x1049c4:      bnd jmp qword ptr [rip + 0x2434d]
0x1049d0:      endbr64
0x1049d4:      bnd jmp qword ptr [rip + 0x24345]
0x1049e0:      endbr64
0x1049e4:      bnd jmp qword ptr [rip + 0x2433d]
0x1049f0:      endbr64
0x1049f4:      bnd jmp qword ptr [rip + 0x24335]
0x104a00:      endbr64
0x104a04:      bnd jmp qword ptr [rip + 0x2432d]
0x104a10:      endbr64
0x104a14:      bnd jmp qword ptr [rip + 0x24325]
0x104a20:      endbr64
0x104a24:      bnd jmp qword ptr [rip + 0x2431d]
0x104a30:      endbr64
0x104a34:      bnd jmp qword ptr [rip + 0x24315]
0x104a40:      endbr64

```

```

0x104a44:      bnd jmp qword ptr [rip + 0x2430d]
0x104a50:      endbr64
0x104a54:      bnd jmp qword ptr [rip + 0x24305]
0x104a60:      endbr64
0x104a64:      bnd jmp qword ptr [rip + 0x242fd]
0x104a70:      endbr64
0x104a74:      bnd jmp qword ptr [rip + 0x242f5]
0x104a80:      endbr64
0x104a84:      bnd jmp qword ptr [rip + 0x242ed]
0x104a90:      endbr64
0x104a94:      bnd jmp qword ptr [rip + 0x242e5]
0x104aa0:      endbr64
0x104aa4:      bnd jmp qword ptr [rip + 0x242dd]
0x104ab0:      endbr64
0x104ab4:      bnd jmp qword ptr [rip + 0x242d5]
0x104ac0:      endbr64
0x104ac4:      bnd jmp qword ptr [rip + 0x242cd]
0x104ad0:      endbr64
0x104ad4:      bnd jmp qword ptr [rip + 0x242c5]
0x104ae0:      endbr64
0x104ae4:      bnd jmp qword ptr [rip + 0x242bd]
0x104af0:      endbr64
0x104af4:      bnd jmp qword ptr [rip + 0x242b5]
0x104b00:      endbr64
0x104b04:      bnd jmp qword ptr [rip + 0x242ad]
0x104b10:      endbr64
0x104b14:      bnd jmp qword ptr [rip + 0x242a5]
0x104b20:      endbr64
0x104b24:      bnd jmp qword ptr [rip + 0x2429d]
0x104b30:      endbr64
0x104b34:      bnd jmp qword ptr [rip + 0x24295]
0x104b40:      endbr64
0x104b44:      bnd jmp qword ptr [rip + 0x2428d]
0x104b50:      endbr64
0x104b54:      bnd jmp qword ptr [rip + 0x24285]
0x104b60:      endbr64
0x104b64:      bnd jmp qword ptr [rip + 0x2427d]
0x104b70:      endbr64
0x104b74:      bnd jmp qword ptr [rip + 0x24275]
0x104b80:      endbr64
0x104b84:      bnd jmp qword ptr [rip + 0x2426d]
0x104b90:      endbr64
0x104b94:      bnd jmp qword ptr [rip + 0x24265]
0x104ba0:      endbr64
0x104ba4:      bnd jmp qword ptr [rip + 0x2425d]
0x104bb0:      endbr64
0x104bb4:      bnd jmp qword ptr [rip + 0x24255]
0x104bc0:      endbr64

```

```

0x104bc4:    bnd jmp qword ptr [rip + 0x2424d]
0x104bd0:    endbr64
0x104bd4:    bnd jmp qword ptr [rip + 0x24245]
0x104be0:    endbr64
0x104be4:    bnd jmp qword ptr [rip + 0x2423d]
0x104bf0:    endbr64
0x104bf4:    bnd jmp qword ptr [rip + 0x24235]
0x104c00:    endbr64
0x104c04:    bnd jmp qword ptr [rip + 0x2422d]
0x104c10:    endbr64
0x104c14:    bnd jmp qword ptr [rip + 0x24225]
0x104c20:    endbr64
0x104c24:    bnd jmp qword ptr [rip + 0x2421d]
0x104c30:    endbr64
0x104c34:    bnd jmp qword ptr [rip + 0x24215]
0x104c40:    endbr64
0x104c44:    bnd jmp qword ptr [rip + 0x2420d]
0x104c50:    endbr64
0x104c54:    bnd jmp qword ptr [rip + 0x24205]
0x104c60:    endbr64
0x104c64:    bnd jmp qword ptr [rip + 0x241fd]
0x104c70:    endbr64
0x104c74:    bnd jmp qword ptr [rip + 0x241f5]
0x104c80:    endbr64
0x104c84:    bnd jmp qword ptr [rip + 0x241ed]
0x104c90:    endbr64
0x104c94:    bnd jmp qword ptr [rip + 0x241e5]
0x104ca0:    endbr64
0x104ca4:    bnd jmp qword ptr [rip + 0x241dd]
0x104cb0:    endbr64
0x104cb4:    bnd jmp qword ptr [rip + 0x241d5]
0x104cc0:    endbr64
0x104cc4:    bnd jmp qword ptr [rip + 0x241cd]
0x104cd0:    endbr64
0x104cd4:    bnd jmp qword ptr [rip + 0x241c5]
0x104ce0:    endbr64
0x104ce4:    bnd jmp qword ptr [rip + 0x241bd]
0x104cf0:    endbr64
0x104cf4:    bnd jmp qword ptr [rip + 0x241b5]
0x104d00:    endbr64
0x104d04:    bnd jmp qword ptr [rip + 0x241ad]
0x104d10:    endbr64
0x104d14:    bnd jmp qword ptr [rip + 0x241a5]
0x104d20:    endbr64
0x104d24:    bnd jmp qword ptr [rip + 0x2419d]
0x104d30:    endbr64
0x104d34:    bnd jmp qword ptr [rip + 0x24195]
0x104d40:    endbr64

```

```

0x104d44:      bnd jmp qword ptr [rip + 0x2418d]
0x104d50:      endbr64
0x104d54:      bnd jmp qword ptr [rip + 0x24185]
0x104d60:      endbr64
0x104d64:      bnd jmp qword ptr [rip + 0x2417d]
0x104d70:      endbr64
0x104d74:      bnd jmp qword ptr [rip + 0x24175]
0x104d80:      endbr64
0x104d84:      bnd jmp qword ptr [rip + 0x2416d]
0x104d90:      endbr64
0x104d94:      bnd jmp qword ptr [rip + 0x24165]
0x104da0:      endbr64
0x104da4:      bnd jmp qword ptr [rip + 0x2415d]
0x104db0:      endbr64
0x104db4:      bnd jmp qword ptr [rip + 0x24155]
0x104dc0:      endbr64
0x104dc4:      bnd jmp qword ptr [rip + 0x2414d]
0x104dd0:      endbr64
0x104dd4:      bnd jmp qword ptr [rip + 0x24145]
0x104de0:      endbr64
0x104de4:      bnd jmp qword ptr [rip + 0x2413d]
0x104df0:      endbr64
0x104df4:      bnd jmp qword ptr [rip + 0x24135]
0x104e00:      endbr64
0x104e04:      bnd jmp qword ptr [rip + 0x2412d]
0x104e10:      endbr64
0x104e14:      bnd jmp qword ptr [rip + 0x24125]
0x104e20:      endbr64
0x104e24:      bnd jmp qword ptr [rip + 0x2411d]
0x104e30:      endbr64
0x104e34:      bnd jmp qword ptr [rip + 0x24115]
0x104e40:      endbr64
0x104e44:      bnd jmp qword ptr [rip + 0x2410d]
0x104e50:      endbr64
0x104e54:      bnd jmp qword ptr [rip + 0x24105]
0x104e60:      endbr64
0x104e64:      bnd jmp qword ptr [rip + 0x240fd]
0x104e70:      endbr64
0x104e74:      bnd jmp qword ptr [rip + 0x240f5]
0x104e80:      endbr64
0x104e84:      bnd jmp qword ptr [rip + 0x240ed]
0x104e90:      endbr64
0x104e94:      bnd jmp qword ptr [rip + 0x240e5]
0x104ea0:      endbr64
0x104ea4:      bnd jmp qword ptr [rip + 0x240dd]
0x104eb0:      endbr64
0x104eb4:      bnd jmp qword ptr [rip + 0x240d5]
0x104ec0:      endbr64

```

```

0x104ec4:    bnd jmp qword ptr [rip + 0x240cd]
0x104ed0:    endbr64
0x104ed4:    bnd jmp qword ptr [rip + 0x240c5]
0x104ee0:    endbr64
0x104ee4:    bnd jmp qword ptr [rip + 0x240bd]
0x104ef0:    endbr64
0x104ef4:    bnd jmp qword ptr [rip + 0x240b5]
0x104f00:    endbr64
0x104f04:    bnd jmp qword ptr [rip + 0x240ad]
0x104f10:    endbr64
0x104f14:    bnd jmp qword ptr [rip + 0x240a5]
0x104f20:    mov     byte ptr [rsp + 8], 0
0x104f25:    jmp     0x1107fa
0x1107fa:    mov     r15, qword ptr [rax]
0x1107fd:    mov     rdi, r15
0x110800:    call    0x1049b0
0x110805:    movsxd  rdx, ebp
0x110808:    mov     rbp, rdx
0x11080b:    sub     rbp, r14
0x11080e:    cmp     rax, rbp
0x110811:    jbe     0x11082b
0x11082b:    cmp     rbp, rax
0x11082e:    mov     rsi, r15
0x110831:    cmova   rbp, rax
0x110835:    mov     rax, qword ptr [r13 + 0x20]
0x110839:    mov     rdx, rbp
0x11083c:    lea     rdi, [rax + 1]
0x110840:    call    0x104bb0
0x110813:    movsxd  rcx, dword ptr [r13 + 0x74]
0x110817:    lea     rsi, [rbp + rcx]
0x11081c:    cmp     rax, rsi
0x11081f:    jb      0x110a90
0x110845:    mov     rax, qword ptr [r13 + 0x20]
0x110849:    mov     rsi, rbp
0x11084c:    lea     rdi, [rax + 1]
0x110850:    call    0x1184d0
0x110a90:    add     r14, rax
0x110a93:    mov     rcx, rdx
0x110a96:    sub     rcx, r14
0x110a99:    sub     r14, rdx
0x110a9c:    add     rax, rcx
0x110a9f:    add     r15, r14
0x110aa2:    jmp     0x11082b
0x110825:    sub     rax, rcx
0x110828:    add     r15, rcx
0x11082b:    cmp     rbp, rax
0x11082e:    mov     rsi, r15
0x110831:    cmova   rbp, rax

```

```

0x110835:    mov     rax, qword ptr [r13 + 0x20]
0x110839:    mov     rdx, rbp
0x11083c:    lea     rdi, [rax + 1]
0x110840:    call    0x104bb0
0x110855:    movzx   ecx, byte ptr [rsp + 8]
0x11085a:    test    cl, cl
0x11085c:    je      0x110866
0x110866:    mov     eax, dword ptr [r13 + 0x64]
0x11086a:    cmp     eax, 0x27
0x11086d:    jle     0x1108a0
0x11085e:    mov     rax, qword ptr [r13 + 0x20]
0x110862:    mov     byte ptr [rax + rbp + 1], cl
0x110866:    mov     eax, dword ptr [r13 + 0x64]
0x11086a:    cmp     eax, 0x27
0x11086d:    jle     0x1108a0
0x1108a0:    mov     rax, qword ptr [r13 + 0x38]
0x1108a4:    cmp     byte ptr [rax + r12 + 0x49], 0
0x1108aa:    je      0x1108b9
0x11086f:    mov     edx, dword ptr [r13 + 0x80]
0x110876:    cmp     edx, 4
0x110879:    je      0x1108a0
0x1108b9:    cmp     ebx, dword ptr [r13 + 0x28]
0x1108bd:    je      0x110948
0x1108ac:    movsxd  rax, dword ptr [r13 + 0x64]
0x1108b0:    mov     rdx, qword ptr [r13 + 0x20]
0x1108b4:    mov     byte ptr [rdx + rax - 3], 0x2a
0x1108b9:    cmp     ebx, dword ptr [r13 + 0x28]
0x1108bd:    je      0x110948
0x11087b:    jg      0x1109e0
0x110948:    cmp     byte ptr [r13 + 0x6c], 1
0x11094d:    mov     rdx, qword ptr [r13 + 0x20]
0x110951:    sbb     eax, eax
0x110953:    and     eax, 0xffffffffec
0x110956:    add     eax, 0x3e
0x110959:    mov     byte ptr [rdx], al
0x11095b:    mov     rax, qword ptr [r13 + 0x38]
0x11095f:    mov     esi, dword ptr [rip + 0x192f7]
0x110965:    add     rax, r12
0x110968:    test    esi, esi
0x11096a:    je      0x110a50
0x1108c3:    mov     rax, qword ptr [r13 + 0x38]
0x1108c7:    mov     ecx, dword ptr [rip + 0x1938f]
0x1108cd:    movzx   edx, byte ptr [rax + r12 + 0x49]
0x1108d3:    test    ecx, ecx
0x1108d5:    je      0x1108e8
0x110881:    cmp     edx, 1
0x110884:    jg      0x1109c0
0x1109e0:    cmp     edx, 5

```

```

0x1109e3:      jne      0x110892
0x110a50:      movzx   edx, byte ptr [rax + 0x49]
0x110a54:      cmp     ebx, dword ptr [r13 + 0x28]
0x110a58:      jne     0x1108e8
0x110970:      cmp     byte ptr [rax + 0x48], 0
0x110974:      movzx   edx, byte ptr [rax + 0x49]
0x110978:      je      0x110a50
0x1108e8:      test    dl, dl
0x1108ea:      jne     0x1109a0
0x1108d7:      add     rax, r12
0x1108da:      cmp     byte ptr [rax + 0x48], 0
0x1108de:      movzx   edx, byte ptr [rax + 0x49]
0x1108e2:      jne     0x11097e
0x11088a:      test    edx, edx
0x11088c:      jns     0x110c30
0x1109c0:      mov     ecx, 0xb
0x1109c5:      mov     esi, ebx
0x1109c7:      mov     rdi, r13
0x1109ca:      xor     eax, eax
0x1109cc:      call    0x110420
0x1109e9:      cmp     eax, 0x49
0x1109ec:      jle     0x1108a0
0x110892:      lea     rdi, [rip + 0x10953]
0x110899:      call    0x108b00
0x110a5e:      cmp     byte ptr [r13 + 0x6c], 1
0x110a63:      jne     0x1108e8
0x11097e:      test    dl, dl
0x110980:      je      0x110b10
0x1108f0:      mov     esi, dword ptr [rip + 0x19a46]
0x1108f6:      mov     r9d, dword ptr [rip + 0x19a27]
0x1108fd:      mov     edi, dword ptr [rip + 0x19a3d]
0x110903:      mov     edx, r9d
0x110906:      call    0x10a570
0x1109a0:      mov     esi, dword ptr [rip + 0x1998e]
0x1109a6:      mov     r9d, dword ptr [rip + 0x19977]
0x1109ad:      mov     edi, dword ptr [rip + 0x19985]
0x1109b3:      jmp     0x110903
0x110c30:      mov     ecx, 0x10
0x110c35:      mov     esi, ebx
0x110c37:      mov     rdi, r13
0x110c3a:      xor     eax, eax
0x110c3c:      call    0x110420
0x1109d1:      jmp     0x1108a0
0x1109f2:      xor     edx, edx
0x1109f4:      mov     esi, ebx
0x1109f6:      mov     rdi, r13
0x1109f9:      xor     eax, eax
0x1109fb:      mov     ecx, 0x36

```

```

0x110a00:    call    0x110420
0x110a69:    mov     esi, dword ptr [rip + 0x198d9]
0x110a6f:    test    dl, dl
0x110a71:    jne     0x110c08
0x110b10:    mov     rdi, qword ptr [r13 + 8]
0x110b14:    mov     esi, ebx
0x110b16:    sub     esi, dword ptr [r13 + 0x2c]
0x110b1a:    mov     edx, 1
0x110b1f:    add     esi, 1
0x110b22:    call    0x10c680
0x110986:    cmp     ebx, dword ptr [r13 + 0x28]
0x11098a:    jne     0x1109a0
0x11090b:    sub     ebx, dword ptr [r13 + 0x2c]
0x11090f:    mov     rdi, qword ptr [r13 + 8]
0x110913:    mov     edx, 1
0x110918:    lea     esi, [rbx + 1]
0x11091b:    call    0x10c680
0x110903:    mov     edx, r9d
0x110906:    call    0x10a570
0x110c41:    jmp     0x1108a0
0x110a05:    mov     esi, ebx
0x110a07:    mov     rdi, r13
0x110a0a:    xor     eax, eax
0x110a0c:    mov     ecx, 0x26
0x110a11:    mov     edx, 1
0x110a16:    call    0x110420
0x110a77:    mov     r9d, dword ptr [rip + 0x198ce]
0x110a7e:    test    dl, dl
0x110a80:    je      0x1108fd
0x110c08:    mov     esi, dword ptr [rip + 0x19736]
0x110c0e:    jmp     0x110a77
0x110b27:    mov     rax, qword ptr [r13 + 0x38]
0x110b2b:    cmp     byte ptr [rax + r12 + 0x49], 0
0x110b31:    jne     0x110c50
0x11098c:    cmp     byte ptr [r13 + 0x6c], 1
0x110991:    je      0x110c08
0x110920:    mov     eax, dword ptr [r13 + 0x64]
0x110924:    mov     rsi, qword ptr [r13 + 0x20]
0x110928:    mov     rdi, qword ptr [r13 + 8]
0x11092c:    add     rsp, 0x28
0x110930:    pop     rbx
0x110931:    lea     edx, [rax - 2]
0x110934:    pop     rbp
0x110935:    pop     r12
0x110937:    pop     r13
0x110939:    pop     r14
0x11093b:    pop     r15
0x11093d:    jmp     0x10c600

```



```

0x110a1b:    mov     esi, ebx
0x110a1d:    mov     rdi, r13
0x110a20:    xor     eax, eax
0x110a22:    mov     ecx, 0x16
0x110a27:    mov     edx, 2
0x110a2c:    call    0x110420
0x1108fd:    mov     edi, dword ptr [rip + 0x19a3d]
0x110903:    mov     edx, r9d
0x110906:    call    0x10a570
0x110a86:    jmp     0x1109ad
0x110b37:    mov     eax, dword ptr [rip + 0x197ff]
0x110b3d:    mov     r15d, dword ptr [rip + 0x197fc]
0x110b44:    mov     dword ptr [rsp + 0xc], eax
0x110b48:    cmp     ebx, dword ptr [r13 + 0x28]
0x110b4c:    je      0x110c70
0x110c50:    mov     eax, dword ptr [rip + 0x196de]
0x110c56:    mov     r15d, dword ptr [rip + 0x196db]
0x110c5d:    mov     dword ptr [rsp + 0xc], eax
0x110c61:    jmp     0x110b48
0x110997:    nop     word ptr [rax + rax]
0x110a31:    mov     ecx, 0xb
0x110a36:    mov     esi, ebx
0x110a38:    mov     rdi, r13
0x110a3b:    mov     edx, 3
0x110a40:    xor     eax, eax
0x110a42:    call    0x110420
0x1109ad:    mov     edi, dword ptr [rip + 0x19985]
0x1109b3:    jmp     0x110903
0x110c70:    cmp     byte ptr [r13 + 0x6c], 1
0x110c75:    jne     0x110b52
0x110b52:    mov     eax, dword ptr [rip + 0x197cc]
0x110b58:    mov     dword ptr [rsp + 8], eax
0x110b5c:    mov     edx, dword ptr [rsp + 8]
0x110b60:    mov     esi, dword ptr [rsp + 0xc]
0x110b64:    mov     edi, r15d
0x110b67:    call    0x10a570
0x110b48:    cmp     ebx, dword ptr [r13 + 0x28]
0x110b4c:    je      0x110c70
0x110a47:    jmp     0x1108a0
0x110c7b:    mov     eax, dword ptr [rip + 0x196cb]
0x110c81:    mov     dword ptr [rsp + 8], eax
0x110c85:    jmp     0x110b5c
0x110b6c:    mov     rax, qword ptr [r13 + 0x20]
0x110b70:    mov     rdi, qword ptr [r13 + 8]
0x110b74:    movsx   esi, byte ptr [rax]
0x110b77:    call    0x10c650
0x110b5c:    mov     edx, dword ptr [rsp + 8]
0x110b60:    mov     esi, dword ptr [rsp + 0xc]

```

```

0x110b64:    mov     edi, r15d
0x110b67:    call   0x10a570
0x110b7c:    mov     rax, qword ptr [r13 + 0x38]
0x110b80:    lea     rdx, [rax + r12]
0x110b84:    cmp     byte ptr [rdx + 0x4b], 0
0x110b88:    je      0x110c90
0x110c90:    mov     r14, qword ptr [rip + 0x1adf9]
0x110c97:    mov     byte ptr [rdx + 0x4b], 1
0x110c9b:    test    r14, r14
0x110c9e:    je      0x110d49
0x110b8e:    movsx   edi, byte ptr [rdx + 0x4e]
0x110b92:    cmp     dil, 2
0x110b96:    je      0x110bb6
0x110d49:    mov     byte ptr [rdx + 0x4e], 2
0x110d4d:    jmp     0x110bb6
0x110ca4:    mov     qword ptr [rsp + 0x18], 0
0x110cad:    mov     qword ptr [rsp + 0x10], 0
0x110cb6:    nop     word ptr cs:[rax + rax]
0x110cc0:    mov     rsi, qword ptr [rax + r12]
0x110cc4:    mov     rdi, qword ptr [r14]
0x110cc7:    mov     edx, 0x15
0x110ccc:    call    0x104a40
0x110bb6:    mov     rax, qword ptr [r13 + 0x20]
0x110bba:    mov     rdi, qword ptr [r13 + 8]
0x110bbe:    lea     edx, [rbp + 1]
0x110bc1:    lea     rsi, [rax + 1]
0x110bc5:    call    0x10c600
0x110b98:    movsx   esi, byte ptr [rdx + 0x4c]
0x110b9c:    cmp     ebx, dword ptr [r13 + 0x28]
0x110ba0:    jne     0x110bad
0x110cd1:    test    eax, eax
0x110cd3:    jne     0x110cee
0x110bca:    mov     eax, dword ptr [r13 + 0x64]
0x110bce:    mov     rdi, qword ptr [r13 + 8]
0x110bd2:    lea     edx, [rax - 4]
0x110bd5:    mov     rax, qword ptr [r13 + 0x20]
0x110bd9:    sub     edx, ebp
0x110bdb:    lea     rsi, [rax + rbp + 2]
0x110be0:    call    0x10c600
0x110ba2:    cmp     byte ptr [r13 + 0x6c], 1
0x110ba7:    je      0x110d52
0x110bad:    movsx   edx, byte ptr [rdx + 0x4d]
0x110bb1:    call    0x10a570
0x110cd5:    mov     rdi, qword ptr [r14]
0x110cd8:    call    0x1049b0
0x110cee:    mov     r14, qword ptr [r14 + 0x18]
0x110cf2:    mov     rax, qword ptr [r13 + 0x38]
0x110cf6:    test    r14, r14

```

```

0x110cf9:    jne     0x110cc0
0x110be5:    mov     edx, dword ptr [rsp + 8]
0x110be9:    mov     esi, dword ptr [rsp + 0xc]
0x110bed:    mov     edi, r15d
0x110bf0:    add     rsp, 0x28
0x110bf4:    pop     rbx
0x110bf5:    pop     rbp
0x110bf6:    pop     r12
0x110bf8:    pop     r13
0x110bfa:    pop     r14
0x110bfc:    pop     r15
0x110bfe:    jmp     0x10a570
0x110d52:    mov     edx, dword ptr [rsp + 8]
0x110d56:    mov     edi, r15d
0x110d59:    call    0x10a570
0x110cdd:    cmp     qword ptr [rsp + 0x10], rax
0x110ce2:    jae     0x110cee
0x110cfb:    mov     rsi, qword ptr [rsp + 0x18]
0x110d00:    lea     rdx, [rax + r12]
0x110d04:    test    rsi, rsi
0x110d07:    je      0x110d49
0x110cc0:    mov     rsi, qword ptr [rax + r12]
0x110cc4:    mov     rdi, qword ptr [r14]
0x110cc7:    mov     edx, 0x15
0x110ccc:    call    0x104a40
0x110d5e:    jmp     0x110bb6
0x110ce4:    mov     qword ptr [rsp + 0x18], r14
0x110ce9:    mov     qword ptr [rsp + 0x10], rax
0x110cee:    mov     r14, qword ptr [r14 + 0x18]
0x110cf2:    mov     rax, qword ptr [r13 + 0x38]
0x110cf6:    test    r14, r14
0x110cf9:    jne     0x110cc0
0x110d09:    mov     eax, dword ptr [rsi + 0x10]
0x110d0c:    xor     ecx, ecx
0x110d0e:    cmp     eax, -1
0x110d11:    cmovne  eax, ecx
0x110d14:    mov     byte ptr [rdx + 0x4e], al
0x110d17:    movsx   edi, al
0x110d1a:    mov     eax, dword ptr [rsi + 8]
0x110d1d:    mov     ecx, eax
0x110d1f:    cmp     eax, -1
0x110d22:    movzx   eax, byte ptr [rip + 0x19613]
0x110d29:    cmovne  eax, ecx
0x110d2c:    mov     byte ptr [rdx + 0x4c], al
0x110d2f:    mov     eax, dword ptr [rsi + 0xc]
0x110d32:    mov     ecx, eax
0x110d34:    cmp     eax, -1
0x110d37:    movzx   eax, byte ptr [rip + 0x195e6]

```

```

0x110d3e:      cmovne  eax, ecx
0x110d41:      mov     byte ptr [rdx + 0x4d], al
0x110d44:      jmp     0x110b92
0x110b92:      cmp     dil, 2
0x110b96:      je      0x110bb6
0x104f2a:      call    0x104850
0x104f2f:      nop
0x104f30:      endbr64
0x104f34:      push    r15
0x104f36:      push    r14
0x104f38:      lea     r14, [rip + 0x1b669]
0x104f3f:      push    r13
0x104f41:      lea     r13, [rip + 0x1b2b4]
0x104f48:      push    r12
0x104f4a:      push    rbp
0x104f4b:      mov     ebp, edi
0x104f4d:      push    rbx
0x104f4e:      mov     rbx, rsi
0x104f51:      sub     rsp, 0x88
0x104f58:      mov     rax, qword ptr fs:[0x28]
0x104f61:      mov     qword ptr [rsp + 0x78], rax
0x104f66:      xor     eax, eax
0x104f68:      mov     dword ptr [rsp + 0x34], 0
0x104f70:      mov     qword ptr [rsp + 0x50], 0
0x104f79:      mov     qword ptr [rsp + 0x58], 0
0x104f82:      call    0x118c50
0x104f87:      lea     rsi, [rip + 0x1d457]
0x104f8e:      mov     edi, 6
0x104f93:      call    0x104d00
0x104f98:      mov     rax, qword ptr [rbx]
0x104f9b:      lea     rdi, [rip + 0x1b5be]
0x104fa2:      mov     qword ptr [rip + 0x2593f], rax
0x104fa9:      call    0x1047e0
0x104fae:      lea     rdx, [rip + 0x1c729]
0x104fb5:      test    rax, rax
0x104fb8:      cmov     rax, rdx
0x104fbc:      mov     qword ptr [rip + 0x2592d], rax
0x104fc3:      xor     eax, eax
0x104fc5:      call    0x117660
0x104fca:      xor     eax, eax
0x104fcc:      call    0x118420
0x104fd1:      xor     eax, eax
0x104fd3:      call    0x118160
0x104fd8:      mov     edi, 0x25
0x104fdd:      call    0x11d670
0x104fe2:      lea     r8, [rip + 0x1b57c]
0x104fe9:      mov     esi, 1
0x104fee:      lea     rcx, [rip + 0x1b586]

```

```

0x104ff5:    mov     rdi, rax
0x104ff8:    mov     qword ptr [rip + 0x258d1], rax
0x104fff:    xor     eax, eax
0x105001:    mov     rdx, -1
0x105008:    call    0x104f10
0x10500d:    lea     rdi, [rip + 0x1b571]
0x105014:    call    0x1047e0
0x105019:    mov     dword ptr [rsp], 1
0x105020:    cmp     rax, 1
0x105024:    sbb     r12d, r12d
0x105027:    xor     r15d, r15d
0x10502a:    or      r12d, 1
0x10502e:    mov     rdx, r14
0x105031:    mov     rsi, rbx
0x105034:    mov     edi, ebp
0x105036:    call    0x104de0
0x10503b:    cmp     eax, -1
0x10503e:    je      0x10510e
0x10510e:    movsxd  rax, dword ptr [rip + 0x25733]
0x105115:    lea     r13, [rip + 0x1c5c2]
0x10511c:    cmp     eax, ebp
0x10511e:    jl      0x106f96
0x105044:    sub     eax, 0x3f
0x105047:    cmp     eax, 0x37
0x10504a:    ja      0x105057
0x106f96:    mov     rdi, qword ptr [rbx + rax*8]
0x106f9a:    lea     edx, [rax + 1]
0x106f9d:    mov     dword ptr [rip + 0x238a5], edx
0x106fa3:    call    0x11d7a0
0x105124:    mov     r14, r13
0x105127:    mov     edi, 0xa
0x10512c:    lea     rbp, [rip + 0x1b51c]
0x105133:    call    0x104810
0x10504c:    movsxd  rax, dword ptr [r13 + rax*4]
0x105051:    add     rax, r13
0x105054:    jmp     rax
0x105057:    mov     rcx, qword ptr [rip + 0x2588a]
0x10505e:    mov     rdi, qword ptr [rip + 0x2581b]
0x105065:    lea     rdx, [rip + 0x1b529]
0x10506c:    xor     eax, eax
0x10506e:    mov     esi, 1
0x105073:    mov     r15d, 1
0x105079:    call    0x104e60
0x106fa8:    mov     r13, rax
0x106fab:    movsxd  rax, dword ptr [rip + 0x23896]
0x106fb2:    cmp     eax, ebp
0x106fb4:    jge     0x105124
0x105138:    lea     rax, [rip + 0x1b4a5]

```

```

0x10513f:    lea     r9, [rip + 0x1b469]
0x105146:    mov     edi, 1
0x10514b:    push    rsi
0x10514c:    lea     r8, [rip + 0x1b465]
0x105153:    lea     rcx, [rip + 0x1b435]
0x10515a:    push    rax
0x10515b:    lea     rdx, [rip + 0x1b403]
0x105162:    lea     rsi, [rip + 0x1b468]
0x105169:    xor     eax, eax
0x10516b:    call    0x104d20
0x1050e4:    mov     r12d, 1
0x1050ea:    jmp     0x10502e
0x105106:    xor     r12d, r12d
0x105109:    jmp     0x10502e
0x1050a7:    lea     rcx, [rip + 0x1b4e1]
0x1050ae:    mov     edi, 1
0x1050b3:    xor     eax, eax
0x1050b5:    xor     r15d, r15d
0x1050b8:    lea     rdx, [rip + 0x1b4a6]
0x1050bf:    lea     rsi, [rip + 0x1b517]
0x1050c6:    call    0x104d20
0x1050cd:    mov     r15d, 1
0x1050d3:    jmp     0x10502e
0x1050ef:    mov     dword ptr [rsp], 0
0x1050f6:    jmp     0x10502e
0x1050d8:    xor     eax, eax
0x1050da:    xor     r15d, r15d
0x1050dd:    call    0x109720
0x1050fb:    mov     r15d, 1
0x105101:    jmp     0x10507e
0x10507e:    mov     rax, qword ptr [rsp + 0x78]
0x105083:    sub     rax, qword ptr fs:[0x28]
0x10508c:    jne     0x107475
0x106fba:    mov     rdi, qword ptr [rbx + rax*8]
0x106fbe:    lea     edx, [rax + 1]
0x106fc1:    mov     dword ptr [rip + 0x23881], edx
0x106fc7:    call    0x11d7a0
0x105170:    mov     rsi, qword ptr [rip + 0x24b09]
0x105177:    mov     edi, 1
0x10517c:    xor     eax, eax
0x10517e:    call    0x104d20
0x10502e:    mov     rdx, r14
0x105031:    mov     rsi, rbx
0x105034:    mov     edi, ebp
0x105036:    call    0x104de0
0x1050cb:    jmp     0x10507e
0x1050e2:    jmp     0x10507e
0x105092:    add     rsp, 0x88

```

```

0x105099:      mov     eax, r15d
0x10509c:      pop     rbx
0x10509d:      pop     rbp
0x10509e:      pop     r12
0x1050a0:      pop     r13
0x1050a2:      pop     r14
0x1050a4:      pop     r15
0x1050a6:      ret
0x107475:      call    0x1049d0
0x106fcc:      cmp     dword ptr [rip + 0x23876], ebp
0x106fd2:      mov     r14, rax
0x106fd5:      jge     0x105127
0x105183:      xor     eax, eax
0x105185:      lea     rdx, [rip + 0x1b464]
0x10518c:      lea     rsi, [rip + 0x1b464]
0x105193:      lea     rdi, [rip + 0x1b459]
0x10519a:      call    0x1092b0
0x106fdb:      mov     rcx, qword ptr [rip + 0x23906]
0x106fe2:      mov     rdi, qword ptr [rip + 0x23897]
0x106fe9:      lea     rdx, [rip + 0x198f8]
0x106ff0:      xor     eax, eax
0x106ff2:      mov     esi, 1
0x106ff7:      call    0x104e60
0x105127:      mov     edi, 0xa
0x10512c:      lea     rbp, [rip + 0x1b51c]
0x105133:      call    0x104810
0x10519f:      xor     eax, eax
0x1051a1:      lea     rdx, [rip + 0x1b456]
0x1051a8:      lea     rsi, [rip + 0x1b458]
0x1051af:      lea     rdi, [rip + 0x1b44d]
0x1051b6:      call    0x1092b0
0x106ffc:      jmp     0x105127
0x1051bb:      xor     esi, esi
0x1051bd:      xor     eax, eax
0x1051bf:      lea     rdx, [rip + 0x1b447]
0x1051c6:      lea     rdi, [rip + 0x1b445]
0x1051cd:      call    0x1092b0
0x1051d2:      xor     eax, eax
0x1051d4:      lea     rdx, [rip + 0x1b443]
0x1051db:      lea     rsi, [rip + 0x1b448]
0x1051e2:      lea     rdi, [rip + 0x1b43d]
0x1051e9:      call    0x1092b0
0x1051ee:      xor     esi, esi
0x1051f0:      xor     eax, eax
0x1051f2:      lea     rdx, [rip + 0x1b437]
0x1051f9:      lea     rdi, [rip + 0x1b435]
0x105200:      call    0x1092b0
0x105205:      lea     rdx, [rip + 0x1b433]

```

```

0x10520c:    xor     esi, esi
0x10520e:    xor     eax, eax
0x105210:    lea     rdi, [rip + 0x1b42d]
0x105217:    call    0x1092b0
0x10521c:    xor     edi, edi
0x10521e:    call    0x10c3d0
0x105223:    xor     eax, eax
0x105225:    call    0x1179a0
0x10522a:    mov     rdi, rbp
0x10522d:    call    0x117c50
0x105232:    xor     eax, eax
0x105234:    call    0x118660
0x105239:    lea     rdi, [rip + 0x1b41b]
0x105240:    call    0x117c50
0x105245:    lea     rax, [rsp + 0x44]
0x10524a:    xor     edi, edi
0x10524c:    mov     rsi, rax
0x10524f:    mov     qword ptr [rsp + 0x20], rax
0x105254:    xor     eax, eax
0x105256:    call    0x109310
0x10525b:    mov     dword ptr [rsp + 0x1c], eax
0x10525f:    mov     eax, dword ptr [rsp + 0x44]
0x105263:    add     dword ptr [rip + 0x25673], eax
0x105269:    xor     eax, eax
0x10526b:    call    0x117280
0x105270:    xor     eax, eax
0x105272:    call    0x117a30
0x105277:    pop     rdi
0x105278:    lea     rdi, [rip + 0x1b3e9]
0x10527f:    pop     r8
0x105281:    cmp     eax, 1
0x105284:    adc     dword ptr [rip + 0x25651], 0
0x10528b:    call    0x1047e0
0x105290:    mov     rbx, rax
0x105293:    test    rax, rax
0x105296:    je      0x107011
0x107011:    lea     rbx, [rip + 0x19543]
0x107018:    jmp     0x10529c
0x10529c:    mov     rdi, rbx
0x10529f:    call    0x1049b0
0x1052a4:    lea     rdi, [rax + 0x21]
0x1052a8:    call    0x11d670
0x1052ad:    mov     rdi, rbx
0x1052b0:    mov     qword ptr [rip + 0x26819], rax
0x1052b7:    call    0x1049b0
0x1052bc:    lea     rdi, [rax + 0x21]
0x1052c0:    call    0x11d670
0x1052c5:    xor     edi, edi

```



```

0x1052c7:    mov     qword ptr [rip + 0x267fa], rax
0x1052ce:    call    0x104be0
0x1052d3:    mov     edi, eax
0x1052d5:    call    0x104ae0
0x1052da:    call    0x104ee0
0x1052df:    mov     dword ptr [rsp + 0x1c], eax
0x1052e3:    call    0x104ee0
0x1052e8:    mov     dword ptr [rsp + 0x18], eax
0x1052ec:    call    0x104920
0x1052f1:    pxor    xmm0, xmm0
0x1052f5:    mov     r8, rbx
0x1052f8:    mov     rdi, qword ptr [rip + 0x267d1]
0x1052ff:    mov     r9d, eax
0x105302:    push    rax
0x105303:    mov     edx, dword ptr [rsp + 0x24]
0x105307:    lea     rcx, [rip + 0x1b361]
0x10530e:    mov     esi, 1
0x105313:    xor     eax, eax
0x105315:    cvtsi2sd    xmm0, edx
0x105319:    mulsd    xmm0, qword ptr [rip + 0x1b89f]
0x105321:    mulsd    xmm0, qword ptr [rip + 0x1b89f]
0x105329:    cvtttsd2si    edx, xmm0
0x10532d:    add     edx, 1
0x105330:    push    rdx
0x105331:    or      rdx, 0xffffffffffffffff
0x105335:    call    0x104f10
0x10533a:    call    0x104920
0x10533f:    mov     r10d, dword ptr [rsp + 0x28]
0x105344:    pxor    xmm0, xmm0
0x105348:    mov     rdi, qword ptr [rip + 0x26779]
0x10534f:    mov     r9d, eax
0x105352:    mov     r8, rbx
0x105355:    mov     esi, 1
0x10535a:    xor     eax, eax
0x10535c:    cvtsi2sd    xmm0, r10d
0x105361:    mulsd    xmm0, qword ptr [rip + 0x1b857]
0x105369:    lea     rcx, [rip + 0x1b30e]
0x105370:    mulsd    xmm0, qword ptr [rip + 0x1b850]
0x105378:    cvtttsd2si    edx, xmm0
0x10537c:    add     edx, 1
0x10537f:    mov     dword ptr [rsp], edx
0x105382:    or      rdx, 0xffffffffffffffff
0x105386:    call    0x104f10
0x10538b:    lea     rdi, [rip + 0x1b2fb]
0x105392:    call    0x117c50
0x105397:    pop     rdx
0x105398:    pop     rcx
0x105399:    cmp     r12d, -1

```

```

0x10539d:    je      0x107092
0x107092:    xor     esi, esi
0x107094:    lea     rdi, [rip + 0x195fa]
0x10709b:    call    0x117e50
0x1053a3:    cmp     dword ptr [rsp], 0
0x1053a7:    mov     dword ptr [rip + 0x248b2], r12d
0x1053ae:    jne     0x106f80
0x1070a0:    mov     r12d, eax
0x1070a3:    jmp     0x1053a3
0x1053b4:    mov     eax, dword ptr [rsp]
0x1053b7:    mov     edi, eax
0x1053b9:    mov     dword ptr [rip + 0x25519], eax
0x1053bf:    call    0x109af0
0x106f80:    xor     esi, esi
0x106f82:    lea     rdi, [rip + 0x19717]
0x106f89:    call    0x117e50
0x1053c4:    lea     rdi, [rip + 0x1b2e7]
0x1053cb:    call    0x117c50
0x106f8e:    mov     dword ptr [rsp], eax
0x106f91:    jmp     0x1053b4
0x1053d0:    xor     eax, eax
0x1053d2:    cmp     dword ptr [rip + 0x24887], 1
0x1053d9:    je      0x107050
0x107050:    mov     esi, 1
0x107055:    lea     rdi, [rip + 0x19664]
0x10705c:    call    0x117e50
0x1053df:    xor     esi, esi
0x1053e1:    lea     rdi, [rip + 0x1b2e8]
0x1053e8:    mov     dword ptr [rip + 0x2486e], eax
0x1053ee:    call    0x117e50
0x107061:    jmp     0x1053df
0x1053f3:    lea     rsi, [rip + 0x1cfeb]
0x1053fa:    lea     rdi, [rip + 0x1b2dd]
0x105401:    mov     dword ptr [rip + 0x25501], eax
0x105407:    call    0x117f10
0x10540c:    lea     rsi, [rip + 0x1cfd2]
0x105413:    lea     rdi, [rip + 0x1b2d3]
0x10541a:    mov     qword ptr [rip + 0x2484f], rax
0x105421:    call    0x117f10
0x105426:    cmp     dword ptr [rip + 0x24833], 0
0x10542d:    lea     rdi, [rip + 0x24814]
0x105434:    mov     qword ptr [rip + 0x2482d], rax
0x10543b:    lea     rax, [rip + 0x247ee]
0x105442:    cmov     rdi, rax
0x105446:    call    0x117c50
0x10544b:    mov     edx, 8
0x105450:    lea     rsi, [rip + 0x25229]
0x105457:    lea     rdi, [rip + 0x251e2]

```

```

0x10545e:    call    0x117f70
0x105463:    mov     rdi, rbp
0x105466:    call    0x117c50
0x10546b:    xor     eax, eax
0x10546d:    call    0x118660
0x105472:    lea     rdi, [rip + 0x1b1e2]
0x105479:    call    0x117c50
0x10547e:    mov     rsi, qword ptr [rsp + 0x10]
0x105483:    mov     edi, dword ptr [rsp + 0xc]
0x105487:    xor     eax, eax
0x105489:    call    0x109310
0x10548e:    mov     edx, dword ptr [rsp + 0x34]
0x105492:    add     dword ptr [rip + 0x25444], edx
0x105498:    cmp     eax, 0x800
0x10549d:    je      0x107066
0x107066:    mov     rcx, qword ptr [rip + 0x2387b]
0x10706d:    mov     rdi, qword ptr [rip + 0x2380c]
0x107074:    xor     eax, eax
0x107076:    mov     r8d, 0x800
0x10707c:    lea     rdx, [rip + 0x19895]
0x107083:    mov     esi, 1
0x107088:    call    0x104e60
0x1054a3:    call    0x1049e0
0x10708d:    jmp     0x1054a3
0x1054a8:    test    eax, eax
0x1054aa:    jne     0x1054b3
0x1054ac:    mov     byte ptr [rip + 0x24753], 0x23
0x1054b3:    xor     eax, eax
0x1054b5:    call    0x118fe0
0x1054b3:    xor     eax, eax
0x1054b5:    call    0x118fe0
0x1054ba:    mov     rbp, rax
0x1054bd:    test    rax, rax
0x1054c0:    je      0x1073ec
0x1073ec:    lea     rdi, [rip + 0x1955d]
0x1073f3:    xor     eax, eax
0x1073f5:    call    0x108b00
0x1054c6:    xor     eax, eax
0x1054c8:    call    0x109e00
0x1054cd:    xor     eax, eax
0x1054cf:    call    0x116c10
0x1054d4:    xor     eax, eax
0x1054d6:    call    0x10d060
0x1054db:    mov     rdi, qword ptr [rip + 0x2478e]
0x1054e2:    call    0x10ddc0
0x1054e7:    cmp     byte ptr [r13], 0x2f
0x1054ec:    je      0x107001
0x107001:    mov     rdi, r13

```

```

0x107004:      call    0x11d7a0
0x1054f2:      mov     rdi, rbp
0x1054f5:      call    0x1049b0
0x107009:      mov     r12, rax
0x10700c:      jmp     0x105532
0x1054fa:      mov     rdi, r13
0x1054fd:      mov     r12, rax
0x105500:      call    0x1049b0
0x105532:      mov     rdi, r12
0x105535:      call    0x10f160
0x105505:      lea     rdi, [r12 + rax + 2]
0x10550a:      call    0x11d670
0x10553a:      mov     rdi, r12
0x10553d:      mov     qword ptr [rip + 0x2537c], rax
0x105544:      call    0x118d10
0x10550f:      mov     r9, r13
0x105512:      mov     r8, rbp
0x105515:      or      rdx, 0xffffffffffffffff
0x105519:      mov     r12, rax
0x10551c:      mov     rdi, rax
0x10551f:      mov     esi, 1
0x105524:      xor     eax, eax
0x105526:      lea     rcx, [rip + 0x1b1d4]
0x10552d:      call    0x104f10
0x105549:      cmp     byte ptr [r14], 0x2f
0x10554d:      je      0x106f70
0x106f70:      mov     rdi, r14
0x106f73:      call    0x11d7a0
0x105553:      mov     rdi, rbp
0x105556:      call    0x1049b0
0x106f78:      mov     r12, rax
0x106f7b:      jmp     0x105593
0x10555b:      mov     rdi, r14
0x10555e:      mov     r12, rax
0x105561:      call    0x1049b0
0x105593:      mov     rdi, r12
0x105596:      xor     r14d, r14d
0x105599:      lea     rbx, [rsp + 0x28]
0x10559e:      call    0x10f160
0x105566:      lea     rdi, [r12 + rax + 2]
0x10556b:      call    0x11d670
0x1055a3:      mov     rdi, r12
0x1055a6:      mov     qword ptr [rip + 0x2530b], rax
0x1055ad:      call    0x118d10
0x105570:      mov     r9, r14
0x105573:      mov     r8, rbp
0x105576:      or      rdx, 0xffffffffffffffff
0x10557a:      mov     r12, rax

```

```

0x10557d:    mov     rdi, rax
0x105580:    mov     esi, 1
0x105585:    xor     eax, eax
0x105587:    lea     rcx, [rip + 0x1b173]
0x10558e:    call    0x104f10
0x1055b2:    mov     rdi, rbp
0x1055b5:    mov     ebp, 1
0x1055ba:    call    0x118d10
0x1055bf:    xor     eax, eax
0x1055c1:    call    0x117280
0x1055c6:    mov     rax, qword ptr [rip + 0x252f3]
0x1055cd:    xor     edi, edi
0x1055cf:    mov     qword ptr [rip + 0x252da], rax
0x1055d6:    mov     rax, qword ptr [rip + 0x252db]
0x1055dd:    mov     qword ptr [rip + 0x252c4], rax
0x1055e4:    xor     eax, eax
0x1055e6:    call    0x107890
0x1055eb:    mov     rdi, qword ptr [rip + 0x252d6]
0x1055f2:    call    0x10bb10
0x1055f7:    mov     edi, 1
0x1055fc:    call    0x109b00
0x105601:    xor     eax, eax
0x105603:    call    0x10a6c0
0x105608:    mov     edi, 1
0x10560d:    mov     qword ptr [rip + 0x25280], 0
0x105618:    mov     dword ptr [rip + 0x25272], 0
0x105622:    mov     dword ptr [rip + 0x25264], 0
0x10562c:    call    0x118b90
0x105631:    mov     rdi, qword ptr [rip + 0x25278]
0x105638:    call    0x111100
0x10563d:    mov     rdi, rax
0x105640:    call    0x10c4b0
0x105645:    lea     rax, [rsp + 0x50]
0x10564a:    mov     dword ptr [rsp + 0x10], r15d
0x10564f:    mov     qword ptr [rsp], 0
0x105657:    mov     r15, rax
0x10565a:    cmp     dword ptr [rip + 0x2527b], 0
0x105661:    jne     0x106d10
0x105667:    mov     rdi, qword ptr [rip + 0x25242]
0x10566e:    call    0x111100
0x106d10:    xor     edi, edi
0x106d12:    call    0x104aa0
0x105673:    mov     rdi, rax
0x105676:    call    0x10c4b0
0x106d17:    mov     edx, 0x21
0x106d1c:    mov     rcx, qword ptr [rip + 0x23b1d]
0x106d23:    mov     esi, 1
0x106d28:    lea     rdi, [rip + 0x19c49]

```

```

0x106d2f:    call    0x104e40
0x10567b:    xor     eax, eax
0x10567d:    call    0x118330
0x106d34:    mov     eax, dword ptr [rip + 0x23c62]
0x106d3a:    xor     esi, esi
0x106d3c:    lea     edi, [rax - 1]
0x106d3f:    call    0x10a4b0
0x105682:    mov     edi, 0x3c
0x105687:    sub     edi, dword ptr [rax]
0x105689:    call    0x104aa0
0x106d44:    xor     edi, edi
0x106d46:    call    0x10b180
0x10568e:    cmp     dword ptr [rip + 0x2524b], 0
0x105695:    je      0x106ba2
0x106d4b:    mov     dword ptr [rip + 0x23b87], 0
0x106d55:    jmp     0x105667
0x106ba2:    mov     rax, qword ptr [rip + 0x23d17]
0x106ba9:    mov     qword ptr [rip + 0x23d00], rax
0x106bb0:    mov     rax, qword ptr [rip + 0x23d01]
0x106bb7:    jmp     0x1056b0
0x10569b:    mov     rax, qword ptr [rip + 0x25216]
0x1056a2:    mov     qword ptr [rip + 0x25207], rax
0x1056a9:    mov     rax, qword ptr [rip + 0x25210]
0x1056b0:    cmp     dword ptr [rip + 0x252e5], 6
0x1056b7:    mov     qword ptr [rip + 0x251ea], rax
0x1056be:    jle     0x106d75
0x1056b0:    cmp     dword ptr [rip + 0x252e5], 6
0x1056b7:    mov     qword ptr [rip + 0x251ea], rax
0x1056be:    jle     0x106d75
0x106d75:    xor     eax, eax
0x106d77:    call    0x10a6c0
0x1056c4:    xor     eax, eax
0x1056c6:    call    0x116d50
0x106d7c:    xor     eax, eax
0x106d7e:    call    0x10a300
0x1056cb:    xor     eax, eax
0x1056cd:    call    0x10e4e0
0x106d83:    jmp     0x1056c4
0x1056d2:    mov     rdi, r14
0x1056d5:    call    0x10c970
0x1056da:    xor     eax, eax
0x1056dc:    call    0x10a820
0x1056e1:    xor     eax, eax
0x1056e3:    call    0x10c8c0
0x1056e8:    mov     r12, rax
0x1056eb:    xor     eax, eax
0x1056ed:    call    0x1096c0
0x1056f2:    xor     edi, edi

```

```

0x1056f4:    xor     eax, eax
0x1056f6:    call   0x107bd0
0x1056fb:    mov     rdi, r12
0x1056fe:    call   0x10c970
0x105703:    mov     rdi, qword ptr [rip + 0x251a6]
0x10570a:    mov     esi, 1
0x10570f:    call   0x114d90
0x105714:    test    ebp, ebp
0x105716:    jne     0x106d5a
0x10571c:    xor     eax, eax
0x10571e:    call   0x109640
0x106d5a:    mov     rdi, qword ptr [rip + 0x23b4f]
0x106d61:    call   0x111100
0x105723:    xor     eax, eax
0x105725:    call   0x10c8c0
0x106d66:    mov     rdi, rax
0x106d69:    xor     eax, eax
0x106d6b:    call   0x1089d0
0x10572a:    mov     r14, rax
0x10572d:    nop     dword ptr [rax]
0x105730:    mov     rdi, r14
0x105733:    call   0x10c970
0x106d70:    jmp     0x10571c
0x105738:    xor     eax, eax
0x10573a:    call   0x10c8c0
0x10573f:    mov     r14, rax
0x105742:    xor     eax, eax
0x105744:    call   0x10d870
0x105749:    xor     eax, eax
0x10574b:    call   0x10d740
0x105750:    xor     eax, eax
0x105752:    call   0x10a820
0x105757:    mov     rdi, r15
0x10575a:    call   0x10db10
0x10575f:    mov     dword ptr [rip + 0x26347], 0
0x105769:    jmp     0x105779
0x105779:    mov     rdi, rbx
0x10577c:    call   0x10b180
0x105781:    test    rax, rax
0x105784:    je      0x105770
0x105770:    xor     edi, edi
0x105772:    xor     eax, eax
0x105774:    call   0x107d10
0x105786:    mov     r12, rax
0x105789:    xor     eax, eax
0x10578b:    call   0x10dea0
0x105790:    mov     rax, qword ptr [r12]
0x105794:    mov     r13, qword ptr [r12 + 0x10]

```

```

0x105799:    movzx    ebp, byte ptr [rax]
0x10579c:    test     r13, r13
0x10579f:    je       0x1057dc
0x1057dc:    lea      eax, [rbp + 0x57]
0x1057df:    cmp      eax, 0x5e
0x1057e2:    ja       0x1057f5
0x1057a1:    cmp      byte ptr [r13 + 0x1b], 0
0x1057a6:    mov      rax, qword ptr [r13]
0x1057aa:    je       0x1067c0
0x1057e4:    lea      rcx, [rip + 0x1aaf1]
0x1057eb:    movsxd   rax, dword ptr [rcx + rax*4]
0x1057ef:    add      rax, rcx
0x1057f2:    jmp      rax
0x1057f5:    test     ebp, ebp
0x1057f7:    je       0x105730
0x1067c0:    test     rax, rax
0x1067c3:    je       0x1057dc
0x1057b0:    lea      rdx, [rip + 0x23869]
0x1057b7:    sub      rax, rdx
0x1057ba:    movabs   rdx, 0xea0ea0ea0ea0ea1
0x1057c4:    mov      rcx, rax
0x1057c7:    imul     rdx
0x1057ca:    mov      rax, rcx
0x1057cd:    sar      rax, 0x3f
0x1057d1:    sar      rdx, 1
0x1057d4:    mov      rbp, rdx
0x1057d7:    sub      rbp, rax
0x1057da:    not      ebp
0x1057dc:    lea      eax, [rbp + 0x57]
0x1057df:    cmp      eax, 0x5e
0x1057e2:    ja       0x1057f5
0x1066c1:    mov      rdi, r14
0x1066c4:    call     0x10c7e0
0x105730:    mov      rdi, r14
0x105733:    call     0x10c970
0x1057fd:    mov      rdi, r14
0x105800:    call     0x10c7e0
0x1067c9:    mov      rdi, qword ptr [rip + 0x240e0]
0x1067d0:    call     0x10f480
0x1066c9:    xor      edi, edi
0x1066cb:    call     0x10d1a0
0x105805:    jmp      0x105813
0x1067d5:    mov      rdi, qword ptr [rip + 0x240cc]
0x1067dc:    call     0x10f480
0x1066d0:    xor      eax, eax
0x1066d2:    call     0x10c8c0
0x105813:    mov      eax, dword ptr [rsp + 0x28]
0x105817:    lea      edx, [rax - 1]

```



```

0x10581a:    mov     dword ptr [rsp + 0x28], edx
0x10581e:    test    eax, eax
0x105820:    jne     0x105807
0x1067e1:    cmp     qword ptr [r13 + 8], 0
0x1067e6:    je      0x1070b7
0x1066d7:    mov     r14, rax
0x1066da:    jmp     0x105730
0x105822:    jmp     0x105725
0x105807:    mov     esi, 3
0x10580c:    mov     edi, ebp
0x10580e:    call    0x107570
0x1070b7:    mov     rdi, qword ptr [r13 + 0x10]
0x1070bb:    test    rdi, rdi
0x1070be:    je      0x106955
0x1067ec:    lea     rbp, [rsp + 0x60]
0x1067f1:    mov     rcx, qword ptr [rip + 0x240b0]
0x1067f8:    xor     eax, eax
0x1067fa:    mov     rdi, r13
0x1067fd:    mov     rdx, qword ptr [rip + 0x240ac]
0x106804:    mov     rsi, rbp
0x106807:    mov     qword ptr [rsp + 0x60], 0
0x106810:    call    0x108b50
0x105725:    call    0x10c8c0
0x106955:    xor     ebp, ebp
0x106957:    jmp     0x10565a
0x1070c4:    xor     eax, eax
0x1070c6:    call    0x119940
0x106815:    test    eax, eax
0x106817:    je      0x106c1f
0x10565a:    cmp     dword ptr [rip + 0x2527b], 0
0x105661:    jne     0x106d10
0x1070cb:    mov     r8d, 1
0x1070d1:    mov     esi, 0x21
0x1070d6:    mov     rdx, qword ptr [rip + 0x237cb]
0x1070dd:    mov     rdi, qword ptr [rip + 0x237cc]
0x1070e4:    mov     rcx, rax
0x1070e7:    mov     rbp, rax
0x1070ea:    call    0x113e30
0x106c1f:    mov     rdi, qword ptr [r12]
0x106c23:    call    0x1049b0
0x10681d:    jle     0x105730
0x1070ef:    mov     rdi, qword ptr [rip + 0x237ba]
0x1070f6:    call    0x111100
0x106c28:    lea     rdi, [rax + 0x51]
0x106c2c:    call    0x11d670
0x106823:    mov     rdi, qword ptr [r13]
0x106827:    mov     dword ptr [rsp + 0xc], eax
0x10682b:    call    0x1049b0

```

```

0x1070fb:    mov     rdi, rax
0x1070fe:    xor     eax, eax
0x107100:    call    0x1089d0
0x106c31:    mov     r9, qword ptr [r13 + 0x20]
0x106c35:    mov     r8, qword ptr [r13]
0x106c39:    lea     rcx, [rip + 0x19d80]
0x106c40:    mov     rbp, rax
0x106c43:    or      rdx, 0xfffffffffffffff
0x106c47:    mov     rdi, rax
0x106c4a:    mov     esi, 1
0x106c4f:    xor     eax, eax
0x106c51:    call    0x104f10
0x106830:    mov     rbp, qword ptr [rsp + 0x60]
0x106835:    mov     r12, rax
0x106838:    mov     rdi, rbp
0x10683b:    call    0x1049b0
0x107105:    mov     rdi, rbp
0x107108:    xor     ebp, ebp
0x10710a:    call    0x118d10
0x106c56:    mov     rdi, rbp
0x106c59:    mov     edx, 0x1c
0x106c5e:    xor     esi, esi
0x106c60:    call    0x108070
0x106840:    lea     rdi, [r12 + rax + 0x21]
0x106845:    call    0x11d670
0x10710f:    jmp     0x10565a
0x106c65:    mov     rdi, rbp
0x106c68:    call    0x118d10
0x10684a:    mov     r8, qword ptr [r13]
0x10684e:    mov     r9, rbp
0x106851:    or      rdx, 0xfffffffffffffff
0x106855:    lea     rcx, [rip + 0x19cd9]
0x10685c:    mov     r12, rax
0x10685f:    mov     rdi, rax
0x106862:    mov     esi, 1
0x106867:    xor     eax, eax
0x106869:    call    0x104f10
0x106c6d:    jmp     0x105730
0x10686e:    mov     edx, 1
0x106873:    mov     esi, 1
0x106878:    mov     rdi, r12
0x10687b:    call    0x10e480
0x106880:    xor     eax, eax
0x106882:    call    0x10a820
0x106887:    mov     rdi, r12
0x10688a:    call    0x118d10
0x10688f:    cmp     byte ptr [r13 + 0x1a], 0
0x106894:    jne     0x106dc9

```

```

0x10689a:    mov     rdi, rbp
0x10689d:    call   0x1185d0
0x106dc9:    mov     rdi, qword ptr [r13]
0x106dcd:    call   0x1049b0
0x1068a2:    test    eax, eax
0x1068a4:    je      0x106dbc
0x106dd2:    lea     rdi, [rax + 0x41]
0x106dd6:    call   0x11d670
0x106dbc:    mov     rdi, rbp
0x106dbf:    call   0x10c4b0
0x1068aa:    movsx   esi, byte ptr [r13 + 0x1a]
0x1068af:    mov     rdi, rbp
0x1068b2:    call   0x1193f0
0x106ddb:    mov     r9, qword ptr [r13]
0x106ddf:    lea     r8, [rip + 0x19921]
0x106de6:    lea     rcx, [rip + 0x19bb3]
0x106ded:    mov     r12, rax
0x106df0:    or      rdx, 0xffffffffffffffff
0x106df4:    mov     esi, 1
0x106df9:    mov     rdi, rax
0x106dfc:    xor     eax, eax
0x106dfe:    call   0x104f10
0x106dc4:    jmp     0x1068aa
0x1068b7:    mov     rdi, rbp
0x1068ba:    mov     r12d, eax
0x1068bd:    call   0x118d10
0x106e03:    mov     rdi, r12
0x106e06:    call   0x10db60
0x1068c2:    cmp     byte ptr [r13 + 0x1a], 0
0x1068c7:    je      0x106ccf
0x106e0b:    xor     eax, eax
0x106e0d:    call   0x10a820
0x106ccf:    cmp     byte ptr [r13 + 0x18], 0
0x106cd4:    jne     0x106f35
0x1068cd:    mov     eax, r12d
0x1068d0:    and     eax, 0x7f
0x1068d3:    add     eax, 1
0x1068d6:    sub     al, 1
0x1068d8:    jg      0x106f1d
0x106e12:    mov     rdi, r12
0x106e15:    call   0x118d10
0x106cda:    xor     eax, eax
0x106cdc:    call   0x10a480
0x106f35:    mov     rdi, qword ptr [rip + 0x2398c]
0x106f3c:    call   0x10bb10
0x1068de:    test    r12b, 0x7f
0x1068e2:    jne     0x106955
0x106f1d:    mov     edx, 0xc

```

```

0x106f22:    xor     esi, esi
0x106f24:    lea     rdi, [rip + 0x197e9]
0x106f2b:    call    0x108070
0x106e1a:    jmp     0x10689a
0x106ce1:    cmp     byte ptr [r13 + 0x19], 0
0x106ce6:    je      0x106cf2
0x106f41:    jmp     0x106cda
0x1068e4:    and     r12d, 0xff00
0x1068eb:    jne     0x10719d
0x106f30:    jmp     0x106cf2
0x106cf2:    test    r12b, 0x7f
0x106cf6:    jne     0x106955
0x106ce8:    mov     dword ptr [rip + 0x23bea], 1
0x106cf2:    test    r12b, 0x7f
0x106cf6:    jne     0x106955
0x1068f1:    mov     rdi, qword ptr [r13 + 0x10]
0x1068f5:    test    rdi, rdi
0x1068f8:    je      0x106943
0x10719d:    xor     eax, eax
0x10719f:    xor     ebp, ebp
0x1071a1:    call    0x10a5a0
0x106cfc:    and     r12d, 0xff00
0x106d03:    je      0x1068f1
0x106943:    cmp     dword ptr [rsp + 0xc], 2
0x106948:    je      0x1071b4
0x1068fa:    xor     eax, eax
0x1068fc:    call    0x119940
0x1071a6:    mov     rdi, qword ptr [r13]
0x1071aa:    call    0x1195e0
0x106d09:    xor     ebp, ebp
0x106d0b:    jmp     0x10565a
0x1071b4:    mov     rdi, qword ptr [rip + 0x236f5]
0x1071bb:    xor     ebp, ebp
0x1071bd:    call    0x1110b0
0x10694e:    cmp     dword ptr [rsp + 0xc], 3
0x106953:    je      0x10695c
0x106901:    mov     r8d, 1
0x106907:    mov     esi, 0x21
0x10690c:    mov     rdx, qword ptr [rip + 0x23f95]
0x106913:    mov     rdi, qword ptr [rip + 0x23f96]
0x10691a:    mov     rcx, rax
0x10691d:    mov     r12, rax
0x106920:    call    0x113e30
0x1071af:    jmp     0x10565a
0x1071c2:    jmp     0x10565a
0x10695c:    mov     rdi, qword ptr [rip + 0x23f45]
0x106963:    xor     ebp, ebp
0x106965:    call    0x1110b0

```

```

0x106925:    mov     rdi, qword ptr [rip + 0x23f84]
0x10692c:    call   0x111100
0x10696a:    jmp     0x10565a
0x106931:    mov     rdi, rax
0x106934:    xor     eax, eax
0x106936:    call   0x1089d0
0x10693b:    mov     rdi, r12
0x10693e:    call   0x118d10
0x10576b:    nop     dword ptr [rax + rax]
0x105827:    mov     esi, 0xffffffff
0x10582c:    mov     rdi, qword ptr [rip + 0x2507d]
0x105833:    sub     esi, ebp
0x105835:    xor     ecx, ecx
0x105837:    xor     edx, edx
0x105839:    mov     r8d, 1
0x10583f:    call   0x113e30
0x105844:    jmp     0x105730
0x105849:    mov     esi, 0xffffffff
0x10584e:    jmp     0x10582c
0x10582c:    mov     rdi, qword ptr [rip + 0x2507d]
0x105833:    sub     esi, ebp
0x105835:    xor     ecx, ecx
0x105837:    xor     edx, edx
0x105839:    mov     r8d, 1
0x10583f:    call   0x113e30
0x105850:    mov     rdi, qword ptr [rip + 0x25059]
0x105857:    xor     esi, esi
0x105859:    call   0x114d90
0x10585e:    mov     rax, qword ptr [rip + 0x2504b]
0x105865:    mov     esi, 1
0x10586a:    mov     rdi, qword ptr [rip + 0x25037]
0x105871:    mov     qword ptr [rip + 0x25028], rax
0x105878:    mov     qword ptr [rip + 0x25029], rax
0x10587f:    xor     eax, eax
0x105881:    cmp     dword ptr [rip + 0x25058], 0
0x105888:    sete    al
0x10588b:    mov     qword ptr [rip + 0x2501e], rdi
0x105892:    mov     dword ptr [rip + 0x25048], eax
0x105898:    call   0x114d90
0x10589d:    mov     rdi, qword ptr [rip + 0x2500c]
0x1058a4:    call   0x10f570
0x1058a9:    mov     rdi, r14
0x1058ac:    call   0x10c7e0
0x1058b1:    xor     eax, eax
0x1058b3:    call   0x109640
0x1058b8:    xor     eax, eax
0x1058ba:    call   0x10c8c0
0x1058bf:    mov     rdi, qword ptr [rip + 0x24fea]

```

```

0x1058c6:    mov     r14, rax
0x1058c9:    call    0x111100
0x1058ce:    mov     rdi, rax
0x1058d1:    call    0x10c4b0
0x1058d6:    mov     rdi, qword ptr [rip + 0x24fd3]
0x1058dd:    call    0x10f480
0x1058e2:    mov     rdi, qword ptr [rip + 0x24fbf]
0x1058e9:    call    0x10f480
0x1058ee:    xor     eax, eax
0x1058f0:    call    0x10a480
0x1058f5:    mov     rdi, qword ptr [rip + 0x24fac]
0x1058fc:    call    0x10f580
0x105901:    xor     eax, eax
0x105903:    mov     edi, 1
0x105908:    mov     dword ptr [rip + 0x24346], 0
0x105912:    call    0x107890
0x105917:    mov     rdi, qword ptr [rip + 0x24f92]
0x10591e:    xor     ecx, ecx
0x105920:    xor     edx, edx
0x105922:    mov     r8d, 1
0x105928:    mov     esi, 0xe
0x10592d:    call    0x113e30
0x105932:    mov     rdi, qword ptr [rip + 0x24f6f]
0x105939:    xor     ecx, ecx
0x10593b:    xor     edx, edx
0x10593d:    mov     r8d, 1
0x105943:    mov     esi, 0xe
0x105948:    call    0x113e30
0x10594d:    mov     rdi, qword ptr [rip + 0x24f5c]
0x105954:    call    0x111010
0x105959:    jmp     0x105730
0x10595e:    xor     eax, eax
0x105960:    call    0x108960
0x105965:    mov     rdi, qword ptr [rip + 0x24f44]
0x10596c:    call    0x111100
0x105971:    mov     rdi, rax
0x105974:    xor     eax, eax
0x105976:    call    0x1089d0
0x10597b:    jmp     0x105730
0x105980:    mov     rdi, qword ptr [rip + 0x24f29]
0x105987:    call    0x10f480
0x10598c:    mov     rdi, qword ptr [rip + 0x24f15]
0x105993:    call    0x10f480
0x105998:    xor     eax, eax
0x10599a:    call    0x10a480
0x10599f:    mov     rdi, qword ptr [rip + 0x24f02]
0x1059a6:    call    0x10f570
0x1059ab:    xor     eax, eax

```

```

0x1059ad:    mov     edi, 1
0x1059b2:    mov     dword ptr [rip + 0x2429c], 1
0x1059bc:    call    0x107890
0x1059c1:    cmp     dword ptr [rip + 0x24fd0], 0xb
0x1059c8:    jle     0x10703f
0x10703f:    mov     edi, 1
0x107044:    xor     eax, eax
0x107046:    call    0x107bd0
0x1059ce:    mov     rdi, qword ptr [rip + 0x24edb]
0x1059d5:    xor     ecx, ecx
0x1059d7:    xor     edx, edx
0x1059d9:    mov     esi, 0xb
0x1059de:    mov     r8d, 1
0x1059e4:    call    0x113e30
0x10704b:    jmp     0x1059ce
0x1059e9:    mov     rdi, qword ptr [rip + 0x24eb8]
0x1059f0:    xor     ecx, ecx
0x1059f2:    xor     edx, edx
0x1059f4:    mov     r8d, 1
0x1059fa:    mov     esi, 0xb
0x1059ff:    call    0x113e30
0x105a04:    mov     rdi, qword ptr [rip + 0x24ea5]
0x105a0b:    call    0x111010
0x105a10:    mov     rdi, qword ptr [rip + 0x24e91]
0x105a17:    call    0x111010
0x105a1c:    jmp     0x105730
0x105a21:    xor     edi, edi
0x105a23:    call    0x10ca20
0x105a28:    lea     rsi, [rsp + 0x38]
0x105a2d:    xor     ecx, ecx
0x105a2f:    xor     edx, edx
0x105a31:    mov     qword ptr [rsp + 0x38], 0
0x105a3a:    lea     rdi, [rip + 0x1ad44]
0x105a41:    call    0x1083e0
0x105a46:    mov     rax, qword ptr [rsp + 0x38]
0x105a4b:    test    rax, rax
0x105a4e:    je      0x105a59
0x105a59:    mov     edi, 1
0x105a5e:    call    0x10ca20
0x105a50:    cmp     byte ptr [rax], 0
0x105a53:    jne     0x107114
0x105a63:    jmp     0x105730
0x107114:    mov     qword ptr [rsp + 0x40], 0
0x10711d:    lea     r12, [rsp + 0x40]
0x107122:    lea     rbp, [rip + 0x1966f]
0x107129:    jmp     0x10714f
0x10714f:    mov     rsi, r12
0x107152:    xor     ecx, ecx

```

```

0x107154:    xor     edx, edx
0x107156:    mov     rdi, rbp
0x107159:    call    0x1083e0
0x10715e:    xor     eax, eax
0x107160:    call    0x10a820
0x107165:    mov     rsi, qword ptr [rsp + 0x40]
0x10716a:    test    rsi, rsi
0x10716d:    je      0x10712b
0x10712b:    lea     rdi, [rip + 0x22ade]
0x107132:    call    0x10db60
0x10716f:    mov     rdi, qword ptr [rsp + 0x38]
0x107174:    call    0x104b30
0x107137:    xor     eax, eax
0x107139:    call    0x10a5a0
0x107179:    test    eax, eax
0x10717b:    jne     0x10712b
0x10713e:    xor     eax, eax
0x107140:    call    0x10a820
0x10717d:    xor     eax, eax
0x10717f:    call    0x10a820
0x107145:    mov     edi, 2
0x10714a:    call    0x104ea0
0x107184:    mov     rdi, qword ptr [rsp + 0x38]
0x107189:    call    0x118d10
0x10718e:    mov     rdi, qword ptr [rsp + 0x40]
0x107193:    call    0x118d10
0x107198:    jmp     0x105a59
0x105a68:    mov     dword ptr [rsp + 0x30], 0
0x105a70:    lea     rcx, [rsp + 0x30]
0x105a75:    mov     rdx, qword ptr [rip + 0x24e2c]
0x105a7c:    mov     esi, 0x2b
0x105a81:    mov     rdi, qword ptr [rip + 0x24e28]
0x105a88:    mov     r8d, 1
0x105a8e:    call    0x113e30
0x105a93:    jmp     0x105730
0x105a98:    mov     dword ptr [rsp + 0x2c], 0
0x105aa0:    lea     rcx, [rsp + 0x2c]
0x105aa5:    mov     rdx, qword ptr [rip + 0x24dfc]
0x105aac:    mov     esi, 0x2c
0x105ab1:    mov     rdi, qword ptr [rip + 0x24df8]
0x105ab8:    mov     r8d, 1
0x105abe:    call    0x113e30
0x105ac3:    jmp     0x105730
0x105ac8:    mov     dword ptr [rsp + 0x30], 1
0x105ad0:    jmp     0x105a70
0x105a70:    lea     rcx, [rsp + 0x30]
0x105a75:    mov     rdx, qword ptr [rip + 0x24e2c]
0x105a7c:    mov     esi, 0x2b

```



```

0x105a81:    mov     rdi, qword ptr [rip + 0x24e28]
0x105a88:    mov     r8d, 1
0x105a8e:    call    0x113e30
0x105ad2:    mov     dword ptr [rsp + 0x2c], 1
0x105ada:    jmp     0x105aa0
0x105aa0:    lea     rcx, [rsp + 0x2c]
0x105aa5:    mov     rdx, qword ptr [rip + 0x24dfc]
0x105aac:    mov     esi, 0x2c
0x105ab1:    mov     rdi, qword ptr [rip + 0x24df8]
0x105ab8:    mov     r8d, 1
0x105abe:    call    0x113e30
0x105adc:    mov     rdx, qword ptr [rip + 0x24dc5]
0x105ae3:    mov     rdi, qword ptr [rip + 0x24dc6]
0x105aea:    xor     ecx, ecx
0x105aec:    mov     esi, 0x2d
0x105af1:    mov     r8d, 1
0x105af7:    call    0x113e30
0x105afc:    mov     rdi, r14
0x105aff:    call    0x10c970
0x105b04:    xor     eax, eax
0x105b06:    call    0x109640
0x105b0b:    xor     eax, eax
0x105b0d:    call    0x10c8c0
0x105b12:    mov     rdi, qword ptr [rip + 0x24d97]
0x105b19:    mov     r14, rax
0x105b1c:    call    0x111100
0x105b21:    mov     rdi, rax
0x105b24:    call    0x10c4b0
0x105b29:    jmp     0x105730
0x105b2e:    mov     rdx, qword ptr [rip + 0x24d73]
0x105b35:    mov     rdi, qword ptr [rip + 0x24d74]
0x105b3c:    xor     ecx, ecx
0x105b3e:    mov     esi, 0x2e
0x105b43:    mov     r8d, 1
0x105b49:    call    0x113e30
0x105b4e:    jmp     0x105730
0x105b53:    lea     r12, [rsp + 0x58]
0x105b58:    mov     rcx, qword ptr [r13 + 0x28]
0x105b5c:    lea     rdx, [rip + 0x1b230]
0x105b63:    mov     rsi, r12
0x105b66:    lea     rdi, [rip + 0x1ac46]
0x105b6d:    call    0x1083e0
0x105b72:    test    rax, rax
0x105b75:    je      0x105730
0x105b7b:    mov     rcx, qword ptr [rsp + 0x58]
0x105b80:    cmp     byte ptr [rcx], 0
0x105b83:    je      0x105730
0x105b89:    mov     rdx, qword ptr [rip + 0x24d18]

```

```

0x105b90:    mov     r8d, 1
0x105b96:    mov     esi, 0x2f
0x105b9b:    mov     rdi, qword ptr [rip + 0x24d0e]
0x105ba2:    call    0x113e30
0x105ba7:    mov     rdi, qword ptr [rsp + 0x58]
0x105bac:    call    0x118d10
0x105bb1:    mov     qword ptr [rsp + 0x58], 0
0x105bba:    jmp     0x105730
0x105bbf:    mov     r8d, dword ptr [rsp + 0x28]
0x105bc4:    mov     rcx, qword ptr [rsp + 0x58]
0x105bc9:    mov     esi, 0x30
0x105bce:    mov     rdx, qword ptr [rip + 0x24cd3]
0x105bd5:    mov     rdi, qword ptr [rip + 0x24cd4]
0x105bdc:    call    0x113e30
0x105be1:    jmp     0x105730
0x105be6:    mov     r8d, dword ptr [rsp + 0x28]
0x105beb:    mov     rcx, qword ptr [rsp + 0x58]
0x105bf0:    mov     esi, 0x31
0x105bf5:    mov     rdx, qword ptr [rip + 0x24cac]
0x105bfc:    mov     rdi, qword ptr [rip + 0x24cad]
0x105c03:    call    0x113e30
0x105c08:    jmp     0x105730
0x105c0d:    mov     rdx, qword ptr [rip + 0x24c94]
0x105c14:    mov     rdi, qword ptr [rip + 0x24c95]
0x105c1b:    xor     ecx, ecx
0x105c1d:    mov     esi, 0x32
0x105c22:    mov     r8d, 1
0x105c28:    call    0x113e30
0x105c2d:    jmp     0x105730
0x105c32:    mov     rdx, qword ptr [rip + 0x24c6f]
0x105c39:    mov     rdi, qword ptr [rip + 0x24c70]
0x105c40:    xor     ecx, ecx
0x105c42:    mov     esi, 0x33
0x105c47:    mov     r8d, 1
0x105c4d:    call    0x113e30
0x105c52:    jmp     0x105730
0x105c57:    mov     r8d, dword ptr [rsp + 0x28]
0x105c5c:    mov     rdx, qword ptr [rip + 0x24c45]
0x105c63:    xor     ecx, ecx
0x105c65:    mov     esi, 0x1b
0x105c6a:    mov     rdi, qword ptr [rip + 0x24c3f]
0x105c71:    call    0x113e30
0x105c76:    jmp     0x105730
0x105c7b:    mov     r8d, dword ptr [rsp + 0x28]
0x105c80:    mov     rdx, qword ptr [rip + 0x24c21]
0x105c87:    xor     ecx, ecx
0x105c89:    mov     esi, 0x1c
0x105c8e:    mov     rdi, qword ptr [rip + 0x24c1b]

```

```

0x105c95:    call    0x113e30
0x105c9a:    jmp     0x105730
0x105c9f:    cmp     dword ptr [rip + 0x23fb2], 0
0x105ca6:    je      0x105850
0x105cac:    test    byte ptr [rsp + 0x28], 1
0x105cb1:    je      0x105730
0x105cb7:    mov     rdi, qword ptr [rip + 0x24bf2]
0x105cbe:    xor     esi, esi
0x105cc0:    call    0x114d90
0x105cc5:    mov     rax, qword ptr [rip + 0x24be4]
0x105ccc:    mov     esi, 1
0x105cd1:    mov     rdi, qword ptr [rip + 0x24bd0]
0x105cd8:    mov     qword ptr [rip + 0x24bc1], rax
0x105cdf:    mov     qword ptr [rip + 0x24bc2], rax
0x105ce6:    xor     eax, eax
0x105ce8:    cmp     dword ptr [rip + 0x24bf1], 0
0x105cef:    sete    al
0x105cf2:    mov     qword ptr [rip + 0x24bb7], rdi
0x105cf9:    mov     dword ptr [rip + 0x24be1], eax
0x105cff:    call    0x114d90
0x105d04:    mov     rdi, r14
0x105d07:    call    0x10c7e0
0x105d0c:    jmp     0x105b04
0x105d11:    lea     r12, [rsp + 0x58]
0x105d16:    mov     rcx, qword ptr [r13 + 0x28]
0x105d1a:    xor     edx, edx
0x105d1c:    lea     rdi, [rip + 0x1aa29]
0x105d23:    mov     rsi, r12
0x105d26:    call    0x1083e0
0x105d2b:    test    rax, rax
0x105d2e:    je      0x105730
0x105d34:    mov     rdi, qword ptr [rsp + 0x58]
0x105d39:    cmp     byte ptr [rdi], 0
0x105d3c:    je      0x105730
0x105d42:    xor     eax, eax
0x105d44:    call    0x119940
0x105d49:    mov     r8d, 1
0x105d4f:    mov     esi, 0x21
0x105d54:    mov     rdx, qword ptr [rip + 0x24b4d]
0x105d5b:    mov     rdi, qword ptr [rip + 0x24b4e]
0x105d62:    mov     rcx, rax
0x105d65:    mov     rbp, rax
0x105d68:    call    0x113e30
0x105d6d:    mov     rdi, qword ptr [rip + 0x24b3c]
0x105d74:    call    0x111100
0x105d79:    mov     rdi, rax
0x105d7c:    xor     eax, eax
0x105d7e:    call    0x1089d0

```

```

0x105d83:    mov     rdi, rbp
0x105d86:    call   0x118d10
0x105d8b:    mov     rdi, qword ptr [rsp + 0x58]
0x105d90:    call   0x118d10
0x105d95:    mov     qword ptr [rsp + 0x58], 0
0x105d9e:    jmp     0x105afc
0x105da3:    lea     r12, [rsp + 0x58]
0x105da8:    mov     rcx, qword ptr [r13 + 0x28]
0x105dac:    xor     edx, edx
0x105dae:    lea     rdi, [rip + 0x1ac3b]
0x105db5:    mov     rsi, r12
0x105db8:    call   0x1083e0
0x105dbd:    test    rax, rax
0x105dc0:    je      0x105730
0x105dc6:    mov     rcx, qword ptr [rsp + 0x58]
0x105dcb:    cmp     byte ptr [rcx], 0
0x105dce:    je      0x105730
0x105dd4:    mov     rdx, qword ptr [rip + 0x24acd]
0x105ddb:    mov     r8d, 1
0x105de1:    mov     esi, 0x24
0x105de6:    jmp     0x105b9b
0x105b9b:    mov     rdi, qword ptr [rip + 0x24d0e]
0x105ba2:    call   0x113e30
0x105deb:    lea     r12, [rsp + 0x58]
0x105df0:    mov     rcx, qword ptr [r13 + 0x28]
0x105df4:    xor     edx, edx
0x105df6:    lea     rdi, [rip + 0x1ac23]
0x105dfd:    mov     rsi, r12
0x105e00:    call   0x1083e0
0x105e05:    test    rax, rax
0x105e08:    je      0x105730
0x105e0e:    mov     rcx, qword ptr [rsp + 0x58]
0x105e13:    cmp     byte ptr [rcx], 0
0x105e16:    je      0x105730
0x105e1c:    mov     rdx, qword ptr [rip + 0x24a85]
0x105e23:    mov     r8d, 1
0x105e29:    mov     esi, 0x25
0x105e2e:    jmp     0x105b9b
0x105e33:    mov     rdx, qword ptr [rip + 0x24a6e]
0x105e3a:    mov     r8d, 1
0x105e40:    mov     esi, 0x21
0x105e45:    mov     rdi, qword ptr [rip + 0x24a64]
0x105e4c:    mov     rcx, qword ptr [rdx + 0x18]
0x105e50:    call   0x113e30
0x105e55:    mov     rdi, qword ptr [rip + 0x24a54]
0x105e5c:    call   0x111100
0x105e61:    mov     rdi, rax
0x105e64:    xor     eax, eax

```

```

0x105e66:    call    0x1089d0
0x105e6b:    mov     rdi, r14
0x105e6e:    call    0x10c7e0
0x105e73:    jmp     0x105b04
0x105e78:    mov     rdx, qword ptr [rip + 0x24a31]
0x105e7f:    mov     r8d, 1
0x105e85:    mov     esi, 0x21
0x105e8a:    mov     rdi, qword ptr [rip + 0x24a17]
0x105e91:    mov     rcx, qword ptr [rdx + 0x18]
0x105e95:    call    0x113e30
0x105e9a:    mov     rdi, qword ptr [rip + 0x24a07]
0x105ea1:    call    0x111100
0x105ea6:    mov     rdi, rax
0x105ea9:    xor     eax, eax
0x105eab:    call    0x1089d0
0x105eb0:    jmp     0x105730
0x105eb5:    mov     rax, qword ptr [rip + 0x249ec]
0x105ebc:    mov     rdi, qword ptr [rax + 0x78]
0x105ec0:    mov     r12, qword ptr [rax + 0x18]
0x105ec4:    add     rdi, 5
0x105ec8:    call    0x11d670
0x105ecd:    mov     rbp, rax
0x105ed0:    mov     rdi, qword ptr [rsp + 0x50]
0x105ed5:    call    0x1049b0
0x105eda:    mov     rdi, r14
0x105edd:    mov     r13, rax
0x105ee0:    call    0x10c7e0
0x105ee5:    test    r13, r13
0x105ee8:    je      0x105f1a
0x105ef1a:    mov     rdi, r12
0x105ef1d:    call    0x1049b0
0x105eea:    mov     r13, qword ptr [rsp + 0x50]
0x105eef:    xor     eax, eax
0x105ef1:    call    0x10ca00
0x105f22:    mov     rdi, r12
0x105f25:    mov     rsi, rax
0x105f28:    call    0x118530
0x105ef6:    cmp     byte ptr [r13 + rax - 1], 0x2f
0x105efc:    je      0x106f46
0x105f2d:    mov     r8, r12
0x105f30:    lea     rcx, [rip + 0x1a809]
0x105f37:    test    eax, eax
0x105f39:    jne     0x105f42
0x106f46:    mov     rdi, r12
0x106f49:    call    0x1049b0
0x105f02:    mov     r13, qword ptr [rsp + 0x50]
0x105f07:    xor     eax, eax
0x105f09:    call    0x10ca00

```

```

0x105f3b:    lea     rcx, [rip + 0x1a805]
0x105f42:    or      rdx, 0xfffffffffffffff
0x105f46:    mov     esi, 1
0x105f4b:    mov     rdi, rbp
0x105f4e:    xor     eax, eax
0x105f50:    call    0x104f10
0x105f42:    or      rdx, 0xfffffffffffffff
0x105f46:    mov     esi, 1
0x105f4b:    mov     rdi, rbp
0x105f4e:    xor     eax, eax
0x105f50:    call    0x104f10
0x106f4e:    mov     rdi, r12
0x106f51:    mov     rsi, rax
0x106f54:    call    0x118530
0x105f0e:    cmp     byte ptr [r13 + rax - 1], 0x20
0x105f14:    je      0x106f46
0x105f55:    mov     rdi, rbp
0x105f58:    call    0x1049b0
0x106f59:    test    eax, eax
0x106f5b:    je      0x1070a8
0x105f5d:    mov     rdi, rbp
0x105f60:    mov     rsi, rax
0x105f63:    call    0x1184d0
0x1070a8:    mov     r8, r12
0x1070ab:    lea     rcx, [rip + 0x19696]
0x1070b2:    jmp     0x105f42
0x106f61:    mov     r8, r12
0x106f64:    lea     rcx, [rip + 0x197d6]
0x106f6b:    jmp     0x105f42
0x105f68:    mov     rdi, rbp
0x105f6b:    call    0x10d610
0x105f70:    mov     rdi, rbp
0x105f73:    call    0x118d10
0x105f78:    xor     eax, eax
0x105f7a:    call    0x10c8c0
0x105f7f:    mov     r14, rax
0x105f82:    jmp     0x105730
0x105f87:    mov     rdi, r14
0x105f8a:    lea     r12, [rip + 0x1b488]
0x105f91:    lea     rbp, [rip + 0x1b2fc]
0x105f98:    call    0x10c7e0
0x105f9d:    mov     rdi, qword ptr [rip + 0x2490c]
0x105fa4:    call    0x1100c0
0x105fa9:    jmp     0x106029
0x106029:    mov     rdi, qword ptr [rip + 0x24880]
0x106030:    call    0x1100e0
0x106035:    cmp     eax, -1
0x106038:    jne     0x105fab

```

```

0x10603e:    lea     rdi, [rip + 0x1a700]
0x106045:    call   0x10d610
0x105fab:    mov     rdx, qword ptr [rip + 0x248fe]
0x105fb2:    cdqe
0x105fb4:    imul    rax, rax, 0x50
0x105fb8:    add     rax, qword ptr [rdx + 0x38]
0x105fbc:    mov     r14, qword ptr [rax]
0x105fbf:    mov     rdi, r14
0x105fc2:    call   0x1049b0
0x10604a:    xor     eax, eax
0x10604c:    call   0x10c8c0
0x105fc7:    lea     rdi, [rax + 5]
0x105fcb:    call   0x11d670
0x106051:    mov     r14, rax
0x106054:    jmp     0x105730
0x105fd0:    mov     rdi, r14
0x105fd3:    mov     r13, rax
0x105fd6:    call   0x1049b0
0x105fdb:    mov     rdi, r14
0x105fde:    mov     rsi, rax
0x105fe1:    call   0x118530
0x105fe6:    mov     r8, r14
0x105fe9:    mov     rcx, rbp
0x105fec:    test    eax, eax
0x105fee:    jne     0x105ff3
0x105ff0:    mov     rcx, r12
0x105ff3:    or      rdx, 0xffffffffffffffff
0x105ff7:    mov     esi, 1
0x105ffc:    mov     rdi, r13
0x105fff:    xor     eax, eax
0x106001:    call   0x104f10
0x105ff3:    or      rdx, 0xffffffffffffffff
0x105ff7:    mov     esi, 1
0x105ffc:    mov     rdi, r13
0x105fff:    xor     eax, eax
0x106001:    call   0x104f10
0x106006:    mov     rdi, r13
0x106009:    call   0x1049b0
0x10600e:    mov     rdi, r13
0x106011:    mov     rsi, rax
0x106014:    call   0x1184d0
0x106019:    mov     rdi, r13
0x10601c:    call   0x10d610
0x106021:    mov     rdi, r13
0x106024:    call   0x118d10
0x106059:    mov     rdi, r14
0x10605c:    call   0x10c7e0
0x106061:    mov     rdi, r15

```

```

0x106064:      call    0x10db10
0x106069:      mov     rbp, qword ptr [rsp + 0x50]
0x10606e:      mov     rdi, rbp
0x106071:      call    0x1049b0
0x106076:      sub     eax, 1
0x106079:      movsxd  rdx, eax
0x10607c:      jmp     0x106097
0x106097:      test    eax, eax
0x106099:      jns     0x10607e
0x10609b:      movzx   eax, byte ptr [rbp]
0x10609f:      cmp     al, 0x2b
0x1060a1:      je      0x1071c7
0x10607e:      lea     rcx, [rbp + rdx]
0x106083:      sub     rdx, 1
0x106087:      cmp     byte ptr [rcx], 0x20
0x10608a:      jne     0x10609b
0x1071c7:      cmp     byte ptr [rbp + 1], 0
0x1071cb:      mov     rdx, qword ptr [rip + 0x236d6]
0x1071d2:      mov     rdi, qword ptr [rip + 0x236d7]
0x1071d9:      jne     0x1072a7
0x1060a7:      jg      0x106edf
0x10608c:      mov     byte ptr [rcx], 0
0x10608f:      mov     rbp, qword ptr [rsp + 0x50]
0x106094:      sub     eax, 1
0x106097:      test    eax, eax
0x106099:      jns     0x10607e
0x1071df:      mov     r8d, 1
0x1071e5:      xor     ecx, ecx
0x1071e7:      mov     esi, 5
0x1071ec:      call    0x113e30
0x1072a7:      lea     rcx, [rbp + 1]
0x1072ab:      mov     r8d, 1
0x1072b1:      mov     esi, 0x24
0x1072b6:      call    0x113e30
0x1060ad:      test    al, al
0x1060af:      je      0x1071f6
0x106edf:      cmp     al, 0x2d
0x106ee1:      jne     0x1060c3
0x1071f1:      jmp     0x106f11
0x1072bb:      jmp     0x106f11
0x1071f6:      mov     rcx, qword ptr [rip + 0x236cb]
0x1071fd:      mov     rdx, qword ptr [rip + 0x236a4]
0x107204:      mov     esi, 1
0x107209:      mov     r8d, 1
0x10720f:      mov     rdi, qword ptr [rip + 0x2369a]
0x107216:      call    0x113e30
0x1060b5:      cmp     al, 0x2a
0x1060b7:      jne     0x1060c3

```



```

0x106ee7:    cmp     byte ptr [rbp + 1], 0
0x106eeb:    mov     rdx, qword ptr [rip + 0x239b6]
0x106ef2:    mov     rdi, qword ptr [rip + 0x239b7]
0x106ef9:    jne     0x10726e
0x1060c3:    xor     eax, eax
0x1060c5:    lea     rsi, [rsp + 0x48]
0x1060ca:    mov     rdi, rbp
0x1060cd:    call    0x11b5b0
0x106f11:    xor     edi, edi
0x106f13:    call    0x10d1a0
0x10721b:    mov     rdi, qword ptr [rip + 0x2368e]
0x107222:    mov     ebp, eax
0x107224:    call    0x111100
0x1060b9:    cmp     byte ptr [rbp + 1], 0
0x1060bd:    je      0x1072fd
0x106eff:    mov     r8d, 1
0x106f05:    xor     ecx, ecx
0x106f07:    mov     esi, 6
0x106f0c:    call    0x113e30
0x10726e:    lea     rcx, [rbp + 1]
0x107272:    mov     r8d, 1
0x107278:    mov     esi, 0x25
0x10727d:    call    0x113e30
0x1060d2:    test    eax, eax
0x1060d4:    js      0x107258
0x106f18:    jmp     0x105723
0x107229:    mov     rdi, rax
0x10722c:    call    0x10c4b0
0x1072fd:    mov     rdx, qword ptr [rip + 0x235a4]
0x107304:    mov     rdi, qword ptr [rip + 0x235a5]
0x10730b:    xor     ecx, ecx
0x10730d:    mov     esi, 7
0x107312:    mov     r8d, 1
0x107318:    call    0x113e30
0x107282:    jmp     0x106f11
0x107258:    mov     rdi, qword ptr [rsp + 0x48]
0x10725d:    mov     edx, 0x1c
0x107262:    xor     esi, esi
0x107264:    call    0x108070
0x1060da:    mov     rdi, qword ptr [rsp + 0x48]
0x1060df:    call    0x118620
0x107231:    xor     edi, edi
0x107233:    call    0x10d1a0
0x10731d:    xor     edi, edi
0x10731f:    call    0x10d1a0
0x107269:    jmp     0x105723
0x1060e4:    test    eax, eax
0x1060e6:    je      0x1072c0

```

```

0x107238:    xor     eax, eax
0x10723a:    call    0x109640
0x107324:    jmp     0x105723
0x1072c0:    mov     rdi, qword ptr [rsp + 0x48]
0x1072c5:    call    0x1185d0
0x1060ec:    xor     eax, eax
0x1060ee:    call    0x10c8c0
0x10723f:    xor     eax, eax
0x107241:    call    0x10c8c0
0x1072ca:    mov     ebp, eax
0x1072cc:    test    eax, eax
0x1072ce:    je      0x1073fa
0x1060f3:    mov     edx, 0x1c
0x1060f8:    xor     esi, esi
0x1060fa:    lea     rdi, [rip + 0x1a631]
0x106101:    call    0x108070
0x107246:    mov     r14, rax
0x107249:    test    ebp, ebp
0x10724b:    je      0x105730
0x1073fa:    mov     rdi, qword ptr [rsp + 0x48]
0x1073ff:    xor     ebp, ebp
0x107401:    call    0x10c4b0
0x1072d4:    xor     edi, edi
0x1072d6:    call    0x10d1a0
0x106106:    jmp     0x105723
0x107251:    xor     ebp, ebp
0x107253:    jmp     0x10565a
0x107406:    xor     edi, edi
0x107408:    call    0x10d1a0
0x1072db:    mov     rdi, qword ptr [rsp + 0x48]
0x1072e0:    call    0x10d610
0x10740d:    mov     rdi, qword ptr [rsp + 0x48]
0x107412:    call    0x10d610
0x1072e5:    mov     rdi, qword ptr [rsp + 0x48]
0x1072ea:    mov     esi, ebp
0x1072ec:    call    0x1193f0
0x107417:    mov     rdi, qword ptr [rsp + 0x48]
0x10741c:    xor     esi, esi
0x10741e:    call    0x1193f0
0x1072f1:    xor     edi, edi
0x1072f3:    call    0x10dbb0
0x107423:    xor     edi, edi
0x107425:    call    0x10dbb0
0x1072f8:    jmp     0x106f11
0x10742a:    xor     edi, edi
0x10742c:    call    0x10d1a0
0x107431:    mov     rdi, qword ptr [rip + 0x23478]
0x107438:    call    0x10f480

```

```

0x10743d:    mov     rdi, qword ptr [rip + 0x23464]
0x107444:    call   0x10f480
0x107449:    xor     eax, eax
0x10744b:    call   0x10a480
0x107450:    mov     rdi, qword ptr [rip + 0x23471]
0x107457:    call   0x10bb10
0x10745c:    xor     eax, eax
0x10745e:    mov     dword ptr [rip + 0x23474], 1
0x107468:    call   0x10c8c0
0x10746d:    mov     r14, rax
0x107470:    jmp     0x10565a
0x10610b:    lea     r12, [rsp + 0x58]
0x106110:    mov     rcx, qword ptr [r13 + 0x28]
0x106114:    xor     edx, edx
0x106116:    lea     rdi, [rip + 0x1a63b]
0x10611d:    mov     rsi, r12
0x106120:    call   0x1083e0
0x106125:    test    rax, rax
0x106128:    je      0x105730
0x10612e:    mov     rcx, qword ptr [rsp + 0x58]
0x106133:    cmp     byte ptr [rcx], 0
0x106136:    je      0x105730
0x10613c:    mov     rdx, qword ptr [rip + 0x24765]
0x106143:    mov     r8d, 1
0x106149:    mov     esi, 0x26
0x10614e:    jmp     0x105b9b
0x106153:    xor     ecx, ecx
0x106155:    xor     edx, edx
0x106157:    xor     esi, esi
0x106159:    xor     r13d, r13d
0x10615c:    lea     rdi, [rip + 0x1a603]
0x106163:    lea     rbp, [rsp + 0x60]
0x106168:    call   0x1086c0
0x10616d:    mov     rdx, qword ptr [rip + 0x24734]
0x106174:    mov     rdi, qword ptr [rip + 0x24735]
0x10617b:    xor     ecx, ecx
0x10617d:    mov     r8d, 1
0x106183:    mov     esi, 0x27
0x106188:    lea     r12, [rsp + 0x58]
0x10618d:    call   0x113e30
0x106192:    mov     dword ptr [rsp + 0xc], 0
0x10619a:    xor     edi, edi
0x10619c:    mov     rcx, rbp
0x10619f:    mov     edx, 1
0x1061a4:    mov     rsi, r12
0x1061a7:    call   0x1086c0
0x1061ac:    test    rax, rax
0x1061af:    je      0x106c72

```

```

0x106c72:    mov     rax, qword ptr [rsp]
0x106c76:    test    rax, rax
0x106c79:    je      0x106c83
0x1061b5:    mov     rdi, qword ptr [rsp + 0x58]
0x1061ba:    cmp     byte ptr [rdi], 0
0x1061bd:    jne     0x1061d1
0x106c83:    mov     rdi, qword ptr [rsp + 0x58]
0x106c88:    test    rdi, rdi
0x106c8b:    je      0x105730
0x106c7b:    mov     rdi, rax
0x106c7e:    call    0x118d10
0x1061bf:    mov     rax, qword ptr [rsp]
0x1061c3:    test    rax, rax
0x1061c6:    je      0x1061d1
0x1061d1:    cmp     dword ptr [rsp + 0x60], 2
0x1061d6:    je      0x106bf1
0x106c91:    call    0x104e70
0x1061c8:    cmp     byte ptr [rax], 0
0x1061cb:    jne     0x106b0b
0x106bf1:    cmp     dword ptr [rsp + 0xc], 0
0x106bf6:    je      0x1061dc
0x1061dc:    mov     qword ptr [rsp + 0x68], rdi
0x1061e1:    mov     rdx, qword ptr [rip + 0x246c0]
0x1061e8:    mov     r8d, 1
0x1061ee:    mov     rcx, rbp
0x1061f1:    mov     rdi, qword ptr [rip + 0x246b8]
0x1061f8:    mov     esi, 0x28
0x1061fd:    call    0x113e30
0x106c96:    mov     rdx, qword ptr [rip + 0x23c0b]
0x106c9d:    xor     ecx, ecx
0x106c9f:    mov     rdi, qword ptr [rip + 0x23c0a]
0x106ca6:    mov     r8d, 1
0x106cac:    mov     esi, 0x2a
0x106cb1:    mov     qword ptr [rsp], rax
0x106cb5:    call    0x113e30
0x106b0b:    call    0x118d10
0x106bfc:    mov     rdi, qword ptr [rip + 0x23cad]
0x106c03:    mov     esi, 1
0x106c08:    call    0x10f660
0x106202:    cmp     dword ptr [rsp + 0x60], -1
0x106207:    je      0x106d8e
0x106cba:    xor     ecx, ecx
0x106cbc:    mov     edx, 3
0x106cc1:    xor     esi, esi
0x106cc3:    xor     edi, edi
0x106cc5:    call    0x1086c0
0x106b10:    mov     rdi, qword ptr [rsp]
0x106b14:    call    0x104e70

```

```

0x106c0d:    mov     dword ptr [rsp + 0xc], 0
0x106c15:    mov     rdi, qword ptr [rsp + 0x58]
0x106c1a:    jmp     0x1061dc
0x106d8e:    xor     eax, eax
0x106d90:    call    0x10a5a0
0x10620d:    test    r13d, r13d
0x106210:    jne     0x106b64
0x106cca:    jmp     0x105730
0x106b19:    mov     rdi, rax
0x106b1c:    mov     qword ptr [rsp + 0x58], rax
0x106b21:    call    0x10d610
0x106d95:    mov     dword ptr [rsp + 0xc], 1
0x106d9d:    jmp     0x10619a
0x106216:    mov     rdi, qword ptr [rsp + 0x58]
0x10621b:    call    0x1049b0
0x106b64:    mov     rdi, qword ptr [rsp]
0x106b68:    mov     r13d, 1
0x106b6e:    call    0x1049b0
0x106b26:    mov     rax, qword ptr [rsp + 0x58]
0x106b2b:    mov     rcx, rbp
0x106b2e:    mov     rdx, qword ptr [rip + 0x23d73]
0x106b35:    mov     rdi, qword ptr [rip + 0x23d74]
0x106b3c:    mov     r8d, 1
0x106b42:    mov     esi, 0x28
0x106b47:    mov     qword ptr [rsp + 0x68], rax
0x106b4c:    call    0x113e30
0x10619a:    xor     edi, edi
0x10619c:    mov     rcx, rbp
0x10619f:    mov     edx, 1
0x1061a4:    mov     rsi, r12
0x1061a7:    call    0x1086c0
0x106220:    cmp     rax, qword ptr [rsp + 0x70]
0x106225:    jbe     0x10619a
0x106b73:    cmp     qword ptr [rsp + 0x70], rax
0x106b78:    jae     0x106216
0x106b51:    cmp     dword ptr [rsp + 0x60], -1
0x106b56:    je      0x106d88
0x10622b:    xor     eax, eax
0x10622d:    call    0x10ce70
0x106b7e:    xor     edi, edi
0x106b80:    xor     r13d, r13d
0x106b83:    call    0x10d1a0
0x106d88:    mov     r13d, 1
0x106d8e:    xor     eax, eax
0x106d90:    call    0x10a5a0
0x106b5c:    mov     dword ptr [rsp + 0xc], 0
0x106b64:    mov     rdi, qword ptr [rsp]
0x106b68:    mov     r13d, 1

```

```

0x106b6e:    call    0x1049b0
0x106232:    xor     eax, eax
0x106234:    call    0x10d870
0x106b88:    mov     rdi, qword ptr [rsp + 0x58]
0x106b8d:    call    0x1049b0
0x106239:    xor     eax, eax
0x10623b:    call    0x10d740
0x106b92:    cmp     rax, qword ptr [rsp + 0x70]
0x106b97:    jbe     0x106232
0x106240:    xor     eax, eax
0x106242:    call    0x10a820
0x106b9d:    jmp     0x10622b
0x106247:    jmp     0x10619a
0x10624c:    xor     ecx, ecx
0x10624e:    xor     edx, edx
0x106250:    xor     esi, esi
0x106252:    xor     r13d, r13d
0x106255:    lea     rdi, [rip + 0x1a51e]
0x10625c:    lea     rbp, [rsp + 0x60]
0x106261:    call    0x1086c0
0x106266:    mov     rdx, qword ptr [rip + 0x2463b]
0x10626d:    mov     rdi, qword ptr [rip + 0x2463c]
0x106274:    xor     ecx, ecx
0x106276:    mov     r8d, 1
0x10627c:    mov     esi, 0x27
0x106281:    lea     r12, [rsp + 0x58]
0x106286:    call    0x113e30
0x10628b:    mov     dword ptr [rsp + 0xc], 0
0x106293:    xor     edi, edi
0x106295:    mov     rcx, rbp
0x106298:    mov     edx, 2
0x10629d:    mov     rsi, r12
0x1062a0:    call    0x1086c0
0x1062a5:    test    rax, rax
0x1062a8:    je      0x106c72
0x1062ae:    mov     rdi, qword ptr [rsp + 0x58]
0x1062b3:    cmp     byte ptr [rdi], 0
0x1062b6:    jne     0x1062ca
0x1062b8:    mov     rax, qword ptr [rsp]
0x1062bc:    test    rax, rax
0x1062bf:    je      0x1062ca
0x1062ca:    cmp     dword ptr [rsp + 0x60], 2
0x1062cf:    je      0x106bbc
0x1062c1:    cmp     byte ptr [rax], 0
0x1062c4:    jne     0x106a74
0x106bbc:    cmp     dword ptr [rsp + 0xc], 0
0x106bc1:    je      0x1062d5
0x1062d5:    mov     qword ptr [rsp + 0x68], rdi

```

```

0x1062da:    mov     rdx, qword ptr [rip + 0x245c7]
0x1062e1:    mov     r8d, 1
0x1062e7:    mov     rcx, rbp
0x1062ea:    mov     rdi, qword ptr [rip + 0x245bf]
0x1062f1:    mov     esi, 0x29
0x1062f6:    call    0x113e30
0x106a74:    call    0x118d10
0x106bc7:    xor     eax, eax
0x106bc9:    call    0x10a5a0
0x1062fb:    cmp     dword ptr [rsp + 0x60], -1
0x106300:    je      0x106da8
0x106a79:    mov     rdi, qword ptr [rsp]
0x106a7d:    call    0x104e70
0x106bce:    mov     rdi, qword ptr [rip + 0x23cdb]
0x106bd5:    mov     esi, 1
0x106bda:    call    0x10f660
0x106da8:    xor     eax, eax
0x106daa:    call    0x10a5a0
0x106306:    test    r13d, r13d
0x106309:    jne     0x106acd
0x106a82:    mov     rdi, rax
0x106a85:    mov     qword ptr [rsp + 0x58], rax
0x106a8a:    call    0x10d610
0x106bdf:    mov     dword ptr [rsp + 0xc], 0
0x106be7:    mov     rdi, qword ptr [rsp + 0x58]
0x106bec:    jmp     0x1062d5
0x106daf:    mov     dword ptr [rsp + 0xc], 1
0x106db7:    jmp     0x106293
0x10630f:    mov     rdi, qword ptr [rsp + 0x58]
0x106314:    call    0x1049b0
0x106acd:    mov     rdi, qword ptr [rsp]
0x106ad1:    mov     r13d, 1
0x106ad7:    call    0x1049b0
0x106a8f:    mov     rax, qword ptr [rsp + 0x58]
0x106a94:    mov     rcx, rbp
0x106a97:    mov     rdx, qword ptr [rip + 0x23e0a]
0x106a9e:    mov     rdi, qword ptr [rip + 0x23e0b]
0x106aa5:    mov     r8d, 1
0x106aab:    mov     esi, 0x29
0x106ab0:    mov     qword ptr [rsp + 0x68], rax
0x106ab5:    call    0x113e30
0x106293:    xor     edi, edi
0x106295:    mov     rcx, rbp
0x106298:    mov     edx, 2
0x10629d:    mov     rsi, r12
0x1062a0:    call    0x1086c0
0x106319:    cmp     rax, qword ptr [rsp + 0x70]
0x10631e:    jbe     0x106293

```

```

0x106adc:    cmp     qword ptr [rsp + 0x70], rax
0x106ae1:    jae     0x10630f
0x106aba:    cmp     dword ptr [rsp + 0x60], -1
0x106abf:    je      0x106da2
0x106324:    xor     eax, eax
0x106326:    call    0x10ce70
0x106ae7:    xor     edi, edi
0x106ae9:    xor     r13d, r13d
0x106aec:    call    0x10d1a0
0x106da2:    mov     r13d, 1
0x106da8:    xor     eax, eax
0x106daa:    call    0x10a5a0
0x106ac5:    mov     dword ptr [rsp + 0xc], 0
0x106acd:    mov     rdi, qword ptr [rsp]
0x106ad1:    mov     r13d, 1
0x106ad7:    call    0x1049b0
0x10632b:    xor     eax, eax
0x10632d:    call    0x10d870
0x106af1:    mov     rdi, qword ptr [rsp + 0x58]
0x106af6:    call    0x1049b0
0x106332:    xor     eax, eax
0x106334:    call    0x10d740
0x106afb:    cmp     rax, qword ptr [rsp + 0x70]
0x106b00:    jbe     0x10632b
0x106339:    xor     eax, eax
0x10633b:    call    0x10a820
0x106b06:    jmp     0x106324
0x106340:    jmp     0x106293
0x106345:    mov     rsi, qword ptr [rip + 0x2455c]
0x10634c:    mov     rdi, qword ptr [rip + 0x2455d]
0x106353:    xor     eax, eax
0x106355:    call    0x108a80
0x10635a:    jmp     0x105afc
0x10635f:    mov     rsi, qword ptr [rip + 0x24542]
0x106366:    mov     rdi, qword ptr [rip + 0x24543]
0x10636d:    xor     eax, eax
0x10636f:    call    0x108a30
0x106374:    jmp     0x105afc
0x106379:    mov     rdi, qword ptr [rip + 0x24530]
0x106380:    call    0x10f490
0x106385:    mov     rdi, rax
0x106388:    mov     r12, rax
0x10638b:    call    0x1049b0
0x106390:    lea     rdi, [rax + 5]
0x106394:    call    0x11d670
0x106399:    mov     rbp, rax
0x10639c:    jmp     0x105ed0
0x105ed0:    mov     rdi, qword ptr [rsp + 0x50]

```



```

0x105ed5:      call    0x1049b0
0x1063a1:      mov     rdx, qword ptr [rip + 0x24500]
0x1063a8:      mov     rdi, qword ptr [rip + 0x24501]
0x1063af:      xor     ecx, ecx
0x1063b1:      mov     esi, 0x1f
0x1063b6:      mov     r8d, 1
0x1063bc:      call    0x113e30
0x1063c1:      jmp     0x105730
0x1063c6:      mov     rdx, qword ptr [rip + 0x244db]
0x1063cd:      mov     rdi, qword ptr [rip + 0x244dc]
0x1063d4:      xor     ecx, ecx
0x1063d6:      mov     esi, 0x20
0x1063db:      mov     r8d, 1
0x1063e1:      call    0x113e30
0x1063e6:      jmp     0x105730
0x1063eb:      mov     rdx, qword ptr [rip + 0x244b6]
0x1063f2:      mov     rdi, qword ptr [rip + 0x244b7]
0x1063f9:      xor     ecx, ecx
0x1063fb:      mov     esi, 0x19
0x106400:      mov     r8d, 1
0x106406:      call    0x113e30
0x10640b:      jmp     0x105730
0x106410:      mov     rdx, qword ptr [rip + 0x24491]
0x106417:      mov     rdi, qword ptr [rip + 0x24492]
0x10641e:      xor     ecx, ecx
0x106420:      mov     esi, 3
0x106425:      mov     r8d, 1
0x10642b:      call    0x113e30
0x106430:      jmp     0x105730
0x106435:      cmp     dword ptr [rip + 0x244cc], 0
0x10643c:      jne     0x10701d
0x106442:      mov     r15d, dword ptr [rsp + 0x10]
0x106447:      mov     rdi, qword ptr [rip + 0x24462]
0x10644e:      call    0x111100
0x10701d:      mov     rdi, qword ptr [rip + 0x238ac]
0x107024:      xor     edx, edx
0x107026:      lea     rsi, [rip + 0x19505]
0x10702d:      call    0x108070
0x106453:      mov     rdi, qword ptr [rip + 0x24466]
0x10645a:      mov     rbp, rax
0x10645d:      call    0x10f3e0
0x107032:      cmp     al, 0x79
0x107034:      jne     0x105730
0x106462:      mov     rdi, qword ptr [rip + 0x2444f]
0x106469:      call    0x10f3e0
0x10703a:      jmp     0x106442
0x10646e:      xor     edi, edi
0x106470:      call    0x109b00

```

```

0x106475:    xor     eax, eax
0x106477:    call   0x10a6c0
0x10647c:    cmp     qword ptr [rip + 0x24f84], 0
0x106484:    je      0x10648d
0x10648d:    xor     eax, eax
0x10648f:    call   0x10de40
0x106486:    xor     eax, eax
0x106488:    call   0x10c820
0x106494:    xor     eax, eax
0x106496:    call   0x119580
0x10649b:    mov     rdi, qword ptr [rip + 0x24426]
0x1064a2:    call   0x10bbc0
0x1064a7:    test    r15d, r15d
0x1064aa:    je      0x10507e
0x1064b0:    mov     rdi, rbp
0x1064b3:    xor     r15d, r15d
0x1064b6:    call   0x1049b0
0x1064bb:    mov     rsi, rbp
0x1064be:    mov     edi, 3
0x1064c3:    mov     rdx, rax
0x1064c6:    call   0x104790
0x1064cb:    jmp     0x10507e
0x1064d0:    mov     rdi, r14
0x1064d3:    xor     ebp, ebp
0x1064d5:    call   0x10c7e0
0x1064da:    jmp     0x1064f0
0x1064f0:    cmp     dword ptr [rsp + 0x28], ebp
0x1064f4:    jg      0x1064dc
0x1064f6:    jmp     0x105723
0x1064dc:    mov     edi, 1
0x1064e1:    add     ebp, 1
0x1064e4:    call   0x10dbb0
0x1064e9:    xor     eax, eax
0x1064eb:    call   0x10a820
0x1064fb:    test    byte ptr [rsp + 0x28], 1
0x106500:    je      0x105730
0x106506:    xor     edi, edi
0x106508:    lea     rbp, [rip + 0x19f49]
0x10650f:    lea     r13, [rip + 0x22b0a]
0x106516:    call   0x104aa0
0x10651b:    mov     rdi, qword ptr [rip + 0x243a6]
0x106522:    call   0x10bb40
0x106527:    mov     rdi, qword ptr [rip + 0x2373a]
0x10652e:    xor     edx, edx
0x106530:    xor     esi, esi
0x106532:    call   0x10e480
0x106537:    xor     eax, eax
0x106539:    call   0x10a820

```

```

0x10653e:    mov     rdi, r14
0x106541:    call   0x10c970
0x106546:    xor     eax, eax
0x106548:    call   0x10c8c0
0x10654d:    mov     r14, rax
0x106550:    xor     eax, eax
0x106552:    call   0x10d870
0x106557:    xor     eax, eax
0x106559:    call   0x10d740
0x10655e:    xor     eax, eax
0x106560:    call   0x10a820
0x106565:    mov     rdi, r15
0x106568:    call   0x10db10
0x10656d:    mov     dword ptr [rip + 0x2436d], 1
0x106577:    jmp     0x10658e
0x10658e:    mov     rdi, rbx
0x106591:    call   0x10b180
0x106596:    test    rax, rax
0x106599:    je      0x106580
0x106580:    mov     rdi, qword ptr [rip + 0x236e1]
0x106587:    xor     eax, eax
0x106589:    call   0x107d10
0x10659b:    mov     rdx, qword ptr [rax]
0x10659e:    mov     rax, qword ptr [rax + 0x10]
0x1065a2:    movzx   r12d, byte ptr [rdx]
0x1065a6:    test    rax, rax
0x1065a9:    je      0x10696f
0x10696f:    lea     eax, [r12 + 0x22]
0x106974:    cmp     eax, 0x29
0x106977:    ja      0x1065e3
0x1065af:    cmp     byte ptr [rax + 0x1b], 0
0x1065b3:    je      0x10696f
0x10697d:    movsxd  rax, dword ptr [rbp + rax*4]
0x106982:    add     rax, rbp
0x106985:    jmp     rax
0x1065e3:    mov     rdi, r14
0x1065e6:    call   0x10c7e0
0x1065b9:    mov     rax, qword ptr [rax]
0x1065bc:    mov     esi, 0x23
0x1065c1:    sub     rax, r13
0x1065c4:    cqo
0x1065c6:    idiv    rsi
0x1065c9:    mov     r12d, eax
0x1065cc:    not     r12d
0x1065cf:    cmp     eax, 6
0x1065d2:    jne     0x106e1f
0x106988:    test    r12d, r12d
0x10698b:    je      0x106611

```

```

0x1069ce:    mov     rdi, r14
0x1069d1:    xor     r12d, r12d
0x1069d4:    call    0x10c7e0
0x106996:    mov     rdi, r14
0x106999:    xor     r12d, r12d
0x10699c:    call    0x10c7e0
0x106a56:    mov     rdi, r14
0x106a59:    call    0x10c7e0
0x1069f9:    cmp     dword ptr [rip + 0x23f08], 0
0x106a00:    je      0x107329
0x106a3f:    xor     edi, edi
0x106a41:    xor     eax, eax
0x106a43:    call    0x107bd0
0x1065eb:    jmp     0x1065fa
0x1065d8:    test    byte ptr [rsp + 0x28], 1
0x1065dd:    jne     0x107383
0x106e1f:    cmp     r12d, -0x3d
0x106e23:    jne     0x10696f
0x106611:    xor     eax, eax
0x106613:    call    0x10dea0
0x106991:    jmp     0x1065e3
0x1069d9:    jmp     0x1069f0
0x1069a1:    jmp     0x1069b8
0x106a5e:    xor     edi, edi
0x106a60:    call    0x10d1a0
0x107329:    mov     rdi, qword ptr [rip + 0x23580]
0x107330:    mov     r15d, dword ptr [rsp + 0x10]
0x107335:    call    0x10f480
0x106a06:    mov     rdi, qword ptr [rip + 0x23ec3]
0x106a0d:    xor     edx, edx
0x106a0f:    lea     rsi, [rip + 0x19b1c]
0x106a16:    call    0x108070
0x106a48:    mov     rdi, qword ptr [rip + 0x23e79]
0x106a4f:    call    0x10bb40
0x1065fa:    mov     eax, dword ptr [rsp + 0x28]
0x1065fe:    lea     edx, [rax - 1]
0x106601:    mov     dword ptr [rsp + 0x28], edx
0x106605:    test    eax, eax
0x106607:    jne     0x1065ed
0x107383:    mov     rdi, r14
0x107386:    xor     ebp, ebp
0x107388:    call    0x10c7e0
0x106e29:    mov     rax, qword ptr [rsp + 0x50]
0x106e2e:    cmp     byte ptr [rax], 0
0x106e31:    je      0x106611
0x106618:    jmp     0x10653e
0x1069f0:    cmp     dword ptr [rsp + 0x28], r12d
0x1069f5:    jg      0x1069db

```

```

0x1069b8:    cmp     dword ptr [rsp + 0x28], r12d
0x1069bd:    jg      0x1069a3
0x106a65:    xor     eax, eax
0x106a67:    call    0x10c8c0
0x10733a:    mov     rdi, qword ptr [rip + 0x23567]
0x107341:    call    0x10f480
0x106a1b:    cmp     al, 0x79
0x106a1d:    je      0x107329
0x106a54:    jmp     0x106a23
0x106609:    call    0x10c8c0
0x1065ed:    mov     esi, 3
0x1065f2:    mov     edi, r12d
0x1065f5:    call    0x107570
0x10738d:    xor     eax, eax
0x10738f:    call    0x10c8c0
0x106e37:    mov     rdi, r14
0x106e3a:    call    0x10c7e0
0x1069f7:    jmp     0x1069bf
0x1069db:    mov     edi, 2
0x1069e0:    add     r12d, 1
0x1069e4:    call    0x10dbb0
0x1069bf:    xor     eax, eax
0x1069c1:    call    0x10c8c0
0x1069a3:    mov     edi, 1
0x1069a8:    add     r12d, 1
0x1069ac:    call    0x10dbb0
0x106a6c:    mov     r14, rax
0x106a6f:    jmp     0x106611
0x107346:    xor     eax, eax
0x107348:    call    0x10a480
0x106a23:    mov     rdi, qword ptr [rip + 0x2323e]
0x106a2a:    xor     edx, edx
0x106a2c:    xor     esi, esi
0x106a2e:    call    0x10e480
0x10660e:    mov     r14, rax
0x106611:    xor     eax, eax
0x106613:    call    0x10dea0
0x107394:    mov     rdi, qword ptr [rip + 0x23515]
0x10739b:    mov     r14, rax
0x10739e:    call    0x10f480
0x106e3f:    mov     rdi, qword ptr [rsp + 0x50]
0x106e44:    xor     eax, eax
0x106e46:    lea     rsi, [rsp + 0x48]
0x106e4b:    call    0x11b5b0
0x1069e9:    xor     eax, eax
0x1069eb:    call    0x10a820
0x1069c6:    mov     r14, rax
0x1069c9:    jmp     0x106611

```

```

0x1069b1:    xor     eax, eax
0x1069b3:    call   0x10a820
0x10734d:    xor     eax, eax
0x10734f:    call   0x10e4e0
0x106a33:    xor     eax, eax
0x106a35:    call   0x10a820
0x1073a3:    mov     rdi, qword ptr [rip + 0x234fe]
0x1073aa:    call   0x10f480
0x106e50:    test    eax, eax
0x106e52:    js      0x107287
0x107354:    xor     eax, eax
0x107356:    call   0x10a820
0x106a3a:    jmp     0x106611
0x1073af:    xor     eax, eax
0x1073b1:    call   0x10a480
0x107287:    mov     rdi, qword ptr [rsp + 0x48]
0x10728c:    mov     edx, 0x1c
0x107291:    xor     esi, esi
0x107293:    call   0x108070
0x106e58:    mov     rdi, qword ptr [rip + 0x23a69]
0x106e5f:    call   0x10bb40
0x10735b:    xor     eax, eax
0x10735d:    call   0x118330
0x1073b6:    xor     eax, eax
0x1073b8:    call   0x10e4e0
0x107298:    xor     eax, eax
0x10729a:    call   0x10c8c0
0x106e64:    xor     edi, edi
0x106e66:    call   0x10d1a0
0x107362:    mov     r8, rax
0x107365:    mov     eax, 0x3c
0x10736a:    sub     eax, dword ptr [r8]
0x10736d:    mov     edi, eax
0x10736f:    call   0x104aa0
0x1073bd:    xor     eax, eax
0x1073bf:    call   0x10a820
0x10729f:    mov     r14, rax
0x1072a2:    jmp     0x106611
0x106e6b:    mov     rdi, qword ptr [rsp + 0x48]
0x106e70:    call   0x10d610
0x107374:    mov     dword ptr [rip + 0x23566], 0
0x10737e:    jmp     0x106447
0x1073c4:    xor     eax, eax
0x1073c6:    call   0x118330
0x106e75:    mov     rdi, qword ptr [rsp + 0x48]
0x106e7a:    call   0x10c4b0
0x106447:    mov     rdi, qword ptr [rip + 0x24462]
0x10644e:    call   0x111100

```

```

0x1073cb:    mov     r8, rax
0x1073ce:    mov     eax, 0x3c
0x1073d3:    sub     eax, dword ptr [r8]
0x1073d6:    mov     edi, eax
0x1073d8:    call    0x104aa0
0x106e7f:    mov     rdi, qword ptr [rsp + 0x48]
0x106e84:    xor     esi, esi
0x106e86:    call    0x1193f0
0x1073dd:    mov     dword ptr [rip + 0x234fd], 0
0x1073e7:    jmp     0x10565a
0x106e8b:    mov     rdi, qword ptr [rip + 0x23a36]
0x106e92:    call    0x10bb10
0x106e97:    xor     edi, edi
0x106e99:    call    0x10dbb0
0x106e9e:    mov     rdi, qword ptr [rip + 0x22dc3]
0x106ea5:    xor     edx, edx
0x106ea7:    xor     esi, esi
0x106ea9:    call    0x10e480
0x106eae:    xor     edi, edi
0x106eb0:    call    0x10d1a0
0x106eb5:    xor     eax, eax
0x106eb7:    call    0x10c8c0
0x106ebc:    mov     rdi, qword ptr [rip + 0x239ed]
0x106ec3:    mov     r14, rax
0x106ec6:    call    0x111100
0x106ecb:    mov     rdi, rax
0x106ece:    call    0x10c4b0
0x106ed3:    xor     eax, eax
0x106ed5:    call    0x10a820
0x106eda:    jmp     0x106611
0x106579:    nop     dword ptr [rax]
0x10661d:    xor     eax, eax
0x10661f:    call    0x1096c0
0x106624:    mov     rdi, qword ptr [rip + 0x24285]
0x10662b:    call    0x111100
0x106630:    mov     rdi, rax
0x106633:    call    0x10c4b0
0x106638:    xor     edi, edi
0x10663a:    xor     eax, eax
0x10663c:    call    0x107bd0
0x106641:    jmp     0x105730
0x106646:    test    byte ptr [rsp + 0x28], 1
0x10664b:    je      0x105730
0x106651:    cmp     dword ptr [rip + 0x23600], 0
0x106658:    je      0x105730
0x10665e:    mov     rdi, qword ptr [rip + 0x2424b]
0x106665:    call    0x10f480
0x10666a:    mov     rdi, qword ptr [rip + 0x24237]

```

```

0x106671:      call    0x10f480
0x106676:      mov     rdx, qword ptr [rip + 0x2422b]
0x10667d:      mov     rdi, qword ptr [rip + 0x2422c]
0x106684:      xor     ecx, ecx
0x106686:      mov     r8d, 1
0x10668c:      mov     esi, 0x23
0x106691:      jmp     0x1059ff
0x1059ff:      call    0x113e30
0x106696:      mov     rdi, r14
0x106699:      xor     ebp, ebp
0x10669b:      call    0x10c7e0
0x1066a0:      jmp     0x1066b6
0x1066b6:      cmp     dword ptr [rsp + 0x28], ebp
0x1066ba:      jg      0x1066a2
0x1066bc:      jmp     0x105723
0x1066a2:      mov     edi, 2
0x1066a7:      add     ebp, 1
0x1066aa:      call    0x10dbb0
0x1066af:      xor     eax, eax
0x1066b1:      call    0x10a820
0x1066df:      mov     rdx, qword ptr [rip + 0x241c2]
0x1066e6:      mov     rdi, qword ptr [rip + 0x241c3]
0x1066ed:      xor     ecx, ecx
0x1066ef:      mov     esi, 2
0x1066f4:      mov     r8d, 1
0x1066fa:      call    0x113e30
0x1066ff:      jmp     0x105730
0x106704:      mov     rdx, qword ptr [rip + 0x2419d]
0x10670b:      mov     rdi, qword ptr [rip + 0x2419e]
0x106712:      xor     ecx, ecx
0x106714:      mov     esi, 0x1a
0x106719:      mov     r8d, 1
0x10671f:      call    0x113e30
0x106724:      jmp     0x105730
0x106729:      xor     ebp, ebp
0x10672b:      jmp     0x106750
0x106750:      cmp     dword ptr [rsp + 0x28], ebp
0x106754:      jg      0x10672d
0x106756:      jmp     0x105730
0x10672d:      mov     r8d, 1
0x106733:      xor     ecx, ecx
0x106735:      mov     esi, 4
0x10673a:      add     ebp, 1
0x10673d:      mov     rdx, qword ptr [rip + 0x24164]
0x106744:      mov     rdi, qword ptr [rip + 0x24165]
0x10674b:      call    0x113e30
0x10675b:      xor     ebp, ebp
0x10675d:      jmp     0x106782

```



```

0x106782:    cmp     dword ptr [rsp + 0x28], ebp
0x106786:    jg      0x10675f
0x106788:    jmp     0x105730
0x10675f:    mov     r8d, 1
0x106765:    xor     ecx, ecx
0x106767:    mov     esi, 0x1d
0x10676c:    add     ebp, 1
0x10676f:    mov     rdx, qword ptr [rip + 0x24132]
0x106776:    mov     rdi, qword ptr [rip + 0x24133]
0x10677d:    call    0x113e30
0x10678d:    xor     ebp, ebp
0x10678f:    jmp     0x1067b4
0x1067b4:    cmp     dword ptr [rsp + 0x28], ebp
0x1067b8:    jg      0x106791
0x1067ba:    jmp     0x105730
0x106791:    mov     r8d, 1
0x106797:    xor     ecx, ecx
0x106799:    mov     esi, 0x1e
0x10679e:    add     ebp, 1
0x1067a1:    mov     rdx, qword ptr [rip + 0x24100]
0x1067a8:    mov     rdi, qword ptr [rip + 0x24101]
0x1067af:    call    0x113e30
0x1067bf:    nop
0x10747a:    nop     word ptr [rax + rax]
0x107480:    endbr64
0x107484:    xor     ebp, ebp
0x107486:    mov     r9, rdx
0x107489:    pop     rsi
0x10748a:    mov     rdx, rsp
0x10748d:    and     rsp, 0xfffffffffffffff0
0x107491:    push    rax
0x107492:    push    rsp
0x107493:    xor     r8d, r8d
0x107496:    xor     ecx, ecx
0x107498:    lea     rdi, [rip - 0x256f]
0x10749f:    call    qword ptr [rip + 0x21b23]
0x1074a5:    hlt
0x1074a6:    nop     word ptr cs:[rax + rax]
0x1074b0:    lea     rdi, [rip + 0x23379]
0x1074b7:    lea     rax, [rip + 0x23372]
0x1074be:    cmp     rax, rdi
0x1074c1:    je      0x1074d8
0x1074d8:    ret
0x1074c3:    mov     rax, qword ptr [rip + 0x21b06]
0x1074ca:    test    rax, rax
0x1074cd:    je      0x1074d8
0x1074cf:    jmp     rax
0x1074d1:    nop     dword ptr [rax]

```

```

0x1074d9:    nop    dword ptr [rax]
0x1074e0:    lea    rdi, [rip + 0x23349]
0x1074e7:    lea    rsi, [rip + 0x23342]
0x1074ee:    sub    rsi, rdi
0x1074f1:    mov    rax, rsi
0x1074f4:    shr    rsi, 0x3f
0x1074f8:    sar    rax, 3
0x1074fc:    add    rsi, rax
0x1074ff:    sar    rsi, 1
0x107502:    je     0x107518
0x107518:    ret
0x107504:    mov    rax, qword ptr [rip + 0x21ae5]
0x10750b:    test   rax, rax
0x10750e:    je     0x107518
0x107510:    jmp    rax
0x107512:    nop    word ptr [rax + rax]
0x107519:    nop    dword ptr [rax]
0x107520:    endbr64
0x107524:    cmp    byte ptr [rip + 0x2335d], 0
0x10752b:    jne    0x107558
0x10752d:    push   rbp
0x10752e:    cmp    qword ptr [rip + 0x21ac2], 0
0x107536:    mov    rbp, rsp
0x107539:    je     0x107547
0x107558:    ret
0x107547:    call   0x1074b0
0x10753b:    mov    rdi, qword ptr [rip + 0x21ac6]
0x107542:    call   0x1047b0
0x10754c:    mov    byte ptr [rip + 0x23335], 1
0x107553:    pop    rbp
0x107554:    ret
0x107555:    nop    dword ptr [rax]
0x107559:    nop    dword ptr [rax]
0x107560:    endbr64
0x107564:    jmp    0x1074e0
0x107569:    nop    dword ptr [rax]
0x107570:    push   r12
0x107572:    lea    eax, [rdi + 0x42]
0x107575:    mov    r12d, esi
0x107578:    sub    rsp, 0x10
0x10757c:    cmp    eax, 0x17
0x10757f:    ja     0x1077e0
0x107585:    lea    rdx, [rip + 0x18a78]
0x10758c:    movsxd rax, dword ptr [rdx + rax*4]
0x107590:    add    rax, rdx
0x107593:    jmp    rax
0x1077e0:    mov    dword ptr [rsp + 0xc], edi
0x1077e4:    and    r12d, 1

```

```

0x1077e8:    je      0x1075ad
0x1075ad:    add     rsp, 0x10
0x1075b1:    mov     eax, r12d
0x1075b4:    pop     r12
0x1075b6:    ret
0x1077ee:    call    0x104f00
0x1077f3:    mov     r8, rax
0x1077f6:    movsxd  rax, dword ptr [rsp + 0xc]
0x1077fb:    mov     rdx, qword ptr [r8]
0x1077fe:    mov     rdi, rax
0x107801:    test    byte ptr [rdx + rax*2 + 1], 0x40
0x107806:    je      0x107818
0x107818:    xor     r12d, r12d
0x10781b:    jmp     0x1075ad
0x107808:    call    0x10cd30
0x10780d:    jmp     0x1075ad
0x107596:    nop     word ptr cs:[rax + rax]
0x1075a0:    xor     eax, eax
0x1075a2:    mov     r12d, 1
0x1075a8:    call    0x10cb00
0x1075b7:    nop     word ptr [rax + rax]
0x1075c0:    xor     eax, eax
0x1075c2:    mov     r12d, 1
0x1075c8:    call    0x10cb30
0x1075cd:    jmp     0x1075ad
0x1075cf:    nop
0x1075d0:    xor     eax, eax
0x1075d2:    mov     r12d, 1
0x1075d8:    call    0x10cb60
0x1075dd:    jmp     0x1075ad
0x1075df:    nop
0x1075e0:    xor     eax, eax
0x1075e2:    mov     r12d, 1
0x1075e8:    call    0x10cc30
0x1075ed:    jmp     0x1075ad
0x1075ef:    nop
0x1075f0:    xor     eax, eax
0x1075f2:    mov     r12d, 1
0x1075f8:    call    0x10ccf0
0x1075fd:    jmp     0x1075ad
0x1075ff:    nop
0x107600:    xor     eax, eax
0x107602:    mov     r12d, 1
0x107608:    call    0x10cd10
0x10760d:    jmp     0x1075ad
0x10760f:    nop
0x107610:    and     r12d, 1
0x107614:    jne     0x107820

```

```

0x10761a:    mov     r12d, 1
0x107620:    add     rsp, 0x10
0x107624:    mov     eax, r12d
0x107627:    pop     r12
0x107629:    ret
0x107820:    xor     eax, eax
0x107822:    call    0x10cde0
0x107827:    jmp     0x1075ad
0x10762a:    nop     word ptr [rax + rax]
0x107630:    and     r12d, 1
0x107634:    je      0x10761a
0x107636:    xor     eax, eax
0x107638:    call    0x10ce70
0x10763d:    jmp     0x1075ad
0x107642:    nop     word ptr [rax + rax]
0x107648:    and     r12d, 1
0x10764c:    je      0x10761a
0x10764e:    xor     eax, eax
0x107650:    call    0x10d380
0x107655:    jmp     0x1075ad
0x10765a:    nop     word ptr [rax + rax]
0x107660:    and     r12d, 1
0x107664:    je      0x10761a
0x107666:    xor     eax, eax
0x107668:    call    0x10d400
0x10766d:    jmp     0x1075ad
0x107672:    nop     word ptr [rax + rax]
0x107678:    and     r12d, 1
0x10767c:    je      0x10761a
0x10767e:    xor     eax, eax
0x107680:    call    0x10d480
0x107685:    jmp     0x1075ad
0x10768a:    nop     word ptr [rax + rax]
0x107690:    xor     eax, eax
0x107692:    mov     r12d, 1
0x107698:    call    0x10ca60
0x10769d:    jmp     0x1075ad
0x1076a2:    nop     word ptr [rax + rax]
0x1076a8:    and     r12d, 1
0x1076ac:    je      0x10761a
0x1076b2:    xor     eax, eax
0x1076b4:    call    0x10ca80
0x1076b9:    jmp     0x1075ad
0x1076be:    nop
0x1076c0:    xor     eax, eax
0x1076c2:    mov     r12d, 1
0x1076c8:    call    0x10cab0
0x1076cd:    jmp     0x1075ad

```

```

0x1076d2:      nop      word ptr [rax + rax]
0x1076d8:      and      r12d, 1
0x1076dc:      je       0x10761a
0x1076e2:      xor      eax, eax
0x1076e4:      call     0x10d700
0x1076e9:      jmp      0x1075ad
0x1076ee:      nop
0x1076f0:      xor      eax, eax
0x1076f2:      mov      r12d, 1
0x1076f8:      call     0x10cae0
0x1076fd:      jmp      0x1075ad
0x107702:      nop      word ptr [rax + rax]
0x107708:      and      r12d, 1
0x10770c:      je       0x10761a
0x107712:      xor      eax, eax
0x107714:      call     0x10d1f0
0x107719:      jmp      0x1075ad
0x10771e:      nop
0x107720:      and      r12d, 1
0x107724:      je       0x10761a
0x10772a:      xor      eax, eax
0x10772c:      call     0x10d250
0x107731:      jmp      0x1075ad
0x107736:      nop      word ptr cs:[rax + rax]
0x107740:      and      r12d, 1
0x107744:      je       0x10761a
0x10774a:      xor      eax, eax
0x10774c:      call     0x10d330
0x107751:      jmp      0x1075ad
0x107756:      nop      word ptr cs:[rax + rax]
0x107760:      and      r12d, 1
0x107764:      je       0x10761a
0x10776a:      xor      eax, eax
0x10776c:      call     0x10d2b0
0x107771:      jmp      0x1075ad
0x107776:      nop      word ptr cs:[rax + rax]
0x107780:      and      r12d, 1
0x107784:      je       0x10761a
0x10778a:      xor      eax, eax
0x10778c:      call     0x10cf80
0x107791:      jmp      0x1075ad
0x107796:      nop      word ptr cs:[rax + rax]
0x1077a0:      and      r12d, 1
0x1077a4:      je       0x10761a
0x1077aa:      mov      edi, 1
0x1077af:      call     0x10d1a0
0x1077b4:      jmp      0x1075ad
0x1077b9:      nop      dword ptr [rax]

```

```

0x1077c0:    and     r12d, 1
0x1077c4:    je      0x10761a
0x1077ca:    xor     eax, eax
0x1077cc:    call    0x10cf10
0x1077d1:    jmp     0x1075ad
0x1077d6:    nop     word ptr cs:[rax + rax]
0x107812:    nop     word ptr [rax + rax]
0x10782c:    nop     dword ptr [rax]
0x107830:    endbr64
0x107834:    xor     eax, eax
0x107836:    cmp     dword ptr [rip + 0x2315f], 6
0x10783d:    jle     0x107869
0x107869:    ret
0x10783f:    mov     edx, dword ptr [rip + 0x22413]
0x107845:    mov     eax, dword ptr [rip + 0x2314d]
0x10784b:    test    edx, edx
0x10784d:    je      0x107860
0x107860:    cmp     eax, 5
0x107863:    setg    al
0x107866:    movzx   eax, al
0x107869:    ret
0x10784f:    cmp     eax, 0xb
0x107852:    setg    al
0x107855:    movzx   eax, al
0x107858:    ret
0x107859:    nop     dword ptr [rax]
0x10786a:    nop     word ptr [rax + rax]
0x107870:    endbr64
0x107874:    xor     eax, eax
0x107876:    cmp     dword ptr [rip + 0x23067], 1
0x10787d:    sete    al
0x107880:    ret
0x107881:    nop     word ptr cs:[rax + rax]
0x10788c:    nop     dword ptr [rax]
0x107890:    endbr64
0x107894:    push    r13
0x107896:    xor     eax, eax
0x107898:    push    r12
0x10789a:    push    rbp
0x10789b:    push    rbx
0x10789c:    mov     ebx, edi
0x10789e:    sub     rsp, 8
0x1078a2:    mov     ebp, dword ptr [rip + 0x230f4]
0x1078a8:    mov     r12d, dword ptr [rip + 0x230e9]
0x1078af:    call    0x10b520
0x1078b4:    mov     ecx, dword ptr [rip + 0x230e2]
0x1078ba:    test    ebx, ebx
0x1078bc:    jne     0x1078c6

```

```

0x1078be:    cmp     ecx, ebp
0x1078c0:    je      0x107b00
0x1078c6:    mov     edx, dword ptr [rip + 0x230bc]
0x1078cc:    test    edx, edx
0x1078ce:    jne     0x107a60
0x107b00:    cmp     dword ptr [rip + 0x22e91], r12d
0x107b07:    jne     0x1078c6
0x1078d4:    mov     rdi, qword ptr [rip + 0x22fd5]
0x1078db:    cmp     ecx, 1
0x1078de:    jle     0x1079e0
0x107a60:    imul    ecx, dword ptr [rip + 0x22f31]
0x107a67:    mov     rdi, qword ptr [rip + 0x22e5a]
0x107a6e:    lea     esi, [rcx + rcx + 4]
0x107a72:    movsxd  rsi, esi
0x107a75:    call    0x11d690
0x107b0d:    add     rsp, 8
0x107b11:    pop     rbx
0x107b12:    pop     rbp
0x107b13:    pop     r12
0x107b15:    pop     r13
0x107b17:    ret
0x1079e0:    mov     r8d, 0x50
0x1079e6:    mov     ecx, 2
0x1079eb:    mov     edx, 0x10000
0x1079f0:    mov     esi, 0x10000
0x1079f5:    call    0x10f590
0x1078e4:    cmp     ecx, 2
0x1078e7:    je      0x107b70
0x107a7a:    mov     ecx, dword ptr [rip + 0x22f1c]
0x107a80:    mov     qword ptr [rip + 0x22e41], rax
0x107a87:    jmp     0x1078d4
0x1079fa:    mov     r8d, 0x50
0x107a00:    mov     ecx, 2
0x107a05:    mov     rdi, qword ptr [rip + 0x22e9c]
0x107a0c:    mov     edx, 0x10000
0x107a11:    mov     esi, 0x10000
0x107a16:    call    0x10f590
0x107b70:    mov     r8d, 0x50
0x107b76:    mov     ecx, 2
0x107b7b:    mov     edx, 0x10000
0x107b80:    mov     esi, 0x10000
0x107b85:    call    0x10f590
0x1078ed:    cmp     ecx, 6
0x1078f0:    jle     0x10790d
0x107a1b:    xor     esi, esi
0x107a1d:    xor     edi, edi
0x107a1f:    call    0x116d30
0x107b8a:    mov     r8d, 0x50

```

```

0x107b90:    mov     ecx, 2
0x107b95:    mov     rdi, qword ptr [rip + 0x22d0c]
0x107b9c:    mov     edx, 0x10000
0x107ba1:    mov     esi, 0x10000
0x107ba6:    call    0x10f590
0x10790d:    mov     r8d, 0x50
0x107913:    mov     ecx, 2
0x107918:    mov     edx, 0x10000
0x10791d:    mov     esi, 0x10000
0x107922:    call    0x10f590
0x1078f2:    mov     eax, dword ptr [rip + 0x22360]
0x1078f8:    movsxd  r8, dword ptr [rip + 0x23099]
0x1078ff:    test    eax, eax
0x107901:    je      0x107a90
0x107a24:    mov     eax, dword ptr [rip + 0x22f72]
0x107a2a:    xor     edi, edi
0x107a2c:    lea     esi, [rax - 1]
0x107a2f:    movsxd  rsi, esi
0x107a32:    call    0x10de50
0x107bab:    xor     esi, esi
0x107bad:    xor     edi, edi
0x107baf:    call    0x116d30
0x107927:    mov     r8d, 0x50
0x10792d:    mov     ecx, 2
0x107932:    mov     rdi, qword ptr [rip + 0x22f6f]
0x107939:    mov     edx, 0x10000
0x10793e:    mov     esi, 0x10000
0x107943:    call    0x10f590
0x107a90:    cmp     r8d, 5
0x107a94:    jle     0x10790d
0x107907:    cmp     r8d, 0xb
0x10790b:    jg      0x107950
0x107a37:    mov     eax, dword ptr [rip + 0x22f5f]
0x107a3d:    mov     edi, dword ptr [rip + 0x22f55]
0x107a43:    cmp     eax, 1
0x107a46:    lea     esi, [rax - 2]
0x107a49:    mov     eax, 0
0x107a4e:    cmov     esi, eax
0x107a51:    add     rsp, 8
0x107a55:    pop     rbx
0x107a56:    pop     rbp
0x107a57:    pop     r12
0x107a59:    pop     r13
0x107a5b:    jmp     0x10c890
0x107bb4:    mov     eax, dword ptr [rip + 0x22de2]
0x107bba:    lea     esi, [rax - 1]
0x107bbd:    movsxd  rsi, esi
0x107bc0:    jmp     0x107aec

```



```

0x107948:      jmp      0x107ad2
0x107a9a:      sub      ecx, 3
0x107a9d:      mov      edx, 1
0x107aa2:      xor      esi, esi
0x107aa4:      movsxd   rcx, ecx
0x107aa7:      call     0x10f590
0x107950:      mov      ebp, r8d
0x107953:      and      r8d, 1
0x107957:      sar      ebp, 1
0x107959:      lea      r13d, [r8 + rbp]
0x10795d:      movsxd   rbp, ebp
0x107960:      call     0x10f470
0x107aec:      movsxd   rdi, dword ptr [rip + 0x22ea5]
0x107af3:      jmp      0x107a32
0x107ad2:      movsxd   rdi, dword ptr [rip + 0x22ebf]
0x107ad9:      xor      esi, esi
0x107adb:      call     0x116d30
0x107aac:      mov      eax, dword ptr [rip + 0x22eea]
0x107ab2:      mov      edx, 1
0x107ab7:      xor      esi, esi
0x107ab9:      movsxd   r8, dword ptr [rip + 0x22ed8]
0x107ac0:      mov      rdi, qword ptr [rip + 0x22de1]
0x107ac7:      lea      ecx, [rax - 3]
0x107aca:      movsxd   rcx, ecx
0x107acd:      call     0x10f590
0x107965:      movsxd   r13, r13d
0x107968:      mov      rdi, rax
0x10796b:      call     0x10c6a0
0x107a32:      call     0x10de50
0x107ae0:      mov      eax, dword ptr [rip + 0x22eb6]
0x107ae6:      lea      esi, [rax - 1]
0x107ae9:      movsxd   rsi, esi
0x107aec:      movsxd   rdi, dword ptr [rip + 0x22ea5]
0x107af3:      jmp      0x107a32
0x107970:      mov      rdi, qword ptr [rip + 0x22f31]
0x107977:      mov      r12d, eax
0x10797a:      call     0x10f470
0x10797f:      mov      rdi, rax
0x107982:      call     0x10c6a0
0x107987:      cmp      r12d, eax
0x10798a:      jg       0x107b20
0x107990:      mov      eax, dword ptr [rip + 0x23006]
0x107996:      mov      r8, r13
0x107999:      xor      esi, esi
0x10799b:      mov      edx, 1
0x1079a0:      mov      rdi, qword ptr [rip + 0x22f09]
0x1079a7:      lea      ecx, [rax - 3]
0x1079aa:      movsxd   rcx, ecx

```

```

0x1079ad:    call    0x10f590
0x107b20:    mov     eax, dword ptr [rip + 0x22e76]
0x107b26:    mov     rdi, qword ptr [rip + 0x22d83]
0x107b2d:    mov     r8, rbp
0x107b30:    mov     rsi, r13
0x107b33:    mov     edx, 1
0x107b38:    lea     ecx, [rax - 3]
0x107b3b:    movsxd  rcx, ecx
0x107b3e:    call    0x10f590
0x1079b2:    mov     r8, rbp
0x1079b5:    mov     edx, 1
0x1079ba:    mov     rsi, r13
0x1079bd:    mov     eax, dword ptr [rip + 0x22fd9]
0x1079c3:    mov     rdi, qword ptr [rip + 0x22ede]
0x1079ca:    lea     ecx, [rax - 3]
0x1079cd:    movsxd  rcx, ecx
0x1079d0:    call    0x10f590
0x107b43:    mov     eax, dword ptr [rip + 0x22e53]
0x107b49:    mov     r8, r13
0x107b4c:    xor     esi, esi
0x107b4e:    mov     rdi, qword ptr [rip + 0x22d53]
0x107b55:    mov     edx, 1
0x107b5a:    lea     ecx, [rax - 3]
0x107b5d:    movsxd  rcx, ecx
0x107b60:    call    0x10f590
0x1079d5:    jmp     0x107ad2
0x107b65:    jmp     0x107ad2
0x10794d:    nop     dword ptr [rax]
0x1079da:    nop     word ptr [rax + rax]
0x107a8c:    nop     dword ptr [rax]
0x107af8:    nop     dword ptr [rax + rax]
0x107b18:    nop     dword ptr [rax + rax]
0x107b6a:    nop     word ptr [rax + rax]
0x107bc5:    nop     word ptr cs:[rax + rax]
0x107bd0:    endbr64
0x107bd4:    push    rbx
0x107bd5:    xor     eax, eax
0x107bd7:    mov     ebx, edi
0x107bd9:    xor     edi, edi
0x107bdb:    call    0x107890
0x107be0:    cmp     ebx, 0x12
0x107be3:    je      0x107cc8
0x107cc8:    mov     edi, 1
0x107ccd:    call    0x109b00
0x107be9:    mov     esi, dword ptr [rip + 0x22ced]
0x107bef:    test    esi, esi
0x107bf1:    je      0x107bf8
0x107cd2:    xor     eax, eax

```

```

0x107cd4:    call    0x10a6c0
0x107bf8:    mov     rdi, qword ptr [rip + 0x22cb1]
0x107bff:    call    0x10f480
0x107bf3:    pop     rbx
0x107bf4:    ret
0x107cd9:    jmp     0x107be9
0x107c04:    mov     rdi, qword ptr [rip + 0x22c9d]
0x107c0b:    call    0x10f480
0x107c10:    mov     rdi, qword ptr [rip + 0x22c99]
0x107c17:    call    0x10f530
0x107c1c:    mov     rdi, qword ptr [rip + 0x22c85]
0x107c23:    call    0x10f530
0x107c28:    mov     ecx, dword ptr [rip + 0x22cb6]
0x107c2e:    test    ecx, ecx
0x107c30:    jne     0x107ce0
0x107c36:    cmp     dword ptr [rip + 0x22d5f], 6
0x107c3d:    jle     0x107c60
0x107ce0:    mov     rdi, qword ptr [rip + 0x22be1]
0x107ce7:    call    0x10bb40
0x107c60:    xor     eax, eax
0x107c62:    call    0x10a6c0
0x107c3f:    mov     edx, dword ptr [rip + 0x22013]
0x107c45:    mov     eax, dword ptr [rip + 0x22d4d]
0x107c4b:    test    edx, edx
0x107c4d:    je      0x107d00
0x107cec:    jmp     0x107c8b
0x107c67:    xor     eax, eax
0x107c69:    call    0x10a300
0x107d00:    cmp     eax, 5
0x107d03:    jle     0x107c60
0x107c53:    cmp     eax, 0xb
0x107c56:    jg      0x107c6e
0x107c8b:    xor     eax, eax
0x107c8d:    call    0x10dea0
0x107c6e:    xor     eax, eax
0x107c70:    call    0x116d50
0x107d09:    jmp     0x107c6e
0x107c58:    nop     dword ptr [rax + rax]
0x107c92:    xor     eax, eax
0x107c94:    call    0x10d870
0x107c75:    mov     rdi, qword ptr [rip + 0x22c34]
0x107c7c:    call    0x111010
0x107c99:    xor     eax, eax
0x107c9b:    call    0x10d740
0x107c81:    mov     eax, dword ptr [rip + 0x21fd1]
0x107c87:    test    eax, eax
0x107c89:    jne     0x107cf0
0x107ca0:    xor     eax, eax

```

```

0x107ca2:      call    0x10a820
0x107cf0:      mov     rdi, qword ptr [rip + 0x22bb1]
0x107cf7:      call    0x111010
0x107ca7:      cmp     ebx, 0x12
0x107caa:      jne     0x107bf3
0x107cfc:      jmp     0x107c8b
0x107cb0:      mov     rdi, qword ptr [rip + 0x22bf9]
0x107cb7:      call    0x111100
0x107cbc:      pop     rbx
0x107cbd:      mov     rdi, rax
0x107cc0:      jmp     0x10c4b0
0x107bf5:      nop     dword ptr [rax]
0x107cc5:      nop     dword ptr [rax]
0x107cde:      nop
0x107cee:      nop
0x107cfe:      nop
0x107d0e:      nop
0x107d10:      endbr64
0x107d14:      push    r12
0x107d16:      xor     eax, eax
0x107d18:      mov     r12, rdi
0x107d1b:      push    rbp
0x107d1c:      push    rbx
0x107d1d:      call    0x10b510
0x107d22:      mov     rdi, rax
0x107d25:      mov     rbp, rax
0x107d28:      call    0x1049b0
0x107d2d:      test    rax, rax
0x107d30:      je      0x107d39
0x107d39:      xor     eax, eax
0x107d3b:      call    0x10a5a0
0x107d32:      cmp     byte ptr [rbp + rax - 1], 7
0x107d37:      jne     0x107d70
0x107d40:      test    r12, r12
0x107d43:      je      0x107ded
0x107d70:      mov     rdi, rbp
0x107d73:      call    0x109fe0
0x107ded:      xor     eax, eax
0x107def:      call    0x10e4e0
0x107d49:      xor     edx, edx
0x107d4b:      xor     esi, esi
0x107d4d:      mov     rdi, r12
0x107d50:      call    0x10e480
0x107d78:      mov     rdi, rax
0x107d7b:      mov     rbx, rax
0x107d7e:      call    0x1049b0
0x107df4:      jmp     0x107d55
0x107d55:      xor     eax, eax

```

```

0x107d57:    call    0x10d740
0x107d83:    lea     rdi, [rax + 0x80]
0x107d8a:    call    0x11d670
0x107d5c:    pop     rbx
0x107d5d:    xor     eax, eax
0x107d5f:    pop     rbp
0x107d60:    pop     r12
0x107d62:    jmp     0x10a820
0x107d8f:    mov     r8, rbx
0x107d92:    mov     esi, 1
0x107d97:    lea     rcx, [rip + 0x18762]
0x107d9e:    mov     rbp, rax
0x107da1:    mov     rdi, rax
0x107da4:    mov     rdx, -1
0x107dab:    xor     eax, eax
0x107dad:    call    0x104f10
0x107db2:    mov     edx, 1
0x107db7:    mov     esi, 2
0x107dbc:    mov     rdi, rbp
0x107dbf:    call    0x10e480
0x107dc4:    mov     rdi, rbp
0x107dc7:    call    0x118d10
0x107dcc:    xor     eax, eax
0x107dce:    call    0x10a5a0
0x107dd3:    xor     eax, eax
0x107dd5:    call    0x10a820
0x107dda:    mov     edi, 1
0x107ddf:    call    0x104ea0
0x107de4:    test    r12, r12
0x107de7:    jne     0x107d49
0x107d67:    nop     word ptr [rax + rax]
0x107df9:    nop     dword ptr [rax]
0x107e00:    endbr64
0x107e04:    push    r13
0x107e06:    mov     edx, 1
0x107e0b:    push    r12
0x107e0d:    mov     r12, rsi
0x107e10:    push    rbp
0x107e11:    mov     rbp, rdi
0x107e14:    sub     rsp, 0x10
0x107e18:    mov     rax, qword ptr fs:[0x28]
0x107e21:    mov     qword ptr [rsp + 8], rax
0x107e26:    xor     eax, eax
0x107e28:    mov     r13, rsp
0x107e2b:    mov     rsi, r13
0x107e2e:    call    0x119200
0x107e33:    test    rax, rax
0x107e36:    je      0x107e48

```

```

0x107e48:    mov     rdi, r12
0x107e4b:    call   0x11d7a0
0x107e38:    mov     rdi, qword ptr [rsp]
0x107e3c:    mov     rsi, r12
0x107e3f:    call   0x104b30
0x107e50:    mov     rsi, r13
0x107e53:    mov     rdi, rbp
0x107e56:    mov     qword ptr [rsp], rax
0x107e5a:    call   0x119150
0x107e44:    test    eax, eax
0x107e46:    je      0x107e5f
0x107e5f:    mov     rax, qword ptr [rsp + 8]
0x107e64:    sub     rax, qword ptr fs:[0x28]
0x107e6d:    jne     0x107e79
0x107e6f:    add     rsp, 0x10
0x107e73:    pop     rbp
0x107e74:    pop     r12
0x107e76:    pop     r13
0x107e78:    ret
0x107e79:    call   0x1049d0
0x107e7e:    nop
0x107e80:    endbr64
0x107e84:    sub     rsp, 0x18
0x107e88:    mov     edx, esi
0x107e8a:    mov     rax, qword ptr fs:[0x28]
0x107e93:    mov     qword ptr [rsp + 8], rax
0x107e98:    xor     eax, eax
0x107e9a:    mov     rsi, rsp
0x107e9d:    call   0x119200
0x107ea2:    test    rax, rax
0x107ea5:    cmovne  rax, qword ptr [rsp]
0x107eaa:    mov     rdx, qword ptr [rsp + 8]
0x107eaf:    sub     rdx, qword ptr fs:[0x28]
0x107eb8:    jne     0x107ebf
0x107eba:    add     rsp, 0x18
0x107ebe:    ret
0x107ebf:    call   0x1049d0
0x107ec4:    nop     word ptr cs:[rax + rax]
0x107ecf:    nop
0x107ed0:    endbr64
0x107ed4:    test    sil, 2
0x107ed8:    je      0x107ee0
0x107ee0:    xor     eax, eax
0x107ee2:    ret
0x107eda:    jmp     0x107570
0x107edf:    nop
0x107ee3:    nop     word ptr cs:[rax + rax]
0x107eee:    nop

```

```

0x107ef0:      endbr64
0x107ef4:      push     r15
0x107ef6:      push     r14
0x107ef8:      push     r13
0x107efa:      push     r12
0x107efc:      push     rbp
0x107efd:      push     rbx
0x107efe:      sub      rsp, 8
0x107f02:      test     rdi, rdi
0x107f05:      je       0x107fc8
0x107fc8:      xor      r15d, r15d
0x107fcb:      jmp      0x107fb0
0x107f0b:      mov      rbp, rdi
0x107f0e:      call     0x1049b0
0x107fb0:      add      rsp, 8
0x107fb4:      mov      rax, r15
0x107fb7:      pop      rbx
0x107fb8:      pop      rbp
0x107fb9:      pop      r12
0x107fbb:      pop      r13
0x107fbd:      pop      r14
0x107fbf:      pop      r15
0x107fc1:      ret
0x107f13:      lea      r13, [rax + 1]
0x107f17:      mov      rdi, r13
0x107f1a:      call     0x11d670
0x107f1f:      movzx    ebx, byte ptr [rbp]
0x107f23:      mov      r15, rax
0x107f26:      test     bl, bl
0x107f28:      je       0x107fd0
0x107fd0:      mov      r12, rax
0x107fd3:      jmp      0x107fab
0x107f2e:      add      rbp, 1
0x107f32:      xor      r12d, r12d
0x107f35:      jmp      0x107f6d
0x107fab:      mov      byte ptr [r12], 0
0x107fb0:      add      rsp, 8
0x107fb4:      mov      rax, r15
0x107fb7:      pop      rbx
0x107fb8:      pop      rbp
0x107fb9:      pop      r12
0x107fbb:      pop      r13
0x107fbd:      pop      r14
0x107fbf:      pop      r15
0x107fc1:      ret
0x107f6d:      movsxd   r14, r12d
0x107f70:      cmp      bl, 9
0x107f73:      jne      0x107f40

```

```

0x107f75:    add     r13, 8
0x107f79:    mov     rdi, r15
0x107f7c:    add     rbp, 1
0x107f80:    add     r12d, 8
0x107f84:    mov     rsi, r13
0x107f87:    call    0x11d690
0x107f40:    call    0x104f00
0x107f8c:    mov     r15, rax
0x107f8f:    movabs  rax, 0x2020202020202020
0x107f99:    mov     qword ptr [r15 + r14], rax
0x107f9d:    movzx   ebx, byte ptr [rbp - 1]
0x107fa1:    test    bl, bl
0x107fa3:    jne     0x107f6d
0x107f45:    movsx   rdx, bl
0x107f49:    add     r12d, 1
0x107f4d:    mov     rax, qword ptr [rax]
0x107f50:    test    byte ptr [rax + rdx*2 + 1], 0x40
0x107f55:    mov     eax, 0x3f
0x107f5a:    cmov     ebx, eax
0x107f5d:    add     rbp, 1
0x107f61:    mov     byte ptr [r15 + r14], bl
0x107f65:    movzx   ebx, byte ptr [rbp - 1]
0x107f69:    test    bl, bl
0x107f6b:    je      0x107fa5
0x107fa5:    movsxd  r12, r12d
0x107fa8:    add     r12, r15
0x107fab:    mov     byte ptr [r12], 0
0x107fb0:    add     rsp, 8
0x107fb4:    mov     rax, r15
0x107fb7:    pop     rbx
0x107fb8:    pop     rbp
0x107fb9:    pop     r12
0x107fbb:    pop     r13
0x107fbd:    pop     r14
0x107fbf:    pop     r15
0x107fc1:    ret
0x107f37:    nop     word ptr [rax + rax]
0x107fc2:    nop     word ptr [rax + rax]
0x107fcd:    nop     dword ptr [rax]
0x107fd5:    nop     word ptr cs:[rax + rax]
0x107fe0:    endbr64
0x107fe4:    push    rbx
0x107fe5:    mov     rbx, rdi
0x107fe8:    call    0x1049b0
0x107fed:    lea     rdi, [rax + rax*2 + 9]
0x107ff2:    call    0x11d670
0x107ff7:    mov     byte ptr [rax], 0x28
0x107ffa:    cmp     byte ptr [rbx + 1], 0

```



```

0x107ffe:    je      0x108050
0x108050:    mov     ecx, 4
0x108055:    mov     edi, 3
0x10805a:    mov     r8d, 2
0x108060:    mov     edx, 1
0x108065:    jmp     0x108035
0x108000:    mov     edx, 1
0x108005:    nop     dword ptr [rax]
0x108008:    movzx   ecx, byte ptr [rbx]
0x10800b:    mov     byte ptr [rax + rdx + 1], 0x2c
0x108010:    add     rbx, 1
0x108014:    mov     byte ptr [rax + rdx], cl
0x108017:    mov     rcx, rdx
0x10801a:    add     rdx, 3
0x10801e:    mov     byte ptr [rax + rdx - 1], 0x20
0x108023:    cmp     byte ptr [rbx + 1], 0
0x108027:    jne     0x108008
0x108035:    movzx   esi, byte ptr [rbx]
0x108038:    mov     byte ptr [rax + rdx], sil
0x10803c:    mov     byte ptr [rax + r8], 0x29
0x108041:    mov     byte ptr [rax + rdi], 0x20
0x108045:    mov     byte ptr [rax + rcx], 0
0x108049:    pop     rbx
0x10804a:    ret
0x108029:    lea     r8, [rcx + 4]
0x10802d:    lea     rdi, [rcx + 5]
0x108031:    add     rcx, 6
0x108035:    movzx   esi, byte ptr [rbx]
0x108038:    mov     byte ptr [rax + rdx], sil
0x10803c:    mov     byte ptr [rax + r8], 0x29
0x108041:    mov     byte ptr [rax + rdi], 0x20
0x108045:    mov     byte ptr [rax + rcx], 0
0x108049:    pop     rbx
0x10804a:    ret
0x108008:    movzx   ecx, byte ptr [rbx]
0x10800b:    mov     byte ptr [rax + rdx + 1], 0x2c
0x108010:    add     rbx, 1
0x108014:    mov     byte ptr [rax + rdx], cl
0x108017:    mov     rcx, rdx
0x10801a:    add     rdx, 3
0x10801e:    mov     byte ptr [rax + rdx - 1], 0x20
0x108023:    cmp     byte ptr [rbx + 1], 0
0x108027:    jne     0x108008
0x10804b:    nop     dword ptr [rax + rax]
0x108067:    nop     word ptr [rax + rax]
0x108070:    endbr64
0x108074:    push    r15
0x108076:    push    r14

```

```

0x108078:    push    r13
0x10807a:    push    r12
0x10807c:    mov     r12, rdi
0x10807f:    push    rbp
0x108080:    mov     rbp, rsi
0x108083:    push    rbx
0x108084:    mov     ebx, edx
0x108086:    sub     rsp, 0x28
0x10808a:    mov     rax, qword ptr fs:[0x28]
0x108093:    mov     qword ptr [rsp + 0x18], rax
0x108098:    mov     eax, edx
0x10809a:    and     eax, 0x10
0x10809d:    mov     dword ptr [rsp + 0xc], eax
0x1080a1:    jne     0x1082f0
0x1080a7:    xor     r13d, r13d
0x1080aa:    xor     eax, eax
0x1080ac:    call    0x10cff0
0x1082f0:    xor     eax, eax
0x1082f2:    call    0x10c8c0
0x1080b1:    test    r12, r12
0x1080b4:    je      0x108103
0x1082f7:    mov     r13, rax
0x1082fa:    jmp     0x1080aa
0x108103:    xor     eax, eax
0x108105:    call    0x10d870
0x1080b6:    mov     rdi, r12
0x1080b9:    call    0x107ef0
0x1080aa:    xor     eax, eax
0x1080ac:    call    0x10cff0
0x10810a:    xor     eax, eax
0x10810c:    call    0x10d740
0x1080be:    mov     r12, rax
0x1080c1:    test    bl, 8
0x1080c4:    jne     0x108330
0x108111:    xor     eax, eax
0x108113:    call    0x10a820
0x1080ca:    mov     rdi, r12
0x1080cd:    call    0x10d610
0x108330:    lea     rdi, [rip + 0x181da]
0x108337:    call    0x10d610
0x108118:    test    bl, 4
0x10811b:    jne     0x108300
0x1080d2:    test    bl, 0x20
0x1080d5:    jne     0x108310
0x10833c:    mov     edi, 1
0x108341:    call    0x10dba0
0x108121:    mov     r14d, ebx
0x108124:    lea     r12, [rsp + 0x14]

```

```

0x108129:    and     r14d, 2
0x10812d:    mov     rdi, r12
0x108130:    call    0x10b180
0x108300:    xor     eax, eax
0x108302:    call    0x10a5a0
0x1080db:    mov     rdi, r12
0x1080de:    call    0x118d10
0x108310:    xor     eax, eax
0x108312:    call    0x10ccf0
0x108346:    mov     rdi, r12
0x108349:    call    0x10d610
0x108135:    test    rax, rax
0x108138:    je      0x1081b2
0x108307:    jmp     0x108121
0x1080e3:    test    rbp, rbp
0x1080e6:    je      0x108103
0x108317:    mov     rdi, r12
0x10831a:    call    0x118d10
0x10834e:    test    bl, 0x20
0x108351:    je      0x1080db
0x1081b2:    xor     eax, eax
0x1081b4:    call    0x10a5a0
0x10813a:    mov     rdx, qword ptr [rax]
0x10813d:    mov     rax, qword ptr [rax + 0x10]
0x108141:    movzx   r15d, byte ptr [rdx]
0x108145:    test    rax, rax
0x108148:    je      0x108250
0x1080e8:    mov     rdi, rbp
0x1080eb:    call    0x107fe0
0x10831f:    test    rbp, rbp
0x108322:    jne     0x1080e8
0x108357:    jmp     0x108310
0x1081b9:    xor     eax, eax
0x1081bb:    call    0x10dea0
0x108250:    cmp     r15d, 7
0x108254:    je      0x1082b0
0x10814e:    cmp     byte ptr [rax + 0x1b], 0
0x108152:    je      0x108250
0x1080f0:    mov     r12, rax
0x1080f3:    mov     rdi, rax
0x1080f6:    call    0x10d610
0x108328:    jmp     0x108103
0x1081c0:    xor     eax, eax
0x1081c2:    call    0x10d740
0x1082b0:    xor     edi, edi
0x1082b2:    call    0x10dba0
0x108256:    mov     eax, dword ptr [rsp + 0x14]
0x10825a:    lea     edx, [rax - 1]

```

```

0x10825d:    mov     dword ptr [rsp + 0x14], edx
0x108261:    test    eax, eax
0x108263:    je      0x1083a8
0x108158:    mov     rcx, qword ptr [rax]
0x10815b:    lea     rax, [rip + 0x20ebe]
0x108162:    sub     rcx, rax
0x108165:    movabs  rax, 0xea0ea0ea0ea0ea1
0x10816f:    imul    rcx
0x108172:    sar     rcx, 0x3f
0x108176:    sar     rdx, 1
0x108179:    sub     rdx, rcx
0x10817c:    mov     r15d, edx
0x10817f:    not     r15d
0x108182:    cmp     edx, 7
0x108185:    jne     0x1081d0
0x1080fb:    mov     rdi, r12
0x1080fe:    call    0x118d10
0x1081c7:    jmp     0x10812d
0x1082b7:    mov     ecx, dword ptr [rsp + 0xc]
0x1082bb:    test    ecx, ecx
0x1082bd:    je      0x1082e1
0x1083a8:    call    0x10d870
0x108269:    test    r14d, r14d
0x10826c:    je      0x108280
0x108187:    xor     edi, edi
0x108189:    xor     eax, eax
0x10818b:    call    0x107bd0
0x1081d0:    cmp     r15d, 7
0x1081d4:    je      0x1082b0
0x10812d:    mov     rdi, r12
0x108130:    call    0x10b180
0x1082e1:    xor     eax, eax
0x1082e3:    jmp     0x10822c
0x1082bf:    test    r13, r13
0x1082c2:    je      0x1082e1
0x1083ad:    jmp     0x108190
0x108280:    test    rbp, rbp
0x108283:    je      0x1083b8
0x10826e:    mov     esi, ebx
0x108270:    mov     edi, r15d
0x108273:    call    0x107570
0x108190:    xor     eax, eax
0x108192:    call    0x10dea0
0x1081da:    cmp     r15d, -0x3d
0x1081de:    jne     0x108256
0x10822c:    mov     rdx, qword ptr [rsp + 0x18]
0x108231:    sub     rdx, qword ptr fs:[0x28]
0x10823a:    jne     0x1083d9

```

```

0x1082c4:    mov     rdi, r13
0x1082c7:    call   0x10c970
0x1083b8:    xor     edi, edi
0x1083ba:    call   0x10dba0
0x108289:    mov     esi, r15d
0x10828c:    mov     rdi, rbp
0x10828f:    call   0x104a00
0x108278:    test    eax, eax
0x10827a:    jne     0x108256
0x108197:    xor     eax, eax
0x108199:    call   0x10d740
0x1081e0:    test    rbp, rbp
0x1081e3:    je      0x108360
0x108240:    add     rsp, 0x28
0x108244:    pop     rbx
0x108245:    pop     rbp
0x108246:    pop     r12
0x108248:    pop     r13
0x10824a:    pop     r14
0x10824c:    pop     r15
0x10824e:    ret
0x1083d9:    call   0x1049d0
0x1082cc:    xor     eax, eax
0x1082ce:    call   0x10d870
0x1083bf:    mov     eax, dword ptr [rsp + 0xc]
0x1083c3:    test    eax, eax
0x1083c5:    je      0x108229
0x108294:    test    rax, rax
0x108297:    jne     0x1083b8
0x10827c:    nop     dword ptr [rax]
0x10819e:    xor     eax, eax
0x1081a0:    call   0x10a820
0x108360:    xor     edi, edi
0x108362:    call   0x10dba0
0x1081e9:    xor     edi, edi
0x1081eb:    movsx   r15d, byte ptr [rbp]
0x1081f0:    call   0x10dba0
0x1082d3:    xor     eax, eax
0x1082d5:    call   0x10d740
0x108229:    mov     eax, r15d
0x10822c:    mov     rdx, qword ptr [rsp + 0x18]
0x108231:    sub     rdx, qword ptr fs:[0x28]
0x10823a:    jne     0x1083d9
0x1083cb:    test    r13, r13
0x1083ce:    jne     0x108202
0x10829d:    call   0x10a5a0
0x1081a5:    mov     rdi, r12
0x1081a8:    call   0x10b180

```

```

0x108367:    mov     edx, dword ptr [rsp + 0xc]
0x10836b:    mov     eax, 0xffffffffc3
0x108370:    test    edx, edx
0x108372:    je      0x10822c
0x1081f5:    mov     esi, dword ptr [rsp + 0xc]
0x1081f9:    test    esi, esi
0x1081fb:    je      0x10821f
0x1082da:    xor     eax, eax
0x1082dc:    call    0x10a820
0x1083d4:    jmp     0x108229
0x108202:    mov     rdi, r13
0x108205:    call    0x10c970
0x1082a2:    jmp     0x108190
0x1081ad:    test    rax, rax
0x1081b0:    jne     0x10813a
0x108378:    test    r13, r13
0x10837b:    je      0x10822c
0x10821f:    cmp     r15d, 7
0x108223:    je      0x1082e1
0x1081fd:    test    r13, r13
0x108200:    je      0x10821f
0x10820a:    xor     eax, eax
0x10820c:    call    0x10d870
0x108381:    mov     rdi, r13
0x108384:    call    0x10c970
0x108211:    xor     eax, eax
0x108213:    call    0x10d740
0x108389:    xor     eax, eax
0x10838b:    call    0x10d870
0x108218:    xor     eax, eax
0x10821a:    call    0x10a820
0x108390:    xor     eax, eax
0x108392:    call    0x10d740
0x108397:    xor     eax, eax
0x108399:    call    0x10a820
0x10839e:    jmp     0x108229
0x1081cc:    nop     dword ptr [rax]
0x10824f:    nop
0x1082a7:    nop     word ptr [rax + rax]
0x1082e8:    nop     dword ptr [rax + rax]
0x1082ff:    nop
0x10830c:    nop     dword ptr [rax]
0x10832d:    nop     dword ptr [rax]
0x108359:    nop     dword ptr [rax]
0x1083a3:    nop     dword ptr [rax + rax]
0x1083b2:    nop     word ptr [rax + rax]
0x1083de:    nop
0x1083e0:    endbr64

```

```

0x1083e4:    push    r15
0x1083e6:    push    r14
0x1083e8:    mov     r14, rdi
0x1083eb:    push    r13
0x1083ed:    mov     r13, rsi
0x1083f0:    push    r12
0x1083f2:    mov     r12, rdx
0x1083f5:    push    rbp
0x1083f6:    mov     rbp, rcx
0x1083f9:    push    rbx
0x1083fa:    sub     rsp, 0x38
0x1083fe:    mov     rax, qword ptr fs:[0x28]
0x108407:    mov     qword ptr [rsp + 0x28], rax
0x10840c:    xor     eax, eax
0x10840e:    call    0x10cff0
0x108413:    test    r14, r14
0x108416:    je      0x108420
0x108420:    test    r12, r12
0x108423:    je      0x108450
0x108418:    mov     rdi, r14
0x10841b:    call    0x10d540
0x108450:    xor     r14d, r14d
0x108453:    xor     eax, eax
0x108455:    lea     r12, [rsp + 0x1c]
0x10845a:    lea     rbx, [rip + 0x17c03]
0x108461:    call    0x10d870
0x108425:    mov     rdi, r12
0x108428:    call    0x10d610
0x108466:    xor     eax, eax
0x108468:    call    0x10d740
0x10842d:    test    rbp, rbp
0x108430:    je      0x108450
0x10846d:    xor     eax, eax
0x10846f:    call    0x10a820
0x108432:    mov     rsi, r12
0x108435:    mov     rdi, rbp
0x108438:    xor     eax, eax
0x10843a:    mov     r14d, 1
0x108440:    call    0x107e00
0x108474:    lea     rax, [rsp + 0x20]
0x108479:    mov     qword ptr [rsp], rax
0x10847d:    mov     rdi, r12
0x108480:    call    0x10b180
0x108445:    jmp     0x108453
0x108485:    test    rax, rax
0x108488:    je      0x10853f
0x108453:    xor     eax, eax
0x108455:    lea     r12, [rsp + 0x1c]

```

```

0x10845a:    lea     rbx, [rip + 0x17c03]
0x108461:    call   0x10d870
0x10853f:    xor     eax, eax
0x108541:    call   0x10a5a0
0x10848e:    mov     rdx, qword ptr [rax]
0x108491:    mov     rax, qword ptr [rax + 0x10]
0x108495:    movzx   r15d, byte ptr [rdx]
0x108499:    test    rax, rax
0x10849c:    je      0x1084d7
0x108546:    xor     eax, eax
0x108548:    call   0x10dea0
0x1084d7:    lea     eax, [r15 + 0x22]
0x1084db:    cmp     eax, 0x29
0x1084de:    ja      0x108505
0x10849e:    cmp     byte ptr [rax + 0x1b], 0
0x1084a2:    je      0x1084d7
0x10854d:    xor     eax, eax
0x10854f:    call   0x10d740
0x1084e0:    movsxd  rax, dword ptr [rbx + rax*4]
0x1084e4:    add     rax, rbx
0x1084e7:    jmp     rax
0x108505:    mov     eax, dword ptr [rsp + 0x1c]
0x108509:    lea     edx, [rax - 1]
0x10850c:    mov     dword ptr [rsp + 0x1c], edx
0x108510:    test    eax, eax
0x108512:    jne     0x1084f0
0x1084a4:    mov     rcx, qword ptr [rax]
0x1084a7:    lea     rax, [rip + 0x20b72]
0x1084ae:    sub     rcx, rax
0x1084b1:    movabs  rax, 0xea0ea0ea0ea0ea1
0x1084bb:    imul    rcx
0x1084be:    sar     rcx, 0x3f
0x1084c2:    sar     rdx, 1
0x1084c5:    sub     rdx, rcx
0x1084c8:    mov     r15d, edx
0x1084cb:    not     r15d
0x1084ce:    cmp     edx, 0x3c
0x1084d1:    je      0x108560
0x108554:    jmp     0x10847d
0x1086a0:    xor     eax, eax
0x1086a2:    jmp     0x10857f
0x108660:    xor     edi, edi
0x108662:    xor     eax, eax
0x108664:    call   0x107bd0
0x1085a8:    test    rbp, rbp
0x1085ab:    je      0x108519
0x108610:    test    rbp, rbp
0x108613:    je      0x108519

```



```

0x108514:      call    0x10d870
0x1084f0:      mov     esi, 3
0x1084f5:      mov     edi, r15d
0x1084f8:      call    0x107570
0x108560:      mov     rdi, r13
0x108563:      call    0x10db10
0x10847d:      mov     rdi, r12
0x108480:      call    0x10b180
0x10857f:      mov     rdx, qword ptr [rsp + 0x28]
0x108584:      sub     rdx, qword ptr fs:[0x28]
0x10858d:      jne     0x1086b3
0x108669:      jmp     0x108519
0x108519:      xor     eax, eax
0x10851b:      call    0x10dea0
0x1085b1:      lea     eax, [r14 + 1]
0x1085b5:      mov     rsi, qword ptr [rsp]
0x1085b9:      mov     rdi, rbp
0x1085bc:      mov     edx, eax
0x1085be:      mov     r15d, eax
0x1085c1:      call    0x119200
0x108619:      xor     edi, edi
0x10861b:      test    r14d, r14d
0x10861e:      je      0x108680
0x1084fd:      test    eax, eax
0x1084ff:      je      0x108670
0x108568:      test    rbp, rbp
0x10856b:      je      0x10857b
0x108593:      add     rsp, 0x38
0x108597:      pop     rbx
0x108598:      pop     rbp
0x108599:      pop     r12
0x10859b:      pop     r13
0x10859d:      pop     r14
0x10859f:      pop     r15
0x1085a1:      ret
0x1086b3:      call    0x1049d0
0x108520:      xor     eax, eax
0x108522:      call    0x10d740
0x1085c6:      test    rax, rax
0x1085c9:      je      0x1086a7
0x108680:      call    0x10d1a0
0x108620:      call    0x10d1a0
0x108670:      call    0x10a5a0
0x10857b:      mov     rax, qword ptr [r13]
0x10857f:      mov     rdx, qword ptr [rsp + 0x28]
0x108584:      sub     rdx, qword ptr fs:[0x28]
0x10858d:      jne     0x1086b3
0x10856d:      mov     rsi, qword ptr [r13]

```

```

0x108571:    mov     rdi, rbp
0x108574:    xor     eax, eax
0x108576:    call    0x107e00
0x108527:    xor     eax, eax
0x108529:    call    0x10a820
0x1086a7:    xor     eax, eax
0x1086a9:    call    0x10a5a0
0x1085cf:    mov     r8, qword ptr [rsp + 0x20]
0x1085d4:    test    r8, r8
0x1085d7:    je      0x1086a7
0x108685:    xor     eax, eax
0x108687:    call    0x10d870
0x108625:    sub     r14d, 1
0x108629:    je      0x10864b
0x108675:    jmp     0x108505
0x10852e:    mov     rdi, r12
0x108531:    call    0x10b180
0x1086ae:    jmp     0x108519
0x1085dd:    xor     edi, edi
0x1085df:    mov     qword ptr [rsp + 8], r8
0x1085e4:    mov     r14d, r15d
0x1085e7:    call    0x10d1a0
0x10868c:    xor     eax, eax
0x10868e:    call    0x10d740
0x10864b:    xor     eax, eax
0x10864d:    call    0x10d870
0x10862b:    mov     rsi, qword ptr [rsp]
0x10862f:    mov     rdi, rbp
0x108632:    mov     edx, r14d
0x108635:    call    0x119200
0x108536:    test    rax, rax
0x108539:    jne     0x10848e
0x1085ec:    mov     rdi, qword ptr [rsp + 8]
0x1085f1:    call    0x10d610
0x108693:    jmp     0x108519
0x108652:    xor     eax, eax
0x108654:    call    0x10d740
0x10863a:    mov     rdi, rax
0x10863d:    test    rax, rax
0x108640:    cmovne  rdi, qword ptr [rsp + 0x20]
0x108646:    call    0x10d610
0x1085f6:    xor     eax, eax
0x1085f8:    call    0x10d870
0x108659:    jmp     0x108519
0x1085fd:    xor     eax, eax
0x1085ff:    call    0x10d740
0x108604:    jmp     0x108519
0x108447:    nop     word ptr [rax + rax]

```

```

0x1084ea:      nop      word ptr [rax + rax]
0x108559:      nop      dword ptr [rax]
0x1085a2:      nop      word ptr [rax + rax]
0x108609:      nop      dword ptr [rax]
0x10865e:      nop
0x10866e:      nop
0x10867a:      nop      word ptr [rax + rax]
0x108698:      nop      dword ptr [rax + rax]
0x1086b8:      nop      dword ptr [rax + rax]
0x1086c0:      endbr64
0x1086c4:      push     r15
0x1086c6:      push     r14
0x1086c8:      push     r13
0x1086ca:      push     r12
0x1086cc:      push     rbp
0x1086cd:      push     rbx
0x1086ce:      sub      rsp, 0x18
0x1086d2:      test     edx, edx
0x1086d4:      je       0x108890
0x108890:      xor      eax, eax
0x108892:      mov      qword ptr [rsp + 8], rdi
0x108897:      call     0x10c8c0
0x1086da:      mov      ebx, edx
0x1086dc:      cmp      edx, 3
0x1086df:      je       0x1088f8
0x10889c:      mov      qword ptr [rip + 0x2205d], rax
0x1088a3:      xor      eax, eax
0x1088a5:      call     0x10cff0
0x1088f8:      mov      rdi, qword ptr [rip + 0x22001]
0x1088ff:      xor      r12d, r12d
0x108902:      call     0x10c970
0x1086e5:      mov      rbp, rcx
0x1086e8:      test     rcx, rcx
0x1086eb:      je       0x108864
0x1088aa:      mov      rdi, qword ptr [rsp + 8]
0x1088af:      test     rdi, rdi
0x1088b2:      je       0x108864
0x108907:      xor      eax, eax
0x108909:      call     0x10d870
0x108864:      xor      r12d, r12d
0x108867:      add      rsp, 0x18
0x10886b:      mov      rax, r12
0x10886e:      pop      rbx
0x10886f:      pop      rbp
0x108870:      pop      r12
0x108872:      pop      r13
0x108874:      pop      r14
0x108876:      pop      r15

```

```

0x108878:      ret
0x1086f1:      mov     dword ptr [rcx], 0
0x1086f7:      xor     eax, eax
0x1086f9:      mov     r13, rsi
0x1086fc:      lea     r15, [rip + 0x2091d]
0x108703:      movabs  r14, 0xea0ea0ea0ea0ea1
0x10870d:      call    0x10d870
0x1088b4:      call    0x10d540
0x10890e:      xor     eax, eax
0x108910:      call    0x10d740
0x108712:      xor     eax, eax
0x108714:      call    0x10d740
0x1088b9:      jmp     0x108864
0x108915:      xor     eax, eax
0x108917:      call    0x10a820
0x108719:      xor     eax, eax
0x10871b:      call    0x10a820
0x10891c:      jmp     0x108867
0x108720:      xor     edi, edi
0x108722:      call    0x10b180
0x108867:      add     rsp, 0x18
0x10886b:      mov     rax, r12
0x10886e:      pop     rbx
0x10886f:      pop     rbp
0x108870:      pop     r12
0x108872:      pop     r13
0x108874:      pop     r14
0x108876:      pop     r15
0x108878:      ret
0x108727:      mov     r12, rax
0x10872a:      test    rax, rax
0x10872d:      je      0x108928
0x108928:      xor     eax, eax
0x10892a:      call    0x10dea0
0x108733:      mov     rax, qword ptr [rax]
0x108736:      movzx   edi, byte ptr [rax]
0x108739:      mov     rax, qword ptr [r12 + 0x10]
0x10873e:      test    rax, rax
0x108741:      je      0x10874d
0x10892f:      xor     eax, eax
0x108931:      call    0x10d740
0x10874d:      cmp     edi, -7
0x108750:      jge     0x1087b0
0x108743:      cmp     byte ptr [rax + 0x1b], 0
0x108747:      jne     0x108808
0x108936:      xor     eax, eax
0x108938:      call    0x10a820
0x108752:      cmp     edi, -0x3d

```

```

0x108755:      je      0x108840
0x1087b0:      cmp     edi, 7
0x1087b3:      je      0x108840
0x108808:      mov     rcx, qword ptr [rax]
0x10880b:      sub     rcx, r15
0x10880e:      mov     rax, rcx
0x108811:      sar     rcx, 0x3f
0x108815:      imul    r14
0x108818:      sar     rdx, 1
0x10881b:      sub     rdx, rcx
0x10881e:      mov     edi, edx
0x108820:      not     edi
0x108822:      cmp     edx, 7
0x108825:      jne     0x10874d
0x10893d:      jmp     0x108867
0x108840:      xor     eax, eax
0x108842:      call    0x10dea0
0x10875b:      cmp     edi, -0x32
0x10875e:      jne     0x1088bb
0x1087b9:      mov     dword ptr [rsp + 8], edi
0x1087bd:      call    0x104f00
0x10882b:      xor     edi, edi
0x10882d:      xor     eax, eax
0x10882f:      call    0x107bd0
0x108847:      xor     eax, eax
0x108849:      call    0x10d870
0x108764:      xor     eax, eax
0x108766:      call    0x10ca40
0x1088bb:      cmp     edi, -0x44
0x1088be:      jne     0x1088df
0x1087c2:      mov     r8, rax
0x1087c5:      movsxd  rax, dword ptr [rsp + 8]
0x1087ca:      mov     rdx, qword ptr [r8]
0x1087cd:      mov     rdi, rax
0x1087d0:      test    byte ptr [rdx + rax*2 + 1], 0x40
0x1087d5:      jne     0x108948
0x108834:      jmp     0x108720
0x10884e:      xor     eax, eax
0x108850:      call    0x10d740
0x10876b:      test    eax, eax
0x10876d:      jne     0x108880
0x1088c0:      cmp     ebx, 1
0x1088c3:      jne     0x1088df
0x1088df:      cmp     edi, -0x45
0x1088e2:      jne     0x1087b9
0x1087db:      xor     eax, eax
0x1087dd:      xor     r12d, r12d
0x1087e0:      call    0x10dea0

```

```

0x108948:      call    0x10cd30
0x108855:      xor     eax, eax
0x108857:      call    0x10a820
0x108773:      mov     dword ptr [rbp], 1
0x10877a:      xor     eax, eax
0x10877c:      call    0x10dea0
0x108880:      xor     eax, eax
0x108882:      call    0x10a5a0
0x1088c5:      xor     eax, eax
0x1088c7:      call    0x10ca40
0x1088e8:      cmp     ebx, 2
0x1088eb:      je      0x1088c5
0x1087e5:      xor     eax, eax
0x1087e7:      call    0x10d870
0x10894d:      mov     dword ptr [rbp], 3
0x108954:      jmp     0x10877a
0x10885c:      mov     rdi, r13
0x10885f:      call    0x10db10
0x108781:      xor     eax, eax
0x108783:      call    0x10d870
0x108887:      jmp     0x10877a
0x1088cc:      cmp     eax, 1
0x1088cf:      mov     eax, 2
0x1088d4:      sbb     eax, -1
0x1088d7:      mov     dword ptr [rbp], eax
0x1088da:      jmp     0x10877a
0x1088ed:      jmp     0x1087b9
0x1087ec:      xor     eax, eax
0x1087ee:      call    0x10d740
0x10877a:      xor     eax, eax
0x10877c:      call    0x10dea0
0x108788:      xor     eax, eax
0x10878a:      call    0x10d740
0x1087f3:      xor     eax, eax
0x1087f5:      call    0x10a820
0x10878f:      xor     eax, eax
0x108791:      call    0x10a820
0x1087fa:      mov     rdi, r13
0x1087fd:      call    0x10db10
0x108796:      mov     rdi, r13
0x108799:      call    0x10db10
0x108802:      jmp     0x108867
0x10879e:      mov     r12, qword ptr [r13]
0x1087a2:      jmp     0x108867
0x1087a7:      nop     word ptr [rax + rax]
0x108804:      nop     dword ptr [rax]
0x108839:      nop     dword ptr [rax]
0x108879:      nop     dword ptr [rax]

```

```

0x10888c:      nop      dword ptr [rax]
0x1088f2:      nop      word ptr [rax + rax]
0x108921:      nop      dword ptr [rax]
0x108942:      nop      word ptr [rax + rax]
0x108959:      nop      dword ptr [rax]
0x108960:      endbr64
0x108964:      mov      rdi, qword ptr [rip + 0x21f2d]
0x10896b:      push     rbx
0x10896c:      test     rdi, rdi
0x10896f:      je       0x1089ac
0x1089ac:      mov      dword ptr [rip + 0x21ede], 0
0x1089b6:      pop      rbx
0x1089b7:      mov      dword ptr [rip + 0x21ecf], 0
0x1089c1:      ret
0x108971:      mov      eax, dword ptr [rip + 0x21f1d]
0x108977:      test     eax, eax
0x108979:      jle      0x10899c
0x10899c:      call     0x118d10
0x10897b:      xor      ebx, ebx
0x10897d:      nop      dword ptr [rax]
0x108980:      mov      rdi, qword ptr [rdi + rbx*8]
0x108984:      add      rbx, 1
0x108988:      call     0x118d10
0x1089a1:      mov      qword ptr [rip + 0x21eec], 0
0x1089ac:      mov      dword ptr [rip + 0x21ede], 0
0x1089b6:      pop      rbx
0x1089b7:      mov      dword ptr [rip + 0x21ecf], 0
0x1089c1:      ret
0x10898d:      cmp      dword ptr [rip + 0x21f01], ebx
0x108993:      mov      rdi, qword ptr [rip + 0x21efe]
0x10899a:      jg       0x108980
0x108980:      mov      rdi, qword ptr [rdi + rbx*8]
0x108984:      add      rbx, 1
0x108988:      call     0x118d10
0x1089c2:      nop      word ptr cs:[rax + rax]
0x1089cd:      nop      dword ptr [rax]
0x1089d0:      endbr64
0x1089d4:      push     rbp
0x1089d5:      mov      rbp, rdi
0x1089d8:      push     rbx
0x1089d9:      sub      rsp, 8
0x1089dd:      mov      esi, dword ptr [rip + 0x21eb1]
0x1089e3:      mov      rdi, qword ptr [rip + 0x21eae]
0x1089ea:      mov      dword ptr [rip + 0x21ea0], esi
0x1089f0:      add      esi, 1
0x1089f3:      mov      dword ptr [rip + 0x21e9b], esi
0x1089f9:      movsxd   rsi, esi
0x1089fc:      shl      rsi, 3

```

```

0x108a00:    call    0x11d690
0x108a05:    movsxd  rdx, dword ptr [rip + 0x21e84]
0x108a0c:    mov     rdi, rbp
0x108a0f:    mov     qword ptr [rip + 0x21e82], rax
0x108a16:    lea     rbx, [rax + rdx*8]
0x108a1a:    call    0x11d7a0
0x108a1f:    mov     qword ptr [rbx], rax
0x108a22:    add     rsp, 8
0x108a26:    pop     rbx
0x108a27:    pop     rbp
0x108a28:    ret
0x108a29:    nop     dword ptr [rax]
0x108a30:    endbr64
0x108a34:    mov     rdx, rsi
0x108a37:    mov     esi, dword ptr [rip + 0x21e57]
0x108a3d:    mov     eax, dword ptr [rip + 0x21e4d]
0x108a43:    lea     ecx, [rsi - 1]
0x108a46:    cmp     ecx, eax
0x108a48:    jle     0x108a70
0x108a70:    xor     eax, eax
0x108a72:    jmp     0x10a5a0
0x108a4a:    add     eax, 1
0x108a4d:    mov     r8d, 1
0x108a53:    mov     esi, 0x21
0x108a58:    mov     rcx, qword ptr [rip + 0x21e39]
0x108a5f:    mov     dword ptr [rip + 0x21e2b], eax
0x108a65:    cdqe
0x108a67:    mov     rcx, qword ptr [rcx + rax*8]
0x108a6b:    jmp     0x113e30
0x108a77:    nop     word ptr [rax + rax]
0x108a80:    endbr64
0x108a84:    mov     eax, dword ptr [rip + 0x21e06]
0x108a8a:    mov     rdx, rsi
0x108a8d:    test    eax, eax
0x108a8f:    je      0x108ac0
0x108ac0:    xor     eax, eax
0x108ac2:    jmp     0x10a5a0
0x108a91:    sub     eax, 1
0x108a94:    mov     r8d, 1
0x108a9a:    mov     esi, 0x21
0x108a9f:    mov     rcx, qword ptr [rip + 0x21df2]
0x108aa6:    mov     dword ptr [rip + 0x21de4], eax
0x108aac:    cdqe
0x108aae:    mov     rcx, qword ptr [rcx + rax*8]
0x108ab2:    jmp     0x113e30
0x108ab7:    nop     word ptr [rax + rax]
0x108ac7:    nop     word ptr [rax + rax]
0x108ad0:    endbr64

```



```

0x108ad4:    sub     rsp, 8
0x108ad8:    xor     edi, edi
0x108ada:    call    0x10bbc0
0x108adf:    xor     eax, eax
0x108ae1:    call    0x10de40
0x108ae6:    xor     eax, eax
0x108ae8:    add     rsp, 8
0x108aec:    jmp     0x119580
0x108af1:    nop     word ptr cs:[rax + rax]
0x108afc:    nop     dword ptr [rax]
0x108b00:    endbr64
0x108b04:    push    rbx
0x108b05:    xor     eax, eax
0x108b07:    mov     rbx, rdi
0x108b0a:    call    0x109c90
0x108b0f:    cmp     eax, 1
0x108b12:    je      0x108b42
0x108b42:    xor     eax, eax
0x108b44:    call    0x108ad0
0x108b14:    mov     rdi, qword ptr [rip + 0x21d65]
0x108b1b:    mov     rcx, qword ptr [rip + 0x21dc6]
0x108b22:    mov     r8, rbx
0x108b25:    xor     eax, eax
0x108b27:    lea     rdx, [rip + 0x179e8]
0x108b2e:    mov     esi, 1
0x108b33:    call    0x104e60
0x108b49:    jmp     0x108b14
0x108b38:    mov     edi, 1
0x108b3d:    call    0x104e20
0x108b4b:    nop     dword ptr [rax + rax]
0x108b50:    endbr64
0x108b54:    push    r15
0x108b56:    push    r14
0x108b58:    push    r13
0x108b5a:    push    r12
0x108b5c:    mov     r12, rsi
0x108b5f:    push    rbp
0x108b60:    push    rbx
0x108b61:    sub     rsp, 0x68
0x108b65:    mov     r14, qword ptr [rdi + 8]
0x108b69:    mov     qword ptr [rsp + 0x18], rdi
0x108b6e:    mov     qword ptr [rsp + 8], rdx
0x108b73:    mov     rdi, r14
0x108b76:    mov     qword ptr [rsp + 0x10], rcx
0x108b7b:    mov     rax, qword ptr fs:[0x28]
0x108b84:    mov     qword ptr [rsp + 0x58], rax
0x108b89:    xor     eax, eax
0x108b8b:    mov     qword ptr [rsp + 0x48], 0

```

```

0x108b94:    mov     qword ptr [rsp + 0x50], 0
0x108b9d:    call    0x1049b0
0x108ba2:    lea     r13, [rax + 1]
0x108ba6:    mov     rdi, r13
0x108ba9:    call    0x11d670
0x108bae:    mov     qword ptr [r12], rax
0x108bb2:    mov     rbx, rax
0x108bb5:    movzx   eax, byte ptr [r14]
0x108bb9:    test    al, al
0x108bbb:    je      0x10925c
0x10925c:    mov     eax, 1
0x109261:    jmp     0x108d3f
0x108bc1:    mov     dword ptr [rsp + 0x24], 0
0x108bc9:    jmp     0x108be5
0x108d3f:    mov     byte ptr [rbx], 0
0x108d42:    jmp     0x108c4b
0x108be5:    lea     rbp, [r14 + 1]
0x108be9:    cmp     al, 0x25
0x108beb:    jne     0x108bd0
0x108c4b:    mov     rdx, qword ptr [rsp + 0x58]
0x108c50:    sub     rdx, qword ptr fs:[0x28]
0x108c59:    jne     0x10929f
0x108bed:    call    0x104f00
0x108bd0:    mov     byte ptr [rbx], al
0x108bd2:    mov     r14, rbp
0x108bd5:    add     rbx, 1
0x108bd9:    movzx   eax, byte ptr [r14]
0x108bdd:    test    al, al
0x108bdf:    je      0x108d38
0x108c5f:    add     rsp, 0x68
0x108c63:    pop     rbx
0x108c64:    pop     rbp
0x108c65:    pop     r12
0x108c67:    pop     r13
0x108c69:    pop     r14
0x108c6b:    pop     r15
0x108c6d:    ret
0x10929f:    call    0x1049d0
0x108bf2:    movsx   rcx, byte ptr [r14 + 1]
0x108bf7:    mov     r15, qword ptr [rsp + 0x10]
0x108bfc:    mov     rax, qword ptr [rax]
0x108bff:    mov     rdx, rcx
0x108c02:    test    byte ptr [rax + rcx*2 + 1], 2
0x108c07:    lea     eax, [rcx - 0x3f]
0x108c0a:    cmovne  r15, qword ptr [rsp + 8]
0x108c10:    cmp     al, 0x3b
0x108c12:    ja      0x108c38
0x108d38:    mov     eax, dword ptr [rsp + 0x24]

```

```

0x108d3c:    add     eax, 1
0x108d3f:    mov     byte ptr [rbx], 0
0x108d42:    jmp     0x108c4b
0x108c14:    lea     rcx, [rip + 0x174f1]
0x108c1b:    movzx   eax, al
0x108c1e:    movsxd  rax, dword ptr [rcx + rax*4]
0x108c22:    add     rax, rcx
0x108c25:    jmp     rax
0x108c38:    mov     rdi, qword ptr [r12]
0x108c3c:    call    0x118d10
0x108f00:    mov     rdi, r15
0x108f03:    call    0x10f510
0x108e40:    xor     eax, eax
0x108e42:    cmp     dl, 0x69
0x108e45:    mov     edx, dword ptr [rip + 0x21aad]
0x108e4b:    setne   al
0x108e4e:    add     eax, 1
0x108e51:    mov     dword ptr [rsp + 0x24], eax
0x108e55:    test    edx, edx
0x108e57:    je      0x109198
0x108ee0:    mov     rdi, r15
0x108ee3:    call    0x10f510
0x108da0:    mov     rdi, qword ptr [r15 + 0x18]
0x108da4:    call    0x1049b0
0x108f45:    mov     esi, dword ptr [rip + 0x219ad]
0x108f4b:    test    esi, esi
0x108f4d:    jne     0x108c28
0x108c88:    mov     rdi, r15
0x108c8b:    call    0x10f490
0x108df8:    mov     edi, 0x10
0x108dfd:    call    0x11d670
0x108dc8:    mov     rdi, r15
0x108dcb:    call    0x10f4b0
0x108d88:    mov     rdi, r15
0x108d8b:    call    0x10f510
0x108fa8:    mov     ecx, dword ptr [rip + 0x2194a]
0x108fae:    test    ecx, ecx
0x108fb0:    jne     0x108c28
0x108c70:    mov     rdi, r15
0x108c73:    call    0x10f510
0x108f18:    mov     rdi, qword ptr [r15 + 0x18]
0x108f1c:    mov     esi, 0x2f
0x108f21:    lea     r14, [rip + 0x17604]
0x108f28:    call    0x104a20
0x108e90:    mov     rdi, r15
0x108e93:    call    0x10f4d0
0x108d50:    mov     rdi, r15
0x108d53:    call    0x10f510

```

```

0x108c41:    mov     qword ptr [r12], 0
0x108c49:    xor     eax, eax
0x108c4b:    mov     rdx, qword ptr [rsp + 0x58]
0x108c50:    sub     rdx, qword ptr fs:[0x28]
0x108c59:    jne     0x10929f
0x108f08:    test    eax, eax
0x108f0a:    je      0x108c88
0x109198:    mov     qword ptr [rsp + 0x50], 0
0x1091a1:    mov     rdi, r15
0x1091a4:    xor     r14d, r14d
0x1091a7:    call    0x1100c0
0x108e5d:    mov     eax, dword ptr [r15 + 0x70]
0x108e61:    test    eax, eax
0x108e63:    je      0x109198
0x108ee8:    cmp     eax, 2
0x108eeb:    je      0x108c88
0x108da9:    lea     rdi, [rax + 3]
0x108dad:    call    0x11d670
0x108f53:    cmp     byte ptr [r14 + 2], 0x7b
0x108f58:    jne     0x108c38
0x108c28:    mov     dword ptr [rip + 0x21cc6], 0
0x108c32:    nop     word ptr [rax + rax]
0x108c38:    mov     rdi, qword ptr [r12]
0x108c3c:    call    0x118d10
0x108c90:    mov     r14, rax
0x108c93:    mov     rdi, r14
0x108c96:    call    0x1049b0
0x108e02:    mov     rdi, r15
0x108e05:    mov     qword ptr [rsp + 0x50], rax
0x108e0a:    call    0x10f4f0
0x108dd0:    mov     edi, eax
0x108dd2:    mov     r14d, eax
0x108dd5:    call    0x104970
0x108d90:    cmp     eax, 3
0x108d93:    je      0x108c88
0x108fb6:    cmp     byte ptr [r14 + 2], 0x7b
0x108fbb:    jne     0x108c38
0x108c78:    cmp     eax, 1
0x108c7b:    jne     0x108d61
0x108f2d:    mov     edi, 8
0x108f32:    cmp     byte ptr [rax + 1], 0
0x108f36:    je      0x108c9f
0x108e98:    mov     edi, eax
0x108e9a:    mov     r14d, eax
0x108e9d:    call    0x104a10
0x108d58:    cmp     eax, 4
0x108d5b:    je      0x108c88
0x108f10:    jmp     0x108d61

```

```

0x1091ac:    mov     qword ptr [rsp + 0x28], rbx
0x1091b1:    mov     qword ptr [rsp + 0x30], rbp
0x1091b6:    mov     qword ptr [rsp + 0x38], r12
0x1091bb:    jmp     0x109232
0x108e69:    mov     edi, 0x15
0x108e6e:    call    0x11d670
0x108ef1:    jmp     0x108d61
0x108db2:    mov     r8, qword ptr [r15 + 0x18]
0x108db6:    mov     qword ptr [rsp + 0x50], rax
0x108dbb:    mov     rdi, rax
0x108dbe:    jmp     0x108caf
0x108f5e:    add     r14, 3
0x108f62:    mov     esi, 0x7d
0x108f67:    mov     rdi, r14
0x108f6a:    call    0x104a00
0x108c9b:    lea     rdi, [rax + 3]
0x108c9f:    call    0x11d670
0x108e0f:    mov     rdi, qword ptr [rsp + 0x50]
0x108e14:    mov     esi, 1
0x108e19:    lea     rcx, [rip + 0x1771c]
0x108e20:    and     eax, 0xffff
0x108e25:    mov     rdx, -1
0x108e2c:    mov     r8d, eax
0x108e2f:    xor     eax, eax
0x108e31:    call    0x104f10
0x108dda:    test    rax, rax
0x108ddd:    je      0x108eab
0x108d99:    jmp     0x108d61
0x108fc1:    add     r14, 3
0x108fc5:    mov     esi, 0x2c
0x108fca:    mov     rdi, r14
0x108fcd:    call    0x104a00
0x108d61:    mov     rdi, qword ptr [rsp + 0x50]
0x108d66:    test    rdi, rdi
0x108d69:    je      0x108d70
0x108c81:    nop     dword ptr [rax]
0x108c9f:    call    0x11d670
0x108f3c:    lea     r14, [rax + 1]
0x108f40:    jmp     0x108c93
0x108ea2:    test    rax, rax
0x108ea5:    jne     0x108de3
0x109232:    mov     rdi, r15
0x109235:    call    0x1100e0
0x108e73:    movdqa  xmm0, xmmword ptr [rip + 0x17d35]
0x108e7b:    mov     qword ptr [rsp + 0x50], rax
0x108e80:    mov     byte ptr [rax + 0x10], 0
0x108e84:    movups  xmmword ptr [rax], xmm0
0x108e87:    jmp     0x108cc9

```

```

0x108caf:    lea     rcx, [rip + 0x185df]
0x108cb6:    mov     rdx, -1
0x108cbd:    mov     esi, 1
0x108cc2:    xor     eax, eax
0x108cc4:    call    0x104f10
0x108f6f:    mov     rbp, rax
0x108f72:    test    rax, rax
0x108f75:    je      0x108c38
0x108ca4:    mov     r8, r14
0x108ca7:    mov     qword ptr [rsp + 0x50], rax
0x108cac:    mov     rdi, rax
0x108caf:    lea     rcx, [rip + 0x185df]
0x108cb6:    mov     rdx, -1
0x108cbd:    mov     esi, 1
0x108cc2:    xor     eax, eax
0x108cc4:    call    0x104f10
0x108e36:    jmp     0x108cc9
0x108eab:    mov     edi, 0x10
0x108eb0:    call    0x11d670
0x108de3:    mov     rdi, qword ptr [rax]
0x108de6:    call    0x11d7a0
0x108fd2:    mov     rbp, rax
0x108fd5:    test    rax, rax
0x108fd8:    je      0x108c38
0x108d70:    mov     qword ptr [r12], 0
0x108d78:    mov     eax, 0xffffffff
0x108d7d:    jmp     0x108c4b
0x108d6b:    call    0x118d10
0x108c93:    mov     rdi, r14
0x108c96:    call    0x1049b0
0x10923a:    cmp     eax, -1
0x10923d:    jne     0x1091c0
0x108cc9:    mov     byte ptr [rbx], 0
0x108ccc:    mov     rdi, qword ptr [rsp + 0x50]
0x108cd1:    lea     r14, [rbp + 1]
0x108cd5:    test    rdi, rdi
0x108cd8:    je      0x108bd9
0x108f7b:    mov     byte ptr [rax], 0
0x108f7e:    mov     edx, 2
0x108f83:    lea     rsi, [rip + 0x175a8]
0x108f8a:    mov     rdi, r14
0x108f8d:    call    0x108070
0x108eb5:    mov     r8d, r14d
0x108eb8:    mov     esi, 1
0x108ebd:    lea     rcx, [rip + 0x17678]
0x108ec4:    mov     qword ptr [rsp + 0x50], rax
0x108ec9:    mov     rdi, rax
0x108ecc:    mov     rdx, -1

```

```

0x108ed3:    xor     eax, eax
0x108ed5:    call    0x104f10
0x108deb:    mov     qword ptr [rsp + 0x50], rax
0x108df0:    jmp     0x108cc9
0x108fde:    mov     byte ptr [rax], 0
0x108fe1:    mov     rax, qword ptr [rsp + 0x18]
0x108fe6:    lea     r15, [rsp + 0x48]
0x108feb:    mov     rcx, qword ptr [rsp + 0x10]
0x108ff0:    mov     rdx, qword ptr [rsp + 8]
0x108ff5:    mov     rsi, r15
0x108ff8:    mov     dword ptr [rip + 0x218f6], 1
0x109002:    mov     r8, qword ptr [rax + 8]
0x109006:    mov     qword ptr [rax + 8], r14
0x10900a:    mov     rdi, rax
0x10900d:    mov     r14, rax
0x109010:    xor     eax, eax
0x109012:    mov     qword ptr [rsp + 0x28], r8
0x109017:    call    0x108b50
0x10923f:    mov     rbx, qword ptr [rsp + 0x28]
0x109244:    mov     rbp, qword ptr [rsp + 0x30]
0x109249:    mov     r12, qword ptr [rsp + 0x38]
0x10924e:    test    r14, r14
0x109251:    jne     0x108cc9
0x1091c0:    cdqe
0x1091c2:    lea     rcx, [rax + rax*4]
0x1091c6:    mov     rax, qword ptr [r15 + 0x38]
0x1091ca:    shl     rcx, 4
0x1091ce:    mov     rdi, qword ptr [rax + rcx]
0x1091d2:    mov     r12, rcx
0x1091d5:    call    0x1049b0
0x108bd9:    movzx   eax, byte ptr [r14]
0x108bdd:    test    al, al
0x108bdf:    je      0x108d38
0x108cde:    call    0x1049b0
0x108f92:    mov     byte ptr [rbp], 0x7d
0x108f96:    cmp     al, 0x79
0x108f98:    je      0x108cc9
0x108eda:    jmp     0x108cc9
0x10901c:    mov     r8, qword ptr [rsp + 0x28]
0x109021:    mov     dword ptr [rip + 0x218cd], 0
0x10902b:    mov     qword ptr [r14 + 8], r8
0x10902f:    test    eax, eax
0x109031:    jle     0x109290
0x109257:    jmp     0x108d61
0x1091da:    mov     rdi, qword ptr [rsp + 0x50]
0x1091df:    lea     rbp, [rax + r14]
0x1091e3:    lea     rsi, [rbp + 4]
0x1091e7:    lea     rbx, [rbp + 3]

```

```

0x1091eb:    call    0x11d690
0x108ce3:    mov     rdi, qword ptr [r12]
0x108ce7:    add     r13, rax
0x108cea:    mov     rsi, r13
0x108ced:    call    0x11d690
0x108f9e:    jmp     0x108d61
0x109290:    mov     byte ptr [rbp], 0x2c
0x109294:    je      0x108c38
0x109037:    mov     rax, qword ptr [rsp + 0x18]
0x10903c:    mov     rdi, qword ptr [rax]
0x10903f:    call    0x1049b0
0x1091f0:    mov     qword ptr [rsp + 0x50], rax
0x1091f5:    mov     byte ptr [rax + r14], 0x22
0x1091fa:    mov     rax, qword ptr [r15 + 0x38]
0x1091fe:    mov     rsi, qword ptr [rax + r12]
0x109202:    mov     rax, qword ptr [rsp + 0x50]
0x109207:    lea     rdi, [rax + r14 + 1]
0x10920c:    mov     r14, rbx
0x10920f:    call    0x1048b0
0x108cf2:    mov     rsi, qword ptr [rsp + 0x50]
0x108cf7:    mov     qword ptr [r12], rax
0x108cfb:    mov     rdi, rax
0x108cfe:    call    0x104df0
0x10929a:    jmp     0x108d61
0x109044:    mov     r14, qword ptr [rsp + 0x48]
0x109049:    mov     qword ptr [rsp + 0x28], rax
0x10904e:    mov     rdi, r14
0x109051:    call    0x1049b0
0x109214:    mov     rax, qword ptr [rsp + 0x50]
0x109219:    mov     byte ptr [rax + rbp + 1], 0x22
0x10921e:    mov     rax, qword ptr [rsp + 0x50]
0x109223:    mov     byte ptr [rax + rbp + 2], 0x20
0x109228:    mov     rax, qword ptr [rsp + 0x50]
0x10922d:    mov     byte ptr [rax + rbp + 3], 0
0x109232:    mov     rdi, r15
0x109235:    call    0x1100e0
0x108d03:    mov     rbx, qword ptr [r12]
0x108d07:    mov     rdi, rbx
0x108d0a:    call    0x1049b0
0x109056:    mov     rdx, qword ptr [rsp + 0x28]
0x10905b:    lea     rdi, [rdx + rax + 0x11]
0x109060:    call    0x11d670
0x108d0f:    mov     rdi, qword ptr [rsp + 0x50]
0x108d14:    add     rbx, rax
0x108d17:    call    0x118d10
0x109065:    mov     r9, r14
0x109068:    mov     esi, 1
0x10906d:    lea     rcx, [rip + 0x174c1]

```



```

0x109074:    mov     r10, rax
0x109077:    mov     rax, qword ptr [rsp + 0x18]
0x10907c:    mov     rdx, -1
0x109083:    mov     rdi, r10
0x109086:    mov     qword ptr [rsp + 0x28], r10
0x10908b:    mov     r8, qword ptr [rax]
0x10908e:    xor     eax, eax
0x109090:    call    0x104f10
0x108d1c:    mov     qword ptr [rsp + 0x50], 0
0x108d25:    movzx   eax, byte ptr [r14]
0x108d29:    test    al, al
0x108d2b:    jne     0x108be5
0x109095:    mov     rdi, r14
0x109098:    lea     r14, [rbp + 1]
0x10909c:    call    0x118d10
0x108d31:    nop     dword ptr [rax]
0x1090a1:    mov     byte ptr [rbp], 0x2c
0x1090a5:    mov     esi, 0x7d
0x1090aa:    mov     rdi, r14
0x1090ad:    mov     qword ptr [rsp + 0x48], 0
0x1090b6:    call    0x104a00
0x1090bb:    mov     rbp, rax
0x1090be:    test    rax, rax
0x1090c1:    je      0x108c38
0x1090c7:    mov     byte ptr [rax], 0
0x1090ca:    mov     rdi, qword ptr [rsp + 0x28]
0x1090cf:    call    0x1049b0
0x1090d4:    mov     r10, qword ptr [rsp + 0x28]
0x1090d9:    cmp     rax, 0x32
0x1090dd:    jbe     0x1090e4
0x1090e4:    mov     rax, qword ptr [rsp + 0x18]
0x1090e9:    mov     rcx, qword ptr [rsp + 0x10]
0x1090ee:    mov     rsi, r15
0x1090f1:    mov     qword ptr [rsp + 0x30], r10
0x1090f6:    mov     dword ptr [rip + 0x217f8], 1
0x109100:    mov     rdx, qword ptr [rsp + 8]
0x109105:    mov     r8, qword ptr [rax + 8]
0x109109:    mov     qword ptr [rax + 8], r14
0x10910d:    mov     r15, rax
0x109110:    mov     rdi, rax
0x109113:    xor     eax, eax
0x109115:    mov     qword ptr [rsp + 0x28], r8
0x10911a:    call    0x108b50
0x1090df:    mov     byte ptr [r10 + 0x32], 0
0x1090e4:    mov     rax, qword ptr [rsp + 0x18]
0x1090e9:    mov     rcx, qword ptr [rsp + 0x10]
0x1090ee:    mov     rsi, r15
0x1090f1:    mov     qword ptr [rsp + 0x30], r10

```

```

0x1090f6:    mov     dword ptr [rip + 0x217f8], 1
0x109100:    mov     rdx, qword ptr [rsp + 8]
0x109105:    mov     r8, qword ptr [rax + 8]
0x109109:    mov     qword ptr [rax + 8], r14
0x10910d:    mov     r15, rax
0x109110:    mov     rdi, rax
0x109113:    xor     eax, eax
0x109115:    mov     qword ptr [rsp + 0x28], r8
0x10911a:    call    0x108b50
0x10911f:    mov     r8, qword ptr [rsp + 0x28]
0x109124:    mov     r10, qword ptr [rsp + 0x30]
0x109129:    mov     dword ptr [rip + 0x217c5], 0
0x109133:    test    eax, eax
0x109135:    mov     qword ptr [r15 + 8], r8
0x109139:    jle     0x109266
0x109266:    mov     byte ptr [rbp], 0x7d
0x10926a:    mov     rdi, r10
0x10926d:    mov     dword ptr [rsp + 8], eax
0x109271:    call    0x118d10
0x10913f:    mov     rax, qword ptr [rsp + 0x18]
0x109144:    mov     r15, qword ptr [rsp + 0x48]
0x109149:    mov     rdi, r10
0x10914c:    lea     rsi, [rsp + 0x50]
0x109151:    mov     qword ptr [rsp + 0x28], r10
0x109156:    mov     rcx, qword ptr [rax + 0x28]
0x10915a:    mov     rdx, r15
0x10915d:    call    0x1083e0
0x109276:    mov     eax, dword ptr [rsp + 8]
0x10927a:    test    eax, eax
0x10927c:    je      0x108c38
0x109162:    mov     rdi, qword ptr [rsp + 0x28]
0x109167:    mov     r14, rax
0x10916a:    call    0x118d10
0x109282:    jmp     0x108d61
0x10916f:    mov     rdi, r15
0x109172:    call    0x118d10
0x109177:    mov     byte ptr [rbp], 0x7d
0x10917b:    mov     qword ptr [rsp + 0x48], 0
0x109184:    test    r14, r14
0x109187:    jne     0x108cc9
0x10918d:    jmp     0x108d61
0x108bcb:    nop     dword ptr [rax + rax]
0x108c6e:    nop
0x108d47:    nop     word ptr [rax + rax]
0x108d82:    nop     word ptr [rax + rax]
0x108d9b:    nop     dword ptr [rax + rax]
0x108dc3:    nop     dword ptr [rax + rax]
0x108df5:    nop     dword ptr [rax]

```

```

0x108e3b:    nop    dword ptr [rax + rax]
0x108e8c:    nop    dword ptr [rax]
0x108edf:    nop
0x108ef6:    nop    word ptr cs:[rax + rax]
0x108f15:    nop    dword ptr [rax]
0x108fa3:    nop    dword ptr [rax + rax]
0x109192:    nop    word ptr [rax + rax]
0x1091bd:    nop    dword ptr [rax]
0x109287:    nop    word ptr [rax + rax]
0x1092a4:    nop    word ptr cs:[rax + rax]
0x1092af:    nop
0x1092b0:    endbr64
0x1092b4:    push   r13
0x1092b6:    mov    r13, rdx
0x1092b9:    push   r12
0x1092bb:    mov    r12, rsi
0x1092be:    push   rbp
0x1092bf:    mov    rbp, rdi
0x1092c2:    call   0x1047e0
0x1092c7:    test   rax, rax
0x1092ca:    je     0x1092d8
0x1092d8:    test   r12, r12
0x1092db:    je     0x109300
0x1092cc:    pop     rbp
0x1092cd:    pop     r12
0x1092cf:    pop     r13
0x1092d1:    ret
0x109300:    mov    rsi, r13
0x109303:    mov    rdi, rbp
0x109306:    pop     rbp
0x109307:    pop     r12
0x109309:    pop     r13
0x10930b:    jmp     0x118380
0x1092dd:    mov    rdi, r12
0x1092e0:    call   0x1047e0
0x1092e5:    mov    rsi, rax
0x1092e8:    test   rax, rax
0x1092eb:    je     0x109300
0x1092ed:    mov    rdi, rbp
0x1092f0:    pop     rbp
0x1092f1:    pop     r12
0x1092f3:    pop     r13
0x1092f5:    jmp     0x118380
0x1092d2:    nop    word ptr [rax + rax]
0x1092fa:    nop    word ptr [rax + rax]
0x109310:    endbr64
0x109314:    push   r15
0x109316:    push   r14

```

```

0x109318:    push    r13
0x10931a:    push    r12
0x10931c:    push    rbp
0x10931d:    push    rbx
0x10931e:    mov     ebx, edi
0x109320:    sub     rsp, 0xb8
0x109327:    mov     qword ptr [rsp + 0x18], rsi
0x10932c:    mov     rax, qword ptr fs:[0x28]
0x109335:    mov     qword ptr [rsp + 0xa8], rax
0x10933d:    xor     eax, eax
0x10933f:    mov     dword ptr [rsi], 0
0x109345:    cmp     edi, 0x7ff
0x10934b:    jg      0x1095d8
0x109351:    lea     rax, [rsp + 0x20]
0x109356:    lea     r12, [rsp + 0x50]
0x10935b:    mov     qword ptr [rsp + 8], rax
0x109360:    lea     r13, [rip + 0x2089e]
0x109367:    jmp     0x1094df
0x1095d8:    mov     eax, ebx
0x1095da:    mov     rdx, qword ptr [rsp + 0xa8]
0x1095e2:    sub     rdx, qword ptr fs:[0x28]
0x1095eb:    jne     0x1095ff
0x1094df:    mov     rsi, qword ptr [rsp + 8]
0x1094e4:    xor     ecx, ecx
0x1094e6:    mov     edx, 6
0x1094eb:    mov     rdi, r12
0x1094ee:    call    0x117320
0x1095ed:    add     rsp, 0xb8
0x1095f4:    pop     rbx
0x1095f5:    pop     rbp
0x1095f6:    pop     r12
0x1095f8:    pop     r13
0x1095fa:    pop     r14
0x1095fc:    pop     r15
0x1095fe:    ret
0x1095ff:    call    0x1049d0
0x1094f3:    movzx   eax, byte ptr [rsp + 0x50]
0x1094f8:    test    al, al
0x1094fa:    je      0x1095d8
0x109500:    cmp     al, 0x5e
0x109502:    jne     0x109370
0x109508:    mov     dword ptr [rsp + 4], 1
0x109510:    jmp     0x1093a2
0x109370:    mov     rdi, r12
0x109373:    call    0x10bd80
0x1093a2:    mov     edi, 1
0x1093a7:    mov     esi, 0x30
0x1093ac:    call    0x11d740

```

```

0x109378:    test    rax, rax
0x10937b:    je      0x109508
0x1093b1:    mov     rdi, qword ptr [rsp + 0x20]
0x1093b6:    mov     rbp, rax
0x1093b9:    test    rdi, rdi
0x1093bc:    je      0x109560
0x109381:    cmp     byte ptr [rax], 0
0x109384:    je      0x1094d0
0x109560:    mov     rdi, rax
0x109563:    add     ebx, 1
0x109566:    call    0x118d10
0x1093c2:    call    0x11d7a0
0x1094d0:    add     ebx, 1
0x1094d3:    cmp     ebx, 0x800
0x1094d9:    je      0x109580
0x10938a:    mov     edx, 0x50
0x10938f:    mov     rsi, rax
0x109392:    mov     rdi, r12
0x109395:    call    0x104cf0
0x10956b:    cmp     ebx, 0x800
0x109571:    jne     0x1094df
0x1093c7:    mov     edi, 8
0x1093cc:    mov     qword ptr [rbp], rax
0x1093d0:    call    0x119100
0x109580:    mov     eax, 0x800
0x109585:    jmp     0x1095da
0x10939a:    mov     dword ptr [rsp + 4], 0
0x1093a2:    mov     edi, 1
0x1093a7:    mov     esi, 0x30
0x1093ac:    call    0x11d740
0x109577:    nop     word ptr [rax + rax]
0x1093d5:    mov     rdi, qword ptr [rsp + 0x30]
0x1093da:    mov     qword ptr [rbp + 0x28], rax
0x1093de:    test    rdi, rdi
0x1093e1:    je      0x1093ec
0x1095da:    mov     rdx, qword ptr [rsp + 0xa8]
0x1095e2:    sub     rdx, qword ptr fs:[0x28]
0x1095eb:    jne     0x1095ff
0x1093ec:    mov     rdi, qword ptr [rsp + 0x28]
0x1093f1:    test    rdi, rdi
0x1093f4:    je      0x109474
0x1093e3:    call    0x11d7a0
0x109474:    call    0x104f00
0x1093f6:    call    0x11d7a0
0x1093e8:    mov     qword ptr [rbp + 0x10], rax
0x1093ec:    mov     rdi, qword ptr [rsp + 0x28]
0x1093f1:    test    rdi, rdi
0x1093f4:    je      0x109474

```

```

0x109479:    mov     r15, qword ptr [rbp]
0x10947d:    mov     rax, qword ptr [rax]
0x109480:    movsx   rdx, byte ptr [r15]
0x109484:    test    byte ptr [rax + rdx*2 + 1], 2
0x109489:    jne     0x109518
0x1093fb:    mov     qword ptr [rbp + 8], rax
0x1093ff:    mov     r14, qword ptr [rsp + 0x38]
0x109404:    mov     eax, 1
0x109409:    test    r14, r14
0x10940c:    je      0x109421
0x10948f:    mov     rdi, r12
0x109492:    call    0x11d7a0
0x109518:    lea     r14, [rip + 0x1fb01]
0x10951f:    jmp     0x109535
0x109421:    mov     byte ptr [rbp + 0x18], al
0x109424:    mov     rdx, qword ptr [rsp + 0x40]
0x109429:    test    rdx, rdx
0x10942c:    je      0x10944c
0x10940e:    call    0x104ef0
0x109497:    cmp     byte ptr [rbp + 0x1b], 0
0x10949b:    mov     qword ptr [rbp + 0x20], rax
0x10949f:    jne     0x1094ac
0x109535:    mov     rsi, r14
0x109538:    mov     rdi, r15
0x10953b:    call    0x104b30
0x10944c:    mov     rdx, qword ptr [rsp + 0x48]
0x109451:    test    rdx, rdx
0x109454:    je      0x109474
0x10942e:    mov     qword ptr [rsp + 0x10], rdx
0x109433:    call    0x104ef0
0x109413:    movsx   rdx, byte ptr [r14]
0x109417:    mov     rax, qword ptr [rax]
0x10941a:    cmp     dword ptr [rax + rdx*4], 0x79
0x10941e:    sete    al
0x109421:    mov     byte ptr [rbp + 0x18], al
0x109424:    mov     rdx, qword ptr [rsp + 0x40]
0x109429:    test    rdx, rdx
0x10942c:    je      0x10944c
0x1094a1:    cmp     qword ptr [rbp + 8], 0
0x1094a6:    je      0x109590
0x1094ac:    mov     eax, dword ptr [rsp + 4]
0x1094b0:    test    eax, eax
0x1094b2:    je      0x1094c5
0x109540:    test    eax, eax
0x109542:    jne     0x109528
0x109456:    mov     qword ptr [rsp + 0x10], rdx
0x10945b:    call    0x104ef0
0x109438:    mov     rdx, qword ptr [rsp + 0x10]

```

```

0x10943d:    mov     rax, qword ptr [rax]
0x109440:    movsx  rdx, byte ptr [rdx]
0x109444:    cmp     dword ptr [rax + rdx*4], 0x79
0x109448:    sete   byte ptr [rbp + 0x19]
0x10944c:    mov     rdx, qword ptr [rsp + 0x48]
0x109451:    test    rdx, rdx
0x109454:    je      0x109474
0x109590:    cmp     qword ptr [rbp + 0x10], 0
0x109595:    jne     0x1094ac
0x1094c5:    mov     rsi, rbp
0x1094c8:    mov     rdi, r12
0x1094cb:    call    0x10af70
0x1094b4:    mov     rdi, r12
0x1094b7:    call    0x109f00
0x109544:    mov     rdi, r15
0x109547:    call    0x118d10
0x109528:    add     r14, 0x23
0x10952c:    cmp     r13, r14
0x10952f:    je      0x10948f
0x109460:    mov     rdx, qword ptr [rsp + 0x10]
0x109465:    mov     rax, qword ptr [rax]
0x109468:    movsx  rdx, byte ptr [rdx]
0x10946c:    cmp     dword ptr [rax + rdx*4], 0x79
0x109470:    sete   byte ptr [rbp + 0x1a]
0x109474:    call    0x104f00
0x10959b:    jmp     0x1094d0
0x1094bc:    test    rax, rax
0x1094bf:    je      0x1095a0
0x10954c:    mov     qword ptr [rbp], r14
0x109550:    mov     byte ptr [rbp + 0x1b], 1
0x109554:    jmp     0x10948f
0x1095a0:    mov     rcx, qword ptr [rip + 0x21341]
0x1095a7:    mov     rdi, qword ptr [rip + 0x212d2]
0x1095ae:    mov     r8, r12
0x1095b1:    xor     eax, eax
0x1095b3:    lea     rdx, [rip + 0x1720e]
0x1095ba:    mov     esi, 1
0x1095bf:    call    0x104e60
0x1095c4:    mov     rax, qword ptr [rsp + 0x18]
0x1095c9:    add     dword ptr [rax], 1
0x1095cc:    jmp     0x1094d0
0x10936c:    nop     dword ptr [rax]
0x109515:    nop     dword ptr [rax]
0x109521:    nop     dword ptr [rax]
0x109559:    nop     dword ptr [rax]
0x109587:    nop     word ptr [rax + rax]
0x1095d1:    nop     dword ptr [rax]
0x109604:    nop     word ptr cs:[rax + rax]

```

```

0x10960f:      nop
0x109610:      endbr64
0x109614:      sub     rsp, 8
0x109618:      xor     edi, edi
0x10961a:      call    0x109b00
0x10961f:      xor     eax, eax
0x109621:      call    0x10a6c0
0x109626:      mov     rdi, qword ptr [rip + 0x2129b]
0x10962d:      add     rsp, 8
0x109631:      jmp     0x10bb40
0x109636:      nop     word ptr cs:[rax + rax]
0x109640:      endbr64
0x109644:      push    rbp
0x109645:      lea     rbp, [rip + 0x205b9]
0x10964c:      push    rbx
0x10964d:      mov     rdi, rbp
0x109650:      sub     rsp, 0x48
0x109654:      mov     rax, qword ptr fs:[0x28]
0x10965d:      mov     qword ptr [rsp + 0x38], rax
0x109662:      xor     eax, eax
0x109664:      call    0x1049b0
0x109669:      mov     rdi, qword ptr [rip + 0x21240]
0x109670:      mov     rbx, rax
0x109673:      call    0x111100
0x109678:      mov     edx, 0x33
0x10967d:      mov     rsi, rsp
0x109680:      sub     edx, ebx
0x109682:      mov     rdi, rax
0x109685:      call    0x1182b0
0x10968a:      mov     rsi, rbp
0x10968d:      mov     rdi, rax
0x109690:      call    0x104df0
0x109695:      mov     rdi, rax
0x109698:      call    0x10d540
0x10969d:      mov     rax, qword ptr [rsp + 0x38]
0x1096a2:      sub     rax, qword ptr fs:[0x28]
0x1096ab:      jne     0x1096b4
0x1096ad:      add     rsp, 0x48
0x1096b1:      pop     rbx
0x1096b2:      pop     rbp
0x1096b3:      ret
0x1096b4:      call    0x1049d0
0x1096b9:      nop     dword ptr [rax]
0x1096c0:      endbr64
0x1096c4:      sub     rsp, 8
0x1096c8:      mov     r8d, 1
0x1096ce:      mov     rdx, qword ptr [rip + 0x211db]
0x1096d5:      mov     rdi, qword ptr [rip + 0x211cc]

```



```

0x1096dc:    mov     rcx, -1
0x1096e3:    mov     esi, 0x22
0x1096e8:    call    0x113e30
0x1096ed:    mov     r8d, 1
0x1096f3:    mov     rdx, qword ptr [rip + 0x211ae]
0x1096fa:    mov     rdi, qword ptr [rip + 0x211af]
0x109701:    mov     rcx, -1
0x109708:    mov     esi, 0x22
0x10970d:    add     rsp, 8
0x109711:    jmp     0x113e30
0x109716:    nop     word ptr cs:[rax + rax]
0x109720:    endbr64
0x109724:    sub     rsp, 8
0x109728:    mov     rdx, qword ptr [rip + 0x211b9]
0x10972f:    mov     edi, 1
0x109734:    xor     eax, eax
0x109736:    lea     rsi, [rip + 0x170b3]
0x10973d:    call    0x104d20
0x109742:    lea     rdi, [rip + 0x170cf]
0x109749:    call    0x1048d0
0x10974e:    lea     rdi, [rip + 0x170eb]
0x109755:    call    0x1048d0
0x10975a:    lea     rdi, [rip + 0x16dde]
0x109761:    call    0x1048d0
0x109766:    lea     rdi, [rip + 0x170fb]
0x10976d:    call    0x1048d0
0x109772:    lea     rdi, [rip + 0x17117]
0x109779:    call    0x1048d0
0x10977e:    lea     rdi, [rip + 0x1713b]
0x109785:    add     rsp, 8
0x109789:    jmp     0x1048d0
0x10978e:    nop
0x109790:    cmp     dword ptr [rip + 0x21ba9], 1
0x109797:    push    rbx
0x109798:    mov     ebx, edi
0x10979a:    je      0x109830
0x109830:    movzx   eax, byte ptr [rip + 0x21b19]
0x109837:    shr     al, 7
0x10983a:    movzx   eax, al
0x10983d:    cmp     eax, edi
0x10983f:    je      0x109809
0x1097a0:    cmp     ebx, 1
0x1097a3:    je      0x10984a
0x109809:    pop     rbx
0x10980a:    ret
0x109841:    cmp     ebx, 1
0x109844:    jne     0x1097a9
0x10984a:    mov     rdi, qword ptr [rip + 0x20517]

```

```

0x109851:    test    rdi, rdi
0x109854:    je      0x109824
0x1097a9:    mov     rdi, qword ptr [rip + 0x20598]
0x1097b0:    test    rdi, rdi
0x1097b3:    je      0x1097c6
0x109824:    movzx   eax, byte ptr [rip + 0x21b25]
0x10982b:    jmp     0x1097f1
0x109856:    lea     rdx, [rip + 0x213]
0x10985d:    mov     esi, 1
0x109862:    call    0x104820
0x1097c6:    movzx   edx, byte ptr [rip + 0x21b83]
0x1097cd:    mov     dword ptr [rip + 0x21b75], 0
0x1097d7:    mov     dword ptr [rip + 0x21b67], 0
0x1097e1:    mov     eax, edx
0x1097e3:    and     eax, 0x7f
0x1097e6:    and     edx, 0x40
0x1097e9:    mov     byte ptr [rip + 0x21b61], al
0x1097ef:    jne     0x109810
0x1097b5:    lea     rdx, [rip + 0x2b4]
0x1097bc:    mov     esi, 1
0x1097c1:    call    0x104820
0x1097f1:    mov     dword ptr [rip + 0x21b45], 1
0x1097fb:    and     eax, 0x7f
0x1097fe:    shl     ebx, 7
0x109801:    or      eax, ebx
0x109803:    mov     byte ptr [rip + 0x21b47], al
0x109809:    pop     rbx
0x10980a:    ret
0x109867:    movzx   eax, byte ptr [rip + 0x21ae2]
0x10986e:    jmp     0x1097f1
0x109810:    mov     dword ptr [rip + 0x21b2a], 0
0x10981a:    mov     edi, 1
0x10981f:    call    0x109870
0x10980b:    nop     dword ptr [rax + rax]
0x10982d:    nop     dword ptr [rax]
0x109870:    cmp     dword ptr [rip + 0x21acd], 1
0x109877:    push    rbx
0x109878:    mov     ebx, edi
0x10987a:    je      0x1098e0
0x1098e0:    movzx   eax, byte ptr [rip + 0x21a69]
0x1098e7:    sar     eax, 6
0x1098ea:    and     eax, 1
0x1098ed:    cmp     eax, edi
0x1098ef:    je      0x1098de
0x10987c:    cmp     ebx, 1
0x10987f:    je      0x1098f6
0x1098de:    pop     rbx
0x1098df:    ret

```

```

0x1098f1:    cmp     ebx, 1
0x1098f4:    jne     0x109881
0x1098f6:    mov     rdi, qword ptr [rip + 0x2048b]
0x1098fd:    test    rdi, rdi
0x109900:    je      0x109934
0x109881:    mov     rdi, qword ptr [rip + 0x204c0]
0x109888:    test    rdi, rdi
0x10988b:    je      0x10989e
0x109934:    movzx   eax, byte ptr [rip + 0x21a15]
0x10993b:    jmp     0x1098c6
0x109902:    lea     rdx, [rip + 0x167]
0x109909:    mov     esi, 1
0x10990e:    call    0x104820
0x10989e:    movzx   eax, byte ptr [rip + 0x21aab]
0x1098a5:    mov     dword ptr [rip + 0x21a9d], 0
0x1098af:    mov     dword ptr [rip + 0x21a8f], 0
0x1098b9:    and     eax, 0xffffffffbf
0x1098bc:    mov     byte ptr [rip + 0x21a8e], al
0x1098c2:    test    al, al
0x1098c4:    js      0x109920
0x10988d:    lea     rdx, [rip + 0x1dc]
0x109894:    mov     esi, 1
0x109899:    call    0x104820
0x1098c6:    mov     dword ptr [rip + 0x21a74], 1
0x1098d0:    and     eax, 0xffffffffbf
0x1098d3:    shl     ebx, 6
0x1098d6:    or      eax, ebx
0x1098d8:    mov     byte ptr [rip + 0x21a72], al
0x1098de:    pop     rbx
0x1098df:    ret
0x109913:    movzx   eax, byte ptr [rip + 0x21a36]
0x10991a:    jmp     0x1098c6
0x109920:    mov     dword ptr [rip + 0x21a16], 0
0x10992a:    mov     edi, 1
0x10992f:    call    0x109790
0x10991c:    nop     dword ptr [rax]
0x10993d:    nop     dword ptr [rax]
0x109940:    push    r12
0x109942:    mov     edx, 5
0x109947:    lea     rsi, [rip + 0x17293]
0x10994e:    mov     r12d, 1
0x109954:    push    rbp
0x109955:    mov     rbp, rdi
0x109958:    sub     rsp, 8
0x10995c:    call    0x104890
0x109961:    test    eax, eax
0x109963:    je      0x1099d3
0x1099d3:    add     rsp, 8

```

```

0x1099d7:    mov     eax, r12d
0x1099da:    pop     rbp
0x1099db:    pop     r12
0x1099dd:    ret
0x109965:    mov     edx, 4
0x10996a:    lea     rsi, [rip + 0x1725f]
0x109971:    mov     rdi, rbp
0x109974:    call    0x104890
0x109979:    test    eax, eax
0x10997b:    je      0x1099d3
0x10997d:    mov     edx, 9
0x109982:    lea     rsi, [rip + 0x1724c]
0x109989:    mov     rdi, rbp
0x10998c:    call    0x104890
0x109991:    test    eax, eax
0x109993:    je      0x1099d3
0x109995:    lea     rsi, [rip + 0x17243]
0x10999c:    mov     rdi, rbp
0x10999f:    call    0x104b30
0x1099a4:    test    eax, eax
0x1099a6:    je      0x1099d3
0x1099a8:    lea     rsi, [rip + 0x17238]
0x1099af:    mov     rdi, rbp
0x1099b2:    call    0x104b30
0x1099b7:    test    eax, eax
0x1099b9:    je      0x1099d3
0x1099bb:    lea     rsi, [rip + 0x1722b]
0x1099c2:    mov     rdi, rbp
0x1099c5:    xor     r12d, r12d
0x1099c8:    call    0x104b30
0x1099cd:    test    eax, eax
0x1099cf:    sete    r12b
0x1099d3:    add     rsp, 8
0x1099d7:    mov     eax, r12d
0x1099da:    pop     rbp
0x1099db:    pop     r12
0x1099dd:    ret
0x1099de:    nop
0x1099e0:    push    r13
0x1099e2:    push    r12
0x1099e4:    mov     r12, qword ptr [rip + 0x20f8d]
0x1099eb:    push    rbp
0x1099ec:    test    r12, r12
0x1099ef:    je      0x109a58
0x109a58:    mov     r12, -1
0x109a5f:    pop     rbp
0x109a60:    mov     rax, r12
0x109a63:    pop     r12

```

```

0x109a65:    pop     r13
0x109a67:    ret
0x1099f1:    mov     r13, rdi
0x1099f4:    jmp     0x109a13
0x109a13:    mov     rbp, qword ptr [r12]
0x109a17:    mov     rdi, r13
0x109a1a:    mov     rsi, rbp
0x109a1d:    call    0x104b30
0x109a22:    test    eax, eax
0x109a24:    jne     0x109a00
0x109a26:    mov     rax, r12
0x109a29:    pop     rbp
0x109a2a:    pop     r12
0x109a2c:    pop     r13
0x109a2e:    ret
0x109a00:    js      0x109a30
0x109a30:    mov     rdi, r13
0x109a33:    call    0x1049b0
0x109a02:    mov     r12, qword ptr [r12 + 8]
0x109a07:    mov     qword ptr [rip + 0x20f6a], r12
0x109a0e:    test    r12, r12
0x109a11:    je      0x109a58
0x109a38:    mov     rsi, rbp
0x109a3b:    mov     rdi, r13
0x109a3e:    mov     rdx, rax
0x109a41:    call    0x104890
0x109a46:    pop     rbp
0x109a47:    neg     eax
0x109a49:    sbb     r12, r12
0x109a4c:    mov     rax, r12
0x109a4f:    pop     r12
0x109a51:    pop     r13
0x109a53:    ret
0x1099f6:    nop     word ptr cs:[rax + rax]
0x109a2f:    nop
0x109a54:    nop     dword ptr [rax]
0x109a68:    nop     dword ptr [rax + rax]
0x109a70:    endbr64
0x109a74:    movsxd  rdx, dword ptr [rip + 0x2148d]
0x109a7b:    push    r12
0x109a7d:    push    rbp
0x109a7e:    mov     ebp, edi
0x109a80:    push    rbx
0x109a81:    cmp     edx, 0x400
0x109a87:    je      0x109ab0
0x109ab0:    xor     ebx, ebx
0x109ab2:    lea     r12, [rip + 0x21467]
0x109ab9:    jmp     0x109acc

```

```

0x109a89:    lea     eax, [rdx + 1]
0x109a8c:    lea     r12, [rip + 0x2148d]
0x109a93:    mov     byte ptr [r12 + rdx], bpl
0x109a97:    pop     rbx
0x109a98:    mov     dword ptr [rip + 0x2146a], eax
0x109a9e:    pop     rbp
0x109a9f:    mov     eax, 1
0x109aa4:    pop     r12
0x109aa6:    ret
0x109acc:    sub     edx, ebx
0x109ace:    movsxd  rsi, ebx
0x109ad1:    mov     edi, 1
0x109ad6:    add     rsi, r12
0x109ad9:    movsxd  rdx, edx
0x109adc:    call    0x118da0
0x109ae1:    test    eax, eax
0x109ae3:    jns     0x109ac0
0x109ae5:    mov     eax, 1
0x109aea:    xor     edx, edx
0x109aec:    jmp     0x109a93
0x109ac0:    mov     edx, dword ptr [rip + 0x21442]
0x109ac6:    add     ebx, eax
0x109ac8:    cmp     edx, ebx
0x109aca:    jle     0x109ae5
0x109a93:    mov     byte ptr [r12 + rdx], bpl
0x109a97:    pop     rbx
0x109a98:    mov     dword ptr [rip + 0x2146a], eax
0x109a9e:    pop     rbp
0x109a9f:    mov     eax, 1
0x109aa4:    pop     r12
0x109aa6:    ret
0x109aa7:    nop     word ptr [rax + rax]
0x109abb:    nop     dword ptr [rax + rax]
0x109aee:    nop
0x109af0:    endbr64
0x109af4:    mov     dword ptr [rip + 0x21862], edi
0x109afa:    ret
0x109afb:    nop     dword ptr [rax + rax]
0x109b00:    endbr64
0x109b04:    push    r12
0x109b06:    push    rbp
0x109b07:    push    rbx
0x109b08:    mov     ebx, edi
0x109b0a:    sub     rsp, 0x50
0x109b0e:    mov     rax, qword ptr fs:[0x28]
0x109b17:    mov     qword ptr [rsp + 0x48], rax
0x109b1c:    xor     eax, eax
0x109b1e:    cmp     edi, 1

```

```

0x109b21:    je      0x109b60
0x109b60:    mov     rax, qword ptr [rip + 0x21889]
0x109b67:    mov     rdx, qword ptr [rip + 0x21852]
0x109b6e:    lea     r12, [rip + 0x2180b]
0x109b75:    mov     esi, 1
0x109b7a:    movdqa  xmm0, xmmword ptr [rip + 0x2183e]
0x109b82:    movdqa  xmm1, xmmword ptr [rip + 0x21846]
0x109b8a:    mov     edi, 1
0x109b8f:    mov     qword ptr [rip + 0x2181a], rax
0x109b96:    mov     eax, dword ptr [rip + 0x2185c]
0x109b9c:    movaps  xmmword ptr [rip + 0x217dd], xmm0
0x109ba3:    movdqa  xmm2, xmmword ptr [rip + 0x21835]
0x109bab:    mov     dword ptr [rip + 0x21807], eax
0x109bb1:    movabs  rax, 0xffffffffffffa3c
0x109bbb:    and     rax, rdx
0x109bbe:    mov     rdx, r12
0x109bc1:    movaps  xmmword ptr [rip + 0x217c8], xmm1
0x109bc8:    mov     qword ptr [rip + 0x217b1], rax
0x109bcf:    mov     eax, dword ptr [rip + 0x217b7]
0x109bd5:    movaps  xmmword ptr [rip + 0x217c4], xmm2
0x109bdc:    and     eax, 0xffffffff5
0x109bdf:    mov     byte ptr [rip + 0x217bb], 0
0x109be6:    mov     dword ptr [rip + 0x217a1], 7
0x109bf0:    or      al, 0x81
0x109bf2:    mov     dword ptr [rip + 0x21794], eax
0x109bf8:    mov     eax, 0x100
0x109bfd:    mov     word ptr [rip + 0x21792], ax
0x109c04:    mov     qword ptr [rip + 0x2178a], 0x1a0000
0x109c0f:    call    0x104d70
0x109b23:    lea     rdx, [rip + 0x21896]
0x109b2a:    mov     esi, 1
0x109b2f:    mov     edi, 1
0x109b34:    call    0x104d70
0x109c14:    mov     rdi, r12
0x109c17:    mov     r12, rsp
0x109c1a:    call    0x104d10
0x109b39:    mov     dword ptr [rip + 0x20e19], ebx
0x109b3f:    mov     rax, qword ptr [rsp + 0x48]
0x109b44:    sub     rax, qword ptr fs:[0x28]
0x109b4d:    jne     0x109c7e
0x109c1f:    mov     edx, 1
0x109c24:    mov     esi, 0x540a
0x109c29:    mov     edi, 1
0x109c2e:    mov     dword ptr [rip + 0x20cfc], eax
0x109c34:    xor     eax, eax
0x109c36:    call    0x104a70
0x109b53:    add     rsp, 0x50
0x109b57:    pop     rbx

```

```

0x109b58:    pop     rbp
0x109b59:    pop     r12
0x109b5b:    ret
0x109c7e:    call    0x1049d0
0x109c3b:    mov     esi, 1
0x109c40:    mov     edi, 1
0x109c45:    call    0x104db0
0x109c4a:    mov     ebp, dword ptr [rip + 0x200d4]
0x109c50:    mov     rsi, r12
0x109c53:    mov     edi, 1
0x109c58:    call    0x104d60
0x109c5d:    mov     rdx, r12
0x109c60:    mov     esi, 1
0x109c65:    mov     edi, 1
0x109c6a:    mov     byte ptr [rsp + 0x11], bpl
0x109c6f:    mov     byte ptr [rsp + 0x12], 0
0x109c74:    call    0x104d70
0x109c79:    jmp     0x109b39
0x109b5c:    nop     dword ptr [rax]
0x109c83:    nop     word ptr cs:[rax + rax]
0x109c8e:    nop
0x109c90:    endbr64
0x109c94:    mov     eax, dword ptr [rip + 0x20cbe]
0x109c9a:    ret
0x109c9b:    nop     dword ptr [rax + rax]
0x109ca0:    endbr64
0x109ca4:    push    r12
0x109ca6:    push    rbx
0x109ca7:    mov     ebx, edi
0x109ca9:    mov     edi, 1
0x109cae:    sub     rsp, 0x58
0x109cb2:    mov     rax, qword ptr fs:[0x28]
0x109cbb:    mov     qword ptr [rsp + 0x48], rax
0x109cc0:    xor     eax, eax
0x109cc2:    mov     r12, rsp
0x109cc5:    mov     rsi, r12
0x109cc8:    call    0x104d60
0x109ccd:    mov     rdx, r12
0x109cd0:    mov     esi, 1
0x109cd5:    mov     edi, 1
0x109cda:    mov     byte ptr [rsp + 0x11], bl
0x109cde:    mov     byte ptr [rsp + 0x12], 0
0x109ce3:    call    0x104d70
0x109ce8:    mov     dword ptr [rip + 0x20036], ebx
0x109cee:    mov     rax, qword ptr [rsp + 0x48]
0x109cf3:    sub     rax, qword ptr fs:[0x28]
0x109cfc:    jne     0x109d06
0x109cfe:    add     rsp, 0x58

```



```

0x109d02:    pop     rbx
0x109d03:    pop     r12
0x109d05:    ret
0x109d06:    call    0x1049d0
0x109d0b:    nop     dword ptr [rax + rax]
0x109d10:    endbr64
0x109d14:    mov     edx, dword ptr [rip + 0x211ee]
0x109d1a:    test    edx, edx
0x109d1c:    jle     0x109d66
0x109d66:    mov     dword ptr [rip + 0x21198], 0
0x109d70:    ret
0x109d1e:    push    rbp
0x109d1f:    xor     ebp, ebp
0x109d21:    push    rbx
0x109d22:    lea     rbx, [rip + 0x211f7]
0x109d29:    sub     rsp, 8
0x109d2d:    jmp     0x109d3c
0x109d3c:    sub     edx, ebp
0x109d3e:    movsxd  rsi, ebp
0x109d41:    mov     edi, 1
0x109d46:    add     rsi, rbx
0x109d49:    movsxd  rdx, edx
0x109d4c:    call    0x118da0
0x109d51:    test    eax, eax
0x109d53:    jns     0x109d30
0x109d55:    mov     dword ptr [rip + 0x211a9], 0
0x109d5f:    add     rsp, 8
0x109d63:    pop     rbx
0x109d64:    pop     rbp
0x109d65:    ret
0x109d30:    mov     edx, dword ptr [rip + 0x211d2]
0x109d36:    add     ebp, eax
0x109d38:    cmp     edx, ebp
0x109d3a:    jle     0x109d55
0x109d2f:    nop
0x109d71:    nop     word ptr cs:[rax + rax]
0x109d7c:    nop     dword ptr [rax]
0x109d80:    endbr64
0x109d84:    push    rbp
0x109d85:    lea     rdx, [rip - 0x31c]
0x109d8c:    push    rbx
0x109d8d:    sub     rsp, 8
0x109d91:    mov     esi, dword ptr [rip + 0x20c05]
0x109d97:    mov     rdi, qword ptr [rip + 0x2004a]
0x109d9e:    call    0x104820
0x109da3:    mov     edx, dword ptr [rip + 0x2115f]
0x109da9:    test    edx, edx
0x109dab:    jle     0x109de5

```

```

0x109de5:    mov     dword ptr [rip + 0x21119], 0
0x109def:    add     rsp, 8
0x109df3:    pop     rbx
0x109df4:    pop     rbp
0x109df5:    ret
0x109dad:    xor     ebp, ebp
0x109daf:    lea     rbx, [rip + 0x2116a]
0x109db6:    jmp     0x109dcc
0x109dcc:    sub     edx, ebp
0x109dce:    movsxd  rsi, ebp
0x109dd1:    mov     edi, 1
0x109dd6:    add     rsi, rbx
0x109dd9:    movsxd  rdx, edx
0x109ddc:    call    0x118da0
0x109de1:    test    eax, eax
0x109de3:    jns     0x109dc0
0x109dc0:    mov     edx, dword ptr [rip + 0x21142]
0x109dc6:    add     ebp, eax
0x109dc8:    cmp     edx, ebp
0x109dca:    jle     0x109de5
0x109db8:    nop     dword ptr [rax + rax]
0x109df6:    nop     word ptr cs:[rax + rax]
0x109e00:    endbr64
0x109e04:    push    rbp
0x109e05:    lea     rdx, [rip - 0x39c]
0x109e0c:    push    rbx
0x109e0d:    sub     rsp, 8
0x109e11:    mov     eax, dword ptr [rip + 0x20b85]
0x109e17:    mov     rdi, qword ptr [rip + 0x2010a]
0x109e1e:    lea     esi, [rax - 1]
0x109e21:    call    0x104820
0x109e26:    mov     edx, dword ptr [rip + 0x210dc]
0x109e2c:    test    edx, edx
0x109e2e:    jle     0x109e65
0x109e65:    mov     dword ptr [rip + 0x21099], 0
0x109e6f:    add     rsp, 8
0x109e73:    pop     rbx
0x109e74:    pop     rbp
0x109e75:    ret
0x109e30:    xor     ebp, ebp
0x109e32:    lea     rbx, [rip + 0x210e7]
0x109e39:    jmp     0x109e4c
0x109e4c:    sub     edx, ebp
0x109e4e:    movsxd  rsi, ebp
0x109e51:    mov     edi, 1
0x109e56:    add     rsi, rbx
0x109e59:    movsxd  rdx, edx
0x109e5c:    call    0x118da0

```

```

0x109e61:    test    eax, eax
0x109e63:    jns     0x109e40
0x109e40:    mov     edx, dword ptr [rip + 0x210c2]
0x109e46:    add     ebp, eax
0x109e48:    cmp     edx, ebp
0x109e4a:    jle     0x109e65
0x109e3b:    nop     dword ptr [rax + rax]
0x109e76:    nop     word ptr cs:[rax + rax]
0x109e80:    endbr64
0x109e84:    push    rbp
0x109e85:    lea     rdx, [rip - 0x41c]
0x109e8c:    push    rbx
0x109e8d:    sub     rsp, 8
0x109e91:    mov     eax, dword ptr [rip + 0x20b05]
0x109e97:    mov     rdi, qword ptr [rip + 0x200aa]
0x109e9e:    lea     esi, [rax - 1]
0x109ea1:    call    0x104820
0x109ea6:    mov     edx, dword ptr [rip + 0x2105c]
0x109eac:    test    edx, edx
0x109eae:    jle     0x109ee5
0x109ee5:    mov     dword ptr [rip + 0x21019], 0
0x109eef:    add     rsp, 8
0x109ef3:    pop     rbx
0x109ef4:    pop     rbp
0x109ef5:    ret
0x109eb0:    xor     ebp, ebp
0x109eb2:    lea     rbx, [rip + 0x21067]
0x109eb9:    jmp     0x109ecc
0x109ecc:    sub     edx, ebp
0x109ece:    movsxd  rsi, ebp
0x109ed1:    mov     edi, 1
0x109ed6:    add     rsi, rbx
0x109ed9:    movsxd  rdx, edx
0x109edc:    call    0x118da0
0x109ee1:    test    eax, eax
0x109ee3:    jns     0x109ec0
0x109ec0:    mov     edx, dword ptr [rip + 0x21042]
0x109ec6:    add     ebp, eax
0x109ec8:    cmp     edx, ebp
0x109eca:    jle     0x109ee5
0x109ebb:    nop     dword ptr [rax + rax]
0x109ef6:    nop     word ptr cs:[rax + rax]
0x109f00:    endbr64
0x109f04:    mov     edx, dword ptr [rip + 0x214f2]
0x109f0a:    movzx   eax, byte ptr [rdi]
0x109f0d:    push    r14
0x109f0f:    push    r13
0x109f11:    mov     r13, rdi

```

```

0x109f14:    push    r12
0x109f16:    push    rbp
0x109f17:    push    rbx
0x109f18:    test    edx, edx
0x109f1a:    jne     0x109fc8
0x109f20:    cmp     al, 0x5e
0x109f22:    jne     0x109fb8
0x109fc8:    test    al, al
0x109fca:    jne     0x109f28
0x109f28:    lea     rbx, [r13 + 1]
0x109f2c:    mov     rbp, r13
0x109f2f:    lea     r14, [rip + 0x1fd6a]
0x109f36:    jmp     0x109f56
0x109fb8:    pop     rbx
0x109fb9:    xor     eax, eax
0x109fbb:    pop     rbp
0x109fbc:    pop     r12
0x109fbe:    pop     r13
0x109fc0:    pop     r14
0x109fc2:    ret
0x109fd0:    mov     rdx, rdi
0x109fd3:    jmp     0x109fa6
0x109f56:    cmp     al, 0x5e
0x109f58:    jne     0x109f40
0x109fa6:    mov     byte ptr [rdx], 0
0x109fa9:    mov     rax, r13
0x109fac:    pop     rbx
0x109fad:    pop     rbp
0x109fae:    pop     r12
0x109fb0:    pop     r13
0x109fb2:    pop     r14
0x109fb4:    ret
0x109f5a:    movzx   r12d, byte ptr [rbp + 1]
0x109f5f:    test    r12b, r12b
0x109f62:    je      0x109fb8
0x109f40:    add     rbp, 1
0x109f44:    mov     byte ptr [rbx - 1], al
0x109f47:    mov     rdx, rbx
0x109f4a:    add     rbx, 1
0x109f4e:    movzx   eax, byte ptr [rbp]
0x109f52:    test    al, al
0x109f54:    je      0x109fa6
0x109f64:    call    0x1047c0
0x109f69:    mov     r8, rax
0x109f6c:    movzx   eax, r12b
0x109f70:    mov     rdx, qword ptr [r8]
0x109f73:    mov     eax, dword ptr [rdx + rax*4]
0x109f76:    cmp     eax, 0x47

```

```

0x109f79:    je      0x109fb8
0x109f7b:    cmp     eax, 0x5a
0x109f7e:    je      0x109fb8
0x109f80:    and     r12d, 0x7f
0x109f84:    add     rbp, 2
0x109f88:    mov     rdx, rbx
0x109f8b:    add     rbx, 1
0x109f8f:    sub     r12d, 0x20
0x109f93:    movsxd  r12, r12d
0x109f96:    movzx   eax, byte ptr [r14 + r12]
0x109f9b:    mov     byte ptr [rbx - 2], al
0x109f9e:    movzx   eax, byte ptr [rbp]
0x109fa2:    test    al, al
0x109fa4:    jne     0x109f56
0x109f38:    nop     dword ptr [rax + rax]
0x109fb5:    nop     dword ptr [rax]
0x109fc3:    nop     dword ptr [rax + rax]
0x109fd5:    nop     word ptr cs:[rax + rax]
0x109fe0:    endbr64
0x109fe4:    push    r15
0x109fe6:    push    r14
0x109fe8:    mov     r14, rdi
0x109feb:    push    r13
0x109fed:    lea     r13, [rip + 0x16c12]
0x109ff4:    push    r12
0x109ff6:    lea     r12, [rip + 0x209c3]
0x109ffd:    push    rbp
0x109ffe:    lea     rbp, [rip + 0x16bef]
0x10a005:    push    rbx
0x10a006:    mov     rbx, rdi
0x10a009:    sub     rsp, 0x18
0x10a00d:    mov     rax, qword ptr fs:[0x28]
0x10a016:    mov     qword ptr [rsp + 8], rax
0x10a01b:    xor     eax, eax
0x10a01d:    mov     byte ptr [rip + 0x2099c], 0
0x10a024:    movzx   r15d, byte ptr [rdi]
0x10a028:    test    r15b, r15b
0x10a02b:    jne     0x10a0c2
0x10a031:    jmp     0x10a0f0
0x10a0c2:    cmp     r15b, 0x1b
0x10a0c6:    jne     0x10a040
0x10a0f0:    mov     rax, qword ptr [rsp + 8]
0x10a0f5:    sub     rax, qword ptr fs:[0x28]
0x10a0fe:    jne     0x10a199
0x10a0cc:    mov     edx, 0x12c
0x10a0d1:    mov     rsi, rbp
0x10a0d4:    mov     rdi, r12
0x10a0d7:    add     r14, 1

```

```

0x10a0db:      call    0x1047d0
0x10a040:      cmp     r15b, 0x20
0x10a044:      je      0x10a148
0x10a104:      add     rsp, 0x18
0x10a108:      mov     rax, r12
0x10a10b:      pop     rbx
0x10a10c:      pop     rbp
0x10a10d:      pop     r12
0x10a10f:      pop     r13
0x10a111:      pop     r14
0x10a113:      pop     r15
0x10a115:      ret
0x10a199:      call    0x1049d0
0x10a0e0:      cmp     byte ptr [r14], 0
0x10a0e4:      jne     0x10a0a5
0x10a148:      mov     edx, 0x12c
0x10a14d:      mov     rsi, r13
0x10a150:      mov     rdi, r12
0x10a153:      call    0x1047d0
0x10a04a:      cmp     r15b, 0x7f
0x10a04e:      je      0x10a160
0x10a0a5:      cmp     rbx, r14
0x10a0a8:      je      0x10a0be
0x10a0e6:      nop     word ptr cs:[rax + rax]
0x10a158:      jmp     0x10a09b
0x10a160:      mov     edx, 0x12c
0x10a165:      lea     rsi, [rip + 0x16a8f]
0x10a16c:      mov     rdi, r12
0x10a16f:      call    0x1047d0
0x10a054:      cmp     r15b, 0xff
0x10a058:      je      0x10a180
0x10a0be:      movzx   r15d, byte ptr [r14]
0x10a0c2:      cmp     r15b, 0x1b
0x10a0c6:      jne     0x10a040
0x10a0aa:      mov     edx, 0x12c
0x10a0af:      lea     rsi, [rip + 0x1668f]
0x10a0b6:      mov     rdi, r12
0x10a0b9:      call    0x1047d0
0x10a09b:      add     r14, 1
0x10a09f:      cmp     byte ptr [r14], 0
0x10a0a3:      je      0x10a0f0
0x10a174:      jmp     0x10a09b
0x10a180:      mov     edx, 0x12c
0x10a185:      lea     rsi, [rip + 0x16a79]
0x10a18c:      mov     rdi, r12
0x10a18f:      call    0x1047d0
0x10a05e:      call    0x104f00
0x10a194:      jmp     0x10a09b

```

```

0x10a063:    mov     r8, rax
0x10a066:    movzx   eax, r15b
0x10a06a:    mov     rsi, qword ptr [r8]
0x10a06d:    test    byte ptr [rsi + rax*2], 2
0x10a071:    je      0x10a120
0x10a120:    lea     rsi, [rsp + 5]
0x10a125:    mov     edx, 0x12c
0x10a12a:    mov     rdi, r12
0x10a12d:    mov     byte ptr [rsp + 5], r15b
0x10a132:    mov     byte ptr [rsp + 6], 0
0x10a137:    call    0x1047d0
0x10a077:    lea     edx, [r15 + 0x40]
0x10a07b:    lea     rsi, [rsp + 5]
0x10a080:    mov     rdi, r12
0x10a083:    mov     byte ptr [rsp + 5], 0x5e
0x10a088:    mov     byte ptr [rsp + 6], dl
0x10a08c:    mov     edx, 0x12c
0x10a091:    mov     byte ptr [rsp + 7], 0
0x10a096:    call    0x1047d0
0x10a13c:    jmp     0x10a09b
0x10a036:    nop     word ptr cs:[rax + rax]
0x10a116:    nop     word ptr cs:[rax + rax]
0x10a141:    nop     dword ptr [rax]
0x10a15d:    nop     dword ptr [rax]
0x10a179:    nop     dword ptr [rax]
0x10a19e:    nop
0x10a1a0:    endbr64
0x10a1a4:    mov     eax, dword ptr [rip + 0x211ae]
0x10a1aa:    mov     r8, rdi
0x10a1ad:    mov     edi, dword ptr [rip + 0x207e5]
0x10a1b3:    push    r13
0x10a1b5:    push    r12
0x10a1b7:    xor     r12d, r12d
0x10a1ba:    lea     edx, [rax + rsi]
0x10a1bd:    push    rbx
0x10a1be:    mov     dword ptr [rip + 0x21194], edx
0x10a1c4:    cmp     edi, eax
0x10a1c6:    jle     0x10a21c
0x10a21c:    mov     eax, r12d
0x10a21f:    pop     rbx
0x10a220:    pop     r12
0x10a222:    pop     r13
0x10a224:    ret
0x10a1c8:    mov     ecx, dword ptr [rip + 0x21186]
0x10a1ce:    cmp     ecx, dword ptr [rip + 0x207c8]
0x10a1d4:    jge     0x10a21c
0x10a1d6:    mov     r12d, edi
0x10a1d9:    sub     r12d, eax

```

```

0x10a1dc:      cmp     edx, edi
0x10a1de:      cmovle  r12d, esi
0x10a1e2:      imul    edi, ecx
0x10a1e5:      mov     rsi, r8
0x10a1e8:      movsxd  r13, r12d
0x10a1eb:      lea     ebx, [rdi + rax]
0x10a1ee:      mov     rdi, qword ptr [rip + 0x21143]
0x10a1f5:      mov     rdx, r13
0x10a1f8:      movsxd  rbx, ebx
0x10a1fb:      add     rdi, rbx
0x10a1fe:      call    0x104bb0
0x10a203:      add     rbx, qword ptr [rip + 0x21126]
0x10a20a:      movzx   esi, byte ptr [rip + 0x21140]
0x10a211:      mov     rdx, r13
0x10a214:      mov     rdi, rbx
0x10a217:      call    0x104a50
0x10a225:      nop     word ptr cs:[rax + rax]
0x10a230:      endbr64
0x10a234:      push    r13
0x10a236:      push    r12
0x10a238:      xor     r12d, r12d
0x10a23b:      push    rbx
0x10a23c:      sub     rsp, 0x10
0x10a240:      mov     rax, qword ptr fs:[0x28]
0x10a249:      mov     qword ptr [rsp + 8], rax
0x10a24e:      xor     eax, eax
0x10a250:      mov     eax, dword ptr [rip + 0x21102]
0x10a256:      mov     byte ptr [rsp + 7], dil
0x10a25b:      mov     edi, dword ptr [rip + 0x20737]
0x10a261:      lea     edx, [rax + 1]
0x10a264:      mov     dword ptr [rip + 0x210ee], edx
0x10a26a:      cmp     eax, edi
0x10a26c:      jge     0x10a2c2
0x10a26e:      mov     ecx, dword ptr [rip + 0x210e0]
0x10a274:      cmp     ecx, dword ptr [rip + 0x20722]
0x10a27a:      jge     0x10a2c2
0x10a2c2:      mov     rax, qword ptr [rsp + 8]
0x10a2c7:      sub     rax, qword ptr fs:[0x28]
0x10a2d0:      jne     0x10a2ee
0x10a27c:      cmp     edx, edi
0x10a27e:      jle     0x10a2e0
0x10a2d2:      add     rsp, 0x10
0x10a2d6:      mov     eax, r12d
0x10a2d9:      pop     rbx
0x10a2da:      pop     r12
0x10a2dc:      pop     r13
0x10a2de:      ret
0x10a2ee:      call    0x1049d0

```



```

0x10a2e0:    mov     r13d, 1
0x10a2e6:    mov     r12d, 1
0x10a2ec:    jmp     0x10a289
0x10a280:    mov     r12d, edi
0x10a283:    sub     r12d, eax
0x10a286:    movsxd  r13, r12d
0x10a289:    imul    edi, ecx
0x10a28c:    lea     rsi, [rsp + 7]
0x10a291:    mov     rdx, r13
0x10a294:    lea     ebx, [rdi + rax]
0x10a297:    mov     rdi, qword ptr [rip + 0x2109a]
0x10a29e:    movsxd  rbx, ebx
0x10a2a1:    add     rdi, rbx
0x10a2a4:    call    0x104bb0
0x10a289:    imul    edi, ecx
0x10a28c:    lea     rsi, [rsp + 7]
0x10a291:    mov     rdx, r13
0x10a294:    lea     ebx, [rdi + rax]
0x10a297:    mov     rdi, qword ptr [rip + 0x2109a]
0x10a29e:    movsxd  rbx, ebx
0x10a2a1:    add     rdi, rbx
0x10a2a4:    call    0x104bb0
0x10a2a9:    add     rbx, qword ptr [rip + 0x21080]
0x10a2b0:    movzx   esi, byte ptr [rip + 0x2109a]
0x10a2b7:    mov     rdx, r13
0x10a2ba:    mov     rdi, rbx
0x10a2bd:    call    0x104a50
0x10a2df:    nop
0x10a2f3:    nop     word ptr cs:[rax + rax]
0x10a2fe:    nop
0x10a300:    endbr64
0x10a304:    push    rbp
0x10a305:    lea     rdx, [rip - 0x89c]
0x10a30c:    push    rbx
0x10a30d:    sub     rsp, 8
0x10a311:    mov     esi, dword ptr [rip + 0x20685]
0x10a317:    mov     rdi, qword ptr [rip + 0x1faca]
0x10a31e:    call    0x104820
0x10a323:    mov     edx, dword ptr [rip + 0x20bdf]
0x10a329:    test    edx, edx
0x10a32b:    jle     0x10a365
0x10a365:    mov     edx, dword ptr [rip + 0x20631]
0x10a36b:    xor     esi, esi
0x10a36d:    imul    edx, dword ptr [rip + 0x20624]
0x10a374:    mov     dword ptr [rip + 0x20b8a], 0
0x10a37e:    mov     rdi, qword ptr [rip + 0x20fb3]
0x10a385:    movsxd  rdx, edx
0x10a388:    call    0x104a50

```

```

0x10a32d:    xor     ebp, ebp
0x10a32f:    lea     rbx, [rip + 0x20bea]
0x10a336:    jmp     0x10a34c
0x10a38d:    mov     edx, dword ptr [rip + 0x20609]
0x10a393:    imul    edx, dword ptr [rip + 0x205fe]
0x10a39a:    xor     esi, esi
0x10a39c:    mov     rdi, qword ptr [rip + 0x20f8d]
0x10a3a3:    movsxd  rdx, edx
0x10a3a6:    call    0x104a50
0x10a34c:    sub     edx, ebp
0x10a34e:    movsxd  rsi, ebp
0x10a351:    mov     edi, 1
0x10a356:    add     rsi, rbx
0x10a359:    movsxd  rdx, edx
0x10a35c:    call    0x118da0
0x10a3ab:    mov     edx, dword ptr [rip + 0x205eb]
0x10a3b1:    imul    edx, dword ptr [rip + 0x205e0]
0x10a3b8:    xor     esi, esi
0x10a3ba:    mov     rdi, qword ptr [rip + 0x20f67]
0x10a3c1:    movsxd  rdx, edx
0x10a3c4:    call    0x104a50
0x10a361:    test    eax, eax
0x10a363:    jns     0x10a340
0x10a3c9:    mov     edx, dword ptr [rip + 0x205cd]
0x10a3cf:    imul    edx, dword ptr [rip + 0x205c2]
0x10a3d6:    xor     esi, esi
0x10a3d8:    mov     rdi, qword ptr [rip + 0x20f41]
0x10a3df:    movsxd  rdx, edx
0x10a3e2:    call    0x104a50
0x10a340:    mov     edx, dword ptr [rip + 0x20bc2]
0x10a346:    add     ebp, eax
0x10a348:    cmp     edx, ebp
0x10a34a:    jle     0x10a365
0x10a3e7:    mov     dword ptr [rip + 0x20f67], 0
0x10a3f1:    mov     dword ptr [rip + 0x20f59], 0
0x10a3fb:    add     rsp, 8
0x10a3ff:    pop     rbx
0x10a400:    pop     rbp
0x10a401:    ret
0x10a338:    nop     dword ptr [rax + rax]
0x10a402:    nop     word ptr cs:[rax + rax]
0x10a40d:    nop     dword ptr [rax]
0x10a410:    endbr64
0x10a414:    sub     rsp, 8
0x10a418:    mov     edx, dword ptr [rip + 0x2057e]
0x10a41e:    imul    edx, dword ptr [rip + 0x20573]
0x10a425:    mov     esi, 0x20
0x10a42a:    mov     rdi, qword ptr [rip + 0x20f07]

```

```

0x10a431:    movsxd  rdx, edx
0x10a434:    call    0x104a50
0x10a439:    mov     edx, dword ptr [rip + 0x2055d]
0x10a43f:    imul    edx, dword ptr [rip + 0x20552]
0x10a446:    movzx   esi, byte ptr [rip + 0x20f04]
0x10a44d:    mov     rdi, qword ptr [rip + 0x20edc]
0x10a454:    movsxd  rdx, edx
0x10a457:    call    0x104a50
0x10a45c:    mov     edx, dword ptr [rip + 0x2053a]
0x10a462:    imul    edx, dword ptr [rip + 0x2052f]
0x10a469:    xor     esi, esi
0x10a46b:    mov     rdi, qword ptr [rip + 0x20eb6]
0x10a472:    add     rsp, 8
0x10a476:    movsxd  rdx, edx
0x10a479:    jmp     0x104a50
0x10a47e:    nop
0x10a480:    endbr64
0x10a484:    mov     edx, dword ptr [rip + 0x20512]
0x10a48a:    imul    edx, dword ptr [rip + 0x20507]
0x10a491:    xor     esi, esi
0x10a493:    mov     rdi, qword ptr [rip + 0x20e8e]
0x10a49a:    movsxd  rdx, edx
0x10a49d:    jmp     0x104a50
0x10a4a2:    nop     word ptr cs:[rax + rax]
0x10a4ad:    nop     dword ptr [rax]
0x10a4b0:    endbr64
0x10a4b4:    mov     dword ptr [rip + 0x20e9a], edi
0x10a4ba:    mov     dword ptr [rip + 0x20e98], esi
0x10a4c0:    ret
0x10a4c1:    nop     word ptr cs:[rax + rax]
0x10a4cc:    nop     dword ptr [rax]
0x10a4d0:    endbr64
0x10a4d4:    mov     eax, dword ptr [rip + 0x20e7a]
0x10a4da:    mov     dword ptr [rdi], eax
0x10a4dc:    mov     eax, dword ptr [rip + 0x20e76]
0x10a4e2:    mov     dword ptr [rsi], eax
0x10a4e4:    ret
0x10a4e5:    nop     word ptr cs:[rax + rax]
0x10a4f0:    endbr64
0x10a4f4:    movzx   eax, byte ptr [rip + 0x20e56]
0x10a4fb:    and     edi, 7
0x10a4fe:    and     eax, 0xffffffff8
0x10a501:    or      eax, edi
0x10a503:    mov     byte ptr [rip + 0x20e48], al
0x10a509:    ret
0x10a50a:    nop     word ptr [rax + rax]
0x10a510:    endbr64
0x10a514:    movzx   eax, byte ptr [rip + 0x20e36]

```

```

0x10a51b:    shl     edi, 3
0x10a51e:    and     edi, 0x38
0x10a521:    and     eax, 0xffffffffc7
0x10a524:    or      eax, edi
0x10a526:    mov     byte ptr [rip + 0x20e25], al
0x10a52c:    ret
0x10a52d:    nop     dword ptr [rax]
0x10a530:    endbr64
0x10a534:    movzx   eax, byte ptr [rip + 0x20e16]
0x10a53b:    shl     edi, 6
0x10a53e:    and     edi, 0x40
0x10a541:    and     eax, 0xffffffffbf
0x10a544:    or      eax, edi
0x10a546:    mov     byte ptr [rip + 0x20e05], al
0x10a54c:    ret
0x10a54d:    nop     dword ptr [rax]
0x10a550:    endbr64
0x10a554:    movzx   eax, byte ptr [rip + 0x20df6]
0x10a55b:    shl     edi, 7
0x10a55e:    and     eax, 0x7f
0x10a561:    or      eax, edi
0x10a563:    mov     byte ptr [rip + 0x20de8], al
0x10a569:    ret
0x10a56a:    nop     word ptr [rax + rax]
0x10a570:    endbr64
0x10a574:    shl     edi, 6
0x10a577:    and     esi, 7
0x10a57a:    shl     edx, 3
0x10a57d:    mov     eax, edi
0x10a57f:    movzx   edi, byte ptr [rip + 0x20dcb]
0x10a586:    and     edx, 0x38
0x10a589:    and     eax, 0x40
0x10a58c:    and     edi, 0xffffffffb8
0x10a58f:    or      edi, eax
0x10a591:    or      edi, esi
0x10a593:    and     edi, 0xffffffffc7
0x10a596:    or      edi, edx
0x10a598:    mov     byte ptr [rip + 0x20db2], dil
0x10a59f:    ret
0x10a5a0:    endbr64
0x10a5a4:    push    rbp
0x10a5a5:    push    rbx
0x10a5a6:    sub     rsp, 8
0x10a5aa:    movsxd  rax, dword ptr [rip + 0x20957]
0x10a5b1:    cmp     eax, 0x400
0x10a5b6:    je      0x10a610
0x10a610:    xor     ebp, ebp
0x10a612:    lea     rbx, [rip + 0x20907]

```

```

0x10a619:    jmp     0x10a62c
0x10a5b8:    lea     edx, [rax + 1]
0x10a5bb:    lea     rbx, [rip + 0x2095e]
0x10a5c2:    mov     dword ptr [rip + 0x20940], edx
0x10a5c8:    mov     byte ptr [rbx + rax], 7
0x10a5cc:    test    edx, edx
0x10a5ce:    jle     0x10a5fd
0x10a62c:    sub     eax, ebp
0x10a62e:    movsxd  rsi, ebp
0x10a631:    mov     edi, 1
0x10a636:    add     rsi, rbx
0x10a639:    movsxd  rdx, eax
0x10a63c:    call    0x118da0
0x10a5fd:    mov     dword ptr [rip + 0x20901], 0
0x10a607:    add     rsp, 8
0x10a60b:    pop     rbx
0x10a60c:    pop     rbp
0x10a60d:    ret
0x10a5d0:    xor     ebp, ebp
0x10a5d2:    jmp     0x10a5e4
0x10a641:    test    eax, eax
0x10a643:    jns     0x10a620
0x10a5e4:    sub     edx, ebp
0x10a5e6:    movsxd  rsi, ebp
0x10a5e9:    mov     edi, 1
0x10a5ee:    add     rsi, rbx
0x10a5f1:    movsxd  rdx, edx
0x10a5f4:    call    0x118da0
0x10a645:    mov     dword ptr [rip + 0x208b9], 1
0x10a64f:    mov     edx, 1
0x10a654:    mov     byte ptr [rip + 0x208c5], 7
0x10a65b:    jmp     0x10a5d0
0x10a620:    add     ebp, eax
0x10a622:    mov     eax, dword ptr [rip + 0x208e0]
0x10a628:    cmp     eax, ebp
0x10a62a:    jle     0x10a645
0x10a5f9:    test    eax, eax
0x10a5fb:    jns     0x10a5d8
0x10a5d8:    mov     edx, dword ptr [rip + 0x2092a]
0x10a5de:    add     ebp, eax
0x10a5e0:    cmp     edx, ebp
0x10a5e2:    jle     0x10a5fd
0x10a5d4:    nop     dword ptr [rax]
0x10a60e:    nop
0x10a61b:    nop     dword ptr [rax + rax]
0x10a660:    endbr64
0x10a664:    test    edi, edi
0x10a666:    je      0x10a688

```

```

0x10a688:    mov     rdi, qword ptr [rip + 0x1f719]
0x10a68f:    test    rdi, rdi
0x10a692:    jne     0x10a674
0x10a668:    mov     rdi, qword ptr [rip + 0x1f759]
0x10a66f:    test    rdi, rdi
0x10a672:    je      0x10a694
0x10a694:    ret
0x10a674:    lea     rdx, [rip - 0xc0b]
0x10a67b:    mov     esi, 1
0x10a680:    jmp     0x104820
0x10a685:    nop     dword ptr [rax]
0x10a695:    nop     word ptr cs:[rax + rax]
0x10a6a0:    endbr64
0x10a6a4:    movzx   eax, byte ptr [rip + 0x20ca6]
0x10a6ab:    mov     byte ptr [rdi], al
0x10a6ad:    ret
0x10a6ae:    nop
0x10a6b0:    endbr64
0x10a6b4:    movzx   eax, byte ptr [rdi]
0x10a6b7:    mov     byte ptr [rip + 0x20c94], al
0x10a6bd:    ret
0x10a6be:    nop
0x10a6c0:    endbr64
0x10a6c4:    push    rbp
0x10a6c5:    push    rbx
0x10a6c6:    sub     rsp, 8
0x10a6ca:    cmp     dword ptr [rip + 0x1f58f], 1
0x10a6d1:    je      0x10a720
0x10a720:    mov     edx, dword ptr [rip + 0x207e2]
0x10a726:    cmp     edx, 0x3fb
0x10a72c:    jg      0x10a750
0x10a6d3:    mov     rdi, qword ptr [rip + 0x1f66e]
0x10a6da:    test    rdi, rdi
0x10a6dd:    je      0x10a6f0
0x10a72e:    movsxd  rax, edx
0x10a731:    lea     rcx, [rip + 0x207e8]
0x10a738:    add     edx, 4
0x10a73b:    add     rax, rcx
0x10a73e:    mov     ecx, dword ptr [rip + 0x1f5db]
0x10a744:    mov     dword ptr [rip + 0x207be], edx
0x10a74a:    mov     dword ptr [rax], ecx
0x10a74c:    jmp     0x10a6d3
0x10a750:    xor     ebx, ebx
0x10a752:    lea     rbp, [rip + 0x207c7]
0x10a759:    jmp     0x10a76c
0x10a6f0:    mov     dword ptr [rip + 0x20c52], 0
0x10a6fa:    mov     dword ptr [rip + 0x20c44], 0
0x10a704:    mov     dword ptr [rip + 0x20c36], 0

```

```

0x10a70e:    mov     dword ptr [rip + 0x20c28], 0
0x10a718:    add     rsp, 8
0x10a71c:    pop     rbx
0x10a71d:    pop     rbp
0x10a71e:    ret
0x10a6df:    lea     rdx, [rip - 0xc76]
0x10a6e6:    mov     esi, 1
0x10a6eb:    call    0x104820
0x10a76c:    sub     edx, ebx
0x10a76e:    movsxd  rsi, ebx
0x10a771:    mov     edi, 1
0x10a776:    add     rsi, rbp
0x10a779:    movsxd  rdx, edx
0x10a77c:    call    0x118da0
0x10a781:    test    eax, eax
0x10a783:    jns     0x10a760
0x10a785:    mov     edx, 4
0x10a78a:    lea     rax, [rip + 0x2078f]
0x10a791:    jmp     0x10a73e
0x10a760:    mov     edx, dword ptr [rip + 0x207a2]
0x10a766:    add     ebx, eax
0x10a768:    cmp     edx, ebx
0x10a76a:    jle     0x10a785
0x10a73e:    mov     ecx, dword ptr [rip + 0x1f5db]
0x10a744:    mov     dword ptr [rip + 0x207be], edx
0x10a74a:    mov     dword ptr [rax], ecx
0x10a74c:    jmp     0x10a6d3
0x10a71f:    nop
0x10a74e:    nop
0x10a75b:    nop     dword ptr [rax + rax]
0x10a793:    nop     word ptr cs:[rax + rax]
0x10a79e:    nop
0x10a7a0:    push    r12
0x10a7a2:    mov     r12d, edi
0x10a7a5:    push    rbp
0x10a7a6:    mov     ebp, esi
0x10a7a8:    sub     rsp, 8
0x10a7ac:    mov     eax, dword ptr [rip + 0x1f71e]
0x10a7b2:    test    eax, eax
0x10a7b4:    je      0x10a800
0x10a800:    xor     eax, eax
0x10a802:    call    0x10a6c0
0x10a7b6:    mov     eax, ebp
0x10a7b8:    mov     esi, dword ptr [rip + 0x201da]
0x10a7be:    mov     edx, dword ptr [rip + 0x201d8]
0x10a7c4:    or      eax, r12d
0x10a7c7:    mov     rdi, qword ptr [rip + 0x1f63a]
0x10a7ce:    js      0x10a7d4

```

```

0x10a807:    jmp     0x10a7b6
0x10a7d4:    sub     edx, 1
0x10a7d7:    sub     esi, 1
0x10a7da:    call    0x104960
0x10a7d0:    cmp     ebp, esi
0x10a7d2:    jl      0x10a810
0x10a7df:    add     rsp, 8
0x10a7e3:    mov     esi, 1
0x10a7e8:    lea     rdx, [rip - 0xd7f]
0x10a7ef:    pop     rbp
0x10a7f0:    mov     rdi, rax
0x10a7f3:    pop     r12
0x10a7f5:    jmp     0x104820
0x10a810:    cmp     r12d, edx
0x10a813:    jge     0x10a7d4
0x10a815:    mov     edx, r12d
0x10a818:    mov     esi, ebp
0x10a81a:    jmp     0x10a7da
0x10a7da:    call    0x104960
0x10a7fa:    nop     word ptr [rax + rax]
0x10a809:    nop     dword ptr [rax]
0x10a81c:    nop     dword ptr [rax]
0x10a820:    endbr64
0x10a824:    push    r15
0x10a826:    push    r14
0x10a828:    push    r13
0x10a82a:    push    r12
0x10a82c:    push    rbp
0x10a82d:    push    rbx
0x10a82e:    sub     rsp, 0x38
0x10a832:    mov     ebx, dword ptr [rip + 0x20160]
0x10a838:    imul    ebx, dword ptr [rip + 0x2015d]
0x10a83f:    mov     rax, qword ptr fs:[0x28]
0x10a848:    mov     qword ptr [rsp + 0x28], rax
0x10a84d:    xor     eax, eax
0x10a84f:    mov     rdi, qword ptr [rip + 0x1f552]
0x10a856:    cmp     dword ptr [rip + 0x20aff], 1
0x10a85d:    sbb     ebx, 0
0x10a860:    test    rdi, rdi
0x10a863:    je      0x10a876
0x10a876:    test    ebx, ebx
0x10a878:    jle     0x10ab70
0x10a865:    lea     rdx, [rip - 0xd7c]
0x10a86c:    mov     esi, 1
0x10a871:    call    0x104820
0x10ab70:    mov     edi, dword ptr [rip + 0x207de]
0x10ab76:    mov     esi, dword ptr [rip + 0x207dc]
0x10ab7c:    call    0x10a7a0

```



```

0x10a87e:    movsxd  rax, ebx
0x10a881:    mov     dword ptr [rsp + 4], 0xffffffff
0x10a889:    xor     ebp, ebp
0x10a88b:    mov     r13d, 0xffffffff
0x10a891:    mov     qword ptr [rsp + 8], rax
0x10a896:    lea     r12, [rip + 0x20683]
0x10a89d:    jmp     0x10a9ce
0x10ab81:    mov     rdi, qword ptr [rip + 0x1f240]
0x10ab88:    test    rdi, rdi
0x10ab8b:    je      0x10ab9e
0x10a9ce:    mov     rdx, qword ptr [rip + 0x20953]
0x10a9d5:    mov     rcx, qword ptr [rip + 0x2095c]
0x10a9dc:    mov     eax, ebp
0x10a9de:    movzx   ebx, byte ptr [rdx + rbp]
0x10a9e2:    cmp     byte ptr [rcx + rbp], bl
0x10a9e5:    jne     0x10a8a8
0x10ab9e:    mov     edx, dword ptr [rip + 0x20364]
0x10aba4:    test    edx, edx
0x10aba6:    jne     0x10ac98
0x10ab8d:    lea     rdx, [rip - 0x1124]
0x10ab94:    mov     esi, 1
0x10ab99:    call    0x104820
0x10a9eb:    mov     rdx, qword ptr [rip + 0x2092e]
0x10a9f2:    mov     rcx, qword ptr [rip + 0x20937]
0x10a9f9:    movzx   edi, byte ptr [rdx + rbp]
0x10a9fd:    cmp     byte ptr [rcx + rbp], dil
0x10aa01:    je      0x10a9bf
0x10a8a8:    cdq
0x10a8a9:    idiv    dword ptr [rip + 0x200e9]
0x10a8af:    mov     r14d, edx
0x10a8b2:    cmp     edx, r13d
0x10a8b5:    jne     0x10aa20
0x10abac:    mov     edx, dword ptr [rip + 0x1fde6]
0x10abb2:    imul    edx, dword ptr [rip + 0x1fde3]
0x10abb9:    mov     rsi, qword ptr [rip + 0x20778]
0x10abc0:    mov     rdi, qword ptr [rip + 0x20761]
0x10abc7:    movsxd  rdx, edx
0x10abca:    call    0x104bb0
0x10ac98:    mov     ebx, 0
0x10ac9d:    lea     rbp, [rip + 0x2027c]
0x10aca4:    jg      0x10acbc
0x10a9bf:    add     rbp, 1
0x10a9c3:    cmp     rbp, qword ptr [rsp + 8]
0x10a9c8:    je      0x10ab70
0x10aa03:    cdq
0x10aa04:    idiv    dword ptr [rip + 0x1ff8e]
0x10aa0a:    mov     r14d, edx
0x10aa0d:    cmp     edx, r13d

```

```

0x10aa10:    je      0x10a8bb
0x10a8bb:    cmp     eax, dword ptr [rsp + 4]
0x10a8bf:    jne     0x10aa20
0x10aa20:    mov     esi, r14d
0x10aa23:    mov     edi, eax
0x10aa25:    mov     dword ptr [rsp + 4], eax
0x10aa29:    mov     r13d, r14d
0x10aa2c:    call    0x10a7a0
0x10abcf:    mov     edx, dword ptr [rip + 0x1fdc3]
0x10abd5:    imul    edx, dword ptr [rip + 0x1fdc0]
0x10abdc:    mov     rdi, qword ptr [rip + 0x2073d]
0x10abe3:    mov     rsi, qword ptr [rip + 0x20746]
0x10abea:    movsxd  rdx, edx
0x10abed:    mov     rax, qword ptr [rsp + 0x28]
0x10abf2:    sub     rax, qword ptr fs:[0x28]
0x10abfb:    jne     0x10ace4
0x10aca6:    jmp     0x10acd5
0x10acbc:    sub     edx, ebx
0x10acbe:    movsxd  rsi, ebx
0x10acc1:    mov     edi, 1
0x10acc6:    add     rsi, rbp
0x10acc9:    movsxd  rdx, edx
0x10accc:    call    0x118da0
0x10aa16:    nop     word ptr cs:[rax + rax]
0x10a8c5:    mov     rax, qword ptr [rip + 0x20a64]
0x10a8cc:    movzx   ebx, byte ptr [rax + rbp]
0x10a8d0:    mov     edi, ebx
0x10a8d2:    mov     r14d, ebx
0x10a8d5:    sar     edi, 6
0x10a8d8:    and     r14d, 7
0x10a8dc:    and     edi, 1
0x10a8df:    call    0x109870
0x10aa31:    jmp     0x10a8c5
0x10ac01:    add     rsp, 0x38
0x10ac05:    pop     rbx
0x10ac06:    pop     rbp
0x10ac07:    pop     r12
0x10ac09:    pop     r13
0x10ac0b:    pop     r14
0x10ac0d:    pop     r15
0x10ac0f:    jmp     0x104bb0
0x10ace4:    call    0x1049d0
0x10acd5:    mov     dword ptr [rip + 0x20229], 0
0x10acdf:    jmp     0x10abac
0x10acd1:    test    eax, eax
0x10acd3:    jns     0x10acb0
0x10a8e4:    mov     eax, ebx
0x10a8e6:    and     eax, 7

```

```

0x10a8e9:    cmp     dword ptr [rip + 0x20a5c], 1
0x10a8f0:    je      0x10aa40
0x10acb0:    mov     edx, dword ptr [rip + 0x20252]
0x10acb6:    add     ebx, eax
0x10acb8:    cmp     edx, ebx
0x10acba:    jle     0x10acd5
0x10aa40:    movzx   edx, byte ptr [rip + 0x20909]
0x10aa47:    and     edx, 7
0x10aa4a:    cmp     eax, edx
0x10aa4c:    je      0x10a92e
0x10a8f6:    cmp     dword ptr [rip + 0x1f363], 1
0x10a8fd:    je      0x10aa5f
0x10a92e:    sar     ebx, 3
0x10a931:    and     ebx, 7
0x10a934:    cmp     dword ptr [rip + 0x20a0d], 1
0x10a93b:    je      0x10aab0
0x10aa52:    cmp     dword ptr [rip + 0x1f207], 1
0x10aa59:    jne     0x10a903
0x10aa5f:    mov     eax, dword ptr [rip + 0x164b4]
0x10aa65:    mov     edx, dword ptr [rip + 0x2049d]
0x10aa6b:    mov     dword ptr [rsp + 0x10], eax
0x10aa6f:    movzx   eax, byte ptr [rip + 0x164a7]
0x10aa76:    mov     byte ptr [rsp + 0x14], al
0x10aa7a:    lea     eax, [r14 + 0x30]
0x10aa7e:    mov     byte ptr [rsp + 0x13], al
0x10aa82:    cmp     edx, 0x3fa
0x10aa88:    jg      0x10ac18
0x10a903:    xor     edi, edi
0x10a905:    cmp     eax, 7
0x10a908:    setne   dil
0x10a90c:    call    0x109790
0x10aab0:    movzx   eax, byte ptr [rip + 0x20899]
0x10aab7:    mov     edx, dword ptr [rip + 0x2044b]
0x10aabd:    sar     eax, 3
0x10aac0:    and     eax, 7
0x10aac3:    cmp     ebx, eax
0x10aac5:    je      0x10a980
0x10a941:    cmp     dword ptr [rip + 0x1f318], 1
0x10a948:    je      0x10aad8
0x10aa8e:    movsxd  rax, edx
0x10aa91:    add     edx, 5
0x10aa94:    add     rax, r12
0x10aa97:    mov     ecx, dword ptr [rsp + 0x10]
0x10aa9b:    mov     dword ptr [rip + 0x20467], edx
0x10aaa1:    mov     dword ptr [rax], ecx
0x10aaa3:    movzx   ecx, byte ptr [rsp + 0x14]
0x10aaa8:    mov     byte ptr [rax + 4], cl
0x10aaab:    jmp     0x10a911

```

```

0x10ac18:    xor     r15d, r15d
0x10ac1b:    jmp     0x10ac2e
0x10a911:    movzx   eax, byte ptr [rip + 0x20a38]
0x10a918:    mov     dword ptr [rip + 0x20a2a], 1
0x10a922:    and     eax, 0xffffffff8
0x10a925:    or      eax, r14d
0x10a928:    mov     byte ptr [rip + 0x20a22], al
0x10a92e:    sar     ebx, 3
0x10a931:    and     ebx, 7
0x10a934:    cmp     dword ptr [rip + 0x20a0d], 1
0x10a93b:    je      0x10aab0
0x10a980:    mov     rax, qword ptr [rip + 0x209b1]
0x10a987:    movzx   ebx, byte ptr [rax + rbp]
0x10a98b:    cmp     edx, 0x400
0x10a991:    je      0x10ab30
0x10aacb:    cmp     dword ptr [rip + 0x1f18e], 1
0x10aad2:    jne     0x10a94e
0x10aad8:    mov     eax, dword ptr [rip + 0x16436]
0x10aade:    mov     edx, dword ptr [rip + 0x20424]
0x10aae4:    mov     dword ptr [rsp + 0x10], eax
0x10aae8:    movzx   eax, byte ptr [rip + 0x16429]
0x10aaef:    mov     byte ptr [rsp + 0x14], al
0x10aaf3:    lea     eax, [rbx + 0x30]
0x10aaf6:    mov     byte ptr [rsp + 0x13], al
0x10aafa:    cmp     edx, 0x3fa
0x10ab00:    jg      0x10ac58
0x10a94e:    xor     edi, edi
0x10a950:    test    ebx, ebx
0x10a952:    setne   dil
0x10a956:    call    0x109790
0x10ac2e:    sub     edx, r15d
0x10ac31:    movsxd  rsi, r15d
0x10ac34:    mov     edi, 1
0x10ac39:    add     rsi, r12
0x10ac3c:    movsxd  rdx, edx
0x10ac3f:    call    0x118da0
0x10ab30:    xor     r14d, r14d
0x10ab33:    jmp     0x10ab46
0x10a997:    lea     eax, [rdx + 1]
0x10a99a:    movsxd  rdx, edx
0x10a99d:    add     r13d, 1
0x10a9a1:    cmp     dword ptr [rip + 0x1fff0], r13d
0x10a9a8:    mov     dword ptr [rip + 0x2055a], eax
0x10a9ae:    mov     byte ptr [r12 + rdx], bl
0x10a9b2:    jne     0x10a9bf
0x10ab06:    movsxd  rax, edx
0x10ab09:    add     edx, 5
0x10ab0c:    add     rax, r12

```

```

0x10ab0f:    mov     ecx, dword ptr [rsp + 0x10]
0x10ab13:    mov     dword ptr [rip + 0x203ef], edx
0x10ab19:    mov     dword ptr [rax], ecx
0x10ab1b:    movzx   ecx, byte ptr [rsp + 0x14]
0x10ab20:    mov     byte ptr [rax + 4], cl
0x10ab23:    jmp     0x10a961
0x10ac58:    xor     r14d, r14d
0x10ac5b:    jmp     0x10ac6e
0x10a95b:    mov     edx, dword ptr [rip + 0x205a7]
0x10a961:    movzx   eax, byte ptr [rip + 0x209e8]
0x10a968:    shl     ebx, 3
0x10a96b:    mov     dword ptr [rip + 0x209d3], 1
0x10a975:    and     eax, 0xffffffffc7
0x10a978:    or      ebx, eax
0x10a97a:    mov     byte ptr [rip + 0x209d0], bl
0x10a980:    mov     rax, qword ptr [rip + 0x209b1]
0x10a987:    movzx   ebx, byte ptr [rax + rbp]
0x10a98b:    cmp     edx, 0x400
0x10a991:    je      0x10ab30
0x10ac44:    test    eax, eax
0x10ac46:    jns     0x10ac20
0x10ab46:    sub     edx, r14d
0x10ab49:    movsxd  rsi, r14d
0x10ab4c:    mov     edi, 1
0x10ab51:    add     rsi, r12
0x10ab54:    movsxd  rdx, edx
0x10ab57:    call    0x118da0
0x10a9b4:    mov     dword ptr [rsp + 4], 0xffffffff
0x10a9bc:    xor     r13d, r13d
0x10a9bf:    add     rbp, 1
0x10a9c3:    cmp     rbp, qword ptr [rsp + 8]
0x10a9c8:    je      0x10ab70
0x10a961:    movzx   eax, byte ptr [rip + 0x209e8]
0x10a968:    shl     ebx, 3
0x10a96b:    mov     dword ptr [rip + 0x209d3], 1
0x10a975:    and     eax, 0xffffffffc7
0x10a978:    or      ebx, eax
0x10a97a:    mov     byte ptr [rip + 0x209d0], bl
0x10a980:    mov     rax, qword ptr [rip + 0x209b1]
0x10a987:    movzx   ebx, byte ptr [rax + rbp]
0x10a98b:    cmp     edx, 0x400
0x10a991:    je      0x10ab30
0x10ac6e:    sub     edx, r14d
0x10ac71:    movsxd  rsi, r14d
0x10ac74:    mov     edi, 1
0x10ac79:    add     rsi, r12
0x10ac7c:    movsxd  rdx, edx
0x10ac7f:    call    0x118da0

```

```

0x10ac48:    mov     edx, 5
0x10ac4d:    mov     rax, r12
0x10ac50:    jmp     0x10aa97
0x10ac20:    mov     edx, dword ptr [rip + 0x202e2]
0x10ac26:    add     r15d, eax
0x10ac29:    cmp     edx, r15d
0x10ac2c:    jle     0x10ac48
0x10ab5c:    test    eax, eax
0x10ab5e:    jns     0x10ab38
0x10ac84:    test    eax, eax
0x10ac86:    jns     0x10ac60
0x10aa97:    mov     ecx, dword ptr [rsp + 0x10]
0x10aa9b:    mov     dword ptr [rip + 0x20467], edx
0x10aaa1:    mov     dword ptr [rax], ecx
0x10aaa3:    movzx   ecx, byte ptr [rsp + 0x14]
0x10aaa8:    mov     byte ptr [rax + 4], cl
0x10aaab:    jmp     0x10a911
0x10ab60:    mov     eax, 1
0x10ab65:    xor     edx, edx
0x10ab67:    jmp     0x10a99a
0x10ab38:    mov     edx, dword ptr [rip + 0x203ca]
0x10ab3e:    add     r14d, eax
0x10ab41:    cmp     edx, r14d
0x10ab44:    jle     0x10ab60
0x10ac88:    mov     edx, 5
0x10ac8d:    mov     rax, r12
0x10ac90:    jmp     0x10ab0f
0x10ac60:    mov     edx, dword ptr [rip + 0x202a2]
0x10ac66:    add     r14d, eax
0x10ac69:    cmp     edx, r14d
0x10ac6c:    jle     0x10ac88
0x10a99a:    movsxd  rdx, edx
0x10a99d:    add     r13d, 1
0x10a9a1:    cmp     dword ptr [rip + 0x1fff0], r13d
0x10a9a8:    mov     dword ptr [rip + 0x2055a], eax
0x10a9ae:    mov     byte ptr [r12 + rdx], bl
0x10a9b2:    jne     0x10a9bf
0x10ab0f:    mov     ecx, dword ptr [rsp + 0x10]
0x10ab13:    mov     dword ptr [rip + 0x203ef], edx
0x10ab19:    mov     dword ptr [rax], ecx
0x10ab1b:    movzx   ecx, byte ptr [rsp + 0x14]
0x10ab20:    mov     byte ptr [rax + 4], cl
0x10ab23:    jmp     0x10a961
0x10a8a2:    nop     word ptr [rax + rax]
0x10aa36:    nop     word ptr cs:[rax + rax]
0x10ab28:    nop     dword ptr [rax + rax]
0x10ab35:    nop     dword ptr [rax]
0x10ab6c:    nop     dword ptr [rax]

```

```

0x10ac14:    nop     dword ptr [rax]
0x10ac1d:    nop     dword ptr [rax]
0x10ac55:    nop     dword ptr [rax]
0x10ac5d:    nop     dword ptr [rax]
0x10ac95:    nop     dword ptr [rax]
0x10aca8:    nop     dword ptr [rax + rax]
0x10ace9:    nop     dword ptr [rax]
0x10acf0:    endbr64
0x10acf4:    push    r12
0x10acf6:    xor     eax, eax
0x10acf8:    mov     r12, rdi
0x10acfb:    push    rbx
0x10acfc:    mov     ebx, esi
0x10acfe:    sub     rsp, 8
0x10ad02:    call    0x10a820
0x10ad07:    mov     rdx, qword ptr [rip + 0x1fc1a]
0x10ad0e:    test    rdx, rdx
0x10ad11:    je      0x10ad17
0x10ad17:    movsxd  rdx, ebx
0x10ad1a:    mov     rsi, r12
0x10ad1d:    xor     edi, edi
0x10ad1f:    call    0x118d30
0x10ad13:    xor     eax, eax
0x10ad15:    call    rdx
0x10ad24:    mov     rdx, qword ptr [rip + 0x1fbf5]
0x10ad2b:    mov     r12d, eax
0x10ad2e:    test    rdx, rdx
0x10ad31:    je      0x10ad37
0x10ad37:    add     rsp, 8
0x10ad3b:    mov     eax, r12d
0x10ad3e:    pop     rbx
0x10ad3f:    pop     r12
0x10ad41:    ret
0x10ad33:    xor     eax, eax
0x10ad35:    call    rdx
0x10ad42:    nop     word ptr cs:[rax + rax]
0x10ad4d:    nop     dword ptr [rax]
0x10ad50:    push    r12
0x10ad52:    mov     edi, 1
0x10ad57:    push    rbp
0x10ad58:    lea     rbp, [rip + 0x1fda1]
0x10ad5f:    push    rbx
0x10ad60:    sub     rsp, 0x50
0x10ad64:    mov     rax, qword ptr fs:[0x28]
0x10ad6d:    mov     qword ptr [rsp + 0x48], rax
0x10ad72:    xor     eax, eax
0x10ad74:    mov     r12, rsp
0x10ad77:    mov     rsi, r12

```

```

0x10ad7a:    call    0x104d60
0x10ad7f:    mov     edx, 0xff
0x10ad84:    mov     esi, 1
0x10ad89:    mov     edi, 1
0x10ad8e:    mov     word ptr [rsp + 0x11], dx
0x10ad93:    mov     rdx, r12
0x10ad96:    call    0x104d70
0x10ad9b:    mov     edi, 1
0x10ada0:    mov     dword ptr [rip + 0x1ef7a], 0xffffffff
0x10adaa:    call    0x118b70
0x10adaf:    mov     dword ptr [rip + 0x20147], 0
0x10adb9:    nop     dword ptr [rax]
0x10adc0:    xor     eax, eax
0x10adc2:    call    0x10a820
0x10adc7:    mov     rdx, qword ptr [rip + 0x1fb5a]
0x10adce:    test    rdx, rdx
0x10add1:    je      0x10add7
0x10add7:    mov     edx, 0x400
0x10addc:    mov     rsi, rbp
0x10addf:    xor     edi, edi
0x10ade1:    call    0x118d30
0x10add3:    xor     eax, eax
0x10add5:    call    rdx
0x10ade6:    mov     rdx, qword ptr [rip + 0x1fb33]
0x10aded:    mov     ebx, eax
0x10adef:    test    rdx, rdx
0x10adf2:    je      0x10adf8
0x10adf8:    mov     dword ptr [rip + 0x20106], ebx
0x10adfe:    test    ebx, ebx
0x10ae00:    js      0x10adc0
0x10adf4:    xor     eax, eax
0x10adf6:    call    rdx
0x10adc0:    xor     eax, eax
0x10adc2:    call    0x10a820
0x10ae02:    xor     edi, edi
0x10ae04:    call    0x118b70
0x10ae09:    mov     rsi, r12
0x10ae0c:    mov     edi, 1
0x10ae11:    call    0x104d60
0x10ae16:    mov     eax, 7
0x10ae1b:    mov     rdx, r12
0x10ae1e:    mov     esi, 1
0x10ae23:    mov     edi, 1
0x10ae28:    mov     word ptr [rsp + 0x11], ax
0x10ae2d:    call    0x104d70
0x10ae32:    mov     eax, dword ptr [rip + 0x200cc]
0x10ae38:    mov     dword ptr [rip + 0x1eee2], 7
0x10ae42:    test    eax, eax

```



```

0x10ae44:    je      0x10ae9f
0x10ae9f:    mov     r8d, 0xffffffff
0x10aea5:    jmp     0x10ae70
0x10ae46:    movsxd  rsi, dword ptr [rip + 0x200b3]
0x10ae4d:    movsx   r8d, byte ptr [rbp + rsi]
0x10ae53:    mov     rdx, rsi
0x10ae56:    mov     ecx, r8d
0x10ae59:    test    r8b, r8b
0x10ae5c:    js      0x10ae90
0x10ae70:    mov     rax, qword ptr [rsp + 0x48]
0x10ae75:    sub     rax, qword ptr fs:[0x28]
0x10ae7e:    jne     0x10aea7
0x10ae90:    and     ecx, 0x7f
0x10ae93:    mov     r8d, 0x1b
0x10ae99:    mov     byte ptr [rbp + rsi], cl
0x10ae9d:    jmp     0x10ae70
0x10ae5e:    sub     eax, 1
0x10ae61:    add     edx, 1
0x10ae64:    mov     dword ptr [rip + 0x2009a], eax
0x10ae6a:    mov     dword ptr [rip + 0x20090], edx
0x10ae70:    mov     rax, qword ptr [rsp + 0x48]
0x10ae75:    sub     rax, qword ptr fs:[0x28]
0x10ae7e:    jne     0x10aea7
0x10ae80:    add     rsp, 0x50
0x10ae84:    mov     eax, r8d
0x10ae87:    pop     rbx
0x10ae88:    pop     rbp
0x10ae89:    pop     r12
0x10ae8b:    ret
0x10aea7:    call    0x1049d0
0x10ae8c:    nop     dword ptr [rax]
0x10aeac:    nop     dword ptr [rax]
0x10aeb0:    endbr64
0x10aeb4:    xor     eax, eax
0x10aeb6:    sub     rsp, 8
0x10aeba:    call    0x118ab0
0x10aebf:    mov     eax, dword ptr [rip + 0x2003f]
0x10aec5:    test    eax, eax
0x10aec7:    jne     0x10aed8
0x10aec9:    xor     eax, eax
0x10aecb:    add     rsp, 8
0x10aecf:    jmp     0x10ad50
0x10aed8:    movsxd  rsi, dword ptr [rip + 0x20021]
0x10aedf:    lea     rdi, [rip + 0x1fc1a]
0x10aee6:    movsx   r8d, byte ptr [rdi + rsi]
0x10aeeb:    mov     rdx, rsi
0x10aeee:    mov     ecx, r8d
0x10aef1:    test    r8b, r8b

```

```

0x10aef4:      js      0x10af10
0x10af10:      and     ecx, 0x7f
0x10af13:      mov     r8d, 0x1b
0x10af19:      mov     byte ptr [rdi + rsi], cl
0x10af1c:      jmp     0x10af08
0x10aef6:      sub     eax, 1
0x10aef9:      add     edx, 1
0x10aefc:      mov     dword ptr [rip + 0x20002], eax
0x10af02:      mov     dword ptr [rip + 0x1fff8], edx
0x10af08:      mov     eax, r8d
0x10af0b:      add     rsp, 8
0x10af0f:      ret
0x10af08:      mov     eax, r8d
0x10af0b:      add     rsp, 8
0x10af0f:      ret
0x10aed4:      nop     dword ptr [rax]
0x10af1e:      nop
0x10af20:      endbr64
0x10af24:      push    r13
0x10af26:      mov     r13, rsi
0x10af29:      push    r12
0x10af2b:      mov     r12, rdx
0x10af2e:      push    rbp
0x10af2f:      mov     rbp, rdi
0x10af32:      mov     edi, 0x18
0x10af37:      push    rbx
0x10af38:      sub     rsp, 8
0x10af3c:      call    0x11d670
0x10af41:      mov     rdi, r13
0x10af44:      mov     rbx, rax
0x10af47:      call    0x11d7a0
0x10af4c:      mov     qword ptr [rbx + 0x10], r12
0x10af50:      mov     qword ptr [rbx], rax
0x10af53:      mov     rax, qword ptr [rbp]
0x10af57:      mov     qword ptr [rbx + 8], rax
0x10af5b:      mov     qword ptr [rbp], rbx
0x10af5f:      add     rsp, 8
0x10af63:      pop     rbx
0x10af64:      pop     rbp
0x10af65:      pop     r12
0x10af67:      pop     r13
0x10af69:      ret
0x10af6a:      nop     word ptr [rax + rax]
0x10af70:      endbr64
0x10af74:      cmp     byte ptr [rdi], 0
0x10af77:      je      0x10b030
0x10b030:      ret
0x10af7d:      push    r13

```

```

0x10af7f:    push    r12
0x10af81:    mov     r12, rsi
0x10af84:    push    rbp
0x10af85:    mov     rbp, rdi
0x10af88:    push    rbx
0x10af89:    sub     rsp, 8
0x10af8d:    mov     r13, qword ptr [rip + 0x1fa0c]
0x10af94:    test    r13, r13
0x10af97:    je      0x10b010
0x10b010:    mov     rbx, qword ptr [rip + 0x1f969]
0x10b017:    lea     r13, [rip + 0x1f962]
0x10b01e:    mov     qword ptr [rip + 0x1f97b], r13
0x10b025:    test    rbx, rbx
0x10b028:    jne     0x10afc4
0x10af99:    mov     rbx, qword ptr [r13]
0x10af9d:    mov     rsi, qword ptr [rbx]
0x10afa0:    call    0x104b30
0x10b02a:    jmp     0x10afd3
0x10afc4:    mov     rsi, qword ptr [rbx]
0x10afc7:    mov     rdi, rbp
0x10afca:    call    0x104b30
0x10afa5:    test    eax, eax
0x10afa7:    jle     0x10b010
0x10afd3:    mov     edi, 0x18
0x10afd8:    call    0x11d670
0x10afcfc:    test    eax, eax
0x10afdc:    jg      0x10afb0
0x10afa9:    nop     dword ptr [rax]
0x10afdd:    mov     rdi, rbp
0x10afe0:    mov     rbx, rax
0x10afe3:    call    0x11d7a0
0x10afb0:    lea     r13, [rbx + 8]
0x10afb4:    mov     rbx, qword ptr [rbx + 8]
0x10afb8:    mov     qword ptr [rip + 0x1f9e1], r13
0x10afbf:    test    rbx, rbx
0x10afc2:    je      0x10afd3
0x10afe8:    mov     qword ptr [rbx + 0x10], r12
0x10afec:    mov     qword ptr [rbx], rax
0x10afef:    mov     rax, qword ptr [r13]
0x10aff3:    mov     qword ptr [rbx + 8], rax
0x10aff7:    mov     qword ptr [r13], rbx
0x10affb:    add     rsp, 8
0x10afff:    pop     rbx
0x10b000:    pop     rbp
0x10b001:    pop     r12
0x10b003:    pop     r13
0x10b005:    ret
0x10b006:    nop     word ptr cs:[rax + rax]

```

```

0x10b02c:    nop     dword ptr [rax]
0x10b031:    nop     word ptr cs:[rax + rax]
0x10b03c:    nop     dword ptr [rax]
0x10b040:    endbr64
0x10b044:    mov     rax, qword ptr [rip + 0x1f935]
0x10b04b:    mov     qword ptr [rip + 0x1f926], rax
0x10b052:    ret
0x10b053:    nop     word ptr cs:[rax + rax]
0x10b05e:    nop
0x10b060:    endbr64
0x10b064:    cmp     qword ptr [rip + 0x1f90c], 0
0x10b06c:    je      0x10b073
0x10b073:    xor     eax, eax
0x10b075:    ret
0x10b06e:    jmp     0x1099e0
0x10b076:    nop     word ptr cs:[rax + rax]
0x10b080:    endbr64
0x10b084:    mov     eax, dword ptr [rip + 0x1f912]
0x10b08a:    push    r12
0x10b08c:    mov     r12, rdi
0x10b08f:    push    rbp
0x10b090:    push    rbx
0x10b091:    sub     eax, 1
0x10b094:    movzx   ebx, byte ptr [rip + 0x202b6]
0x10b09b:    mov     dword ptr [rip + 0x202b3], eax
0x10b0a1:    mov     dword ptr [rip + 0x202ad], 0
0x10b0ab:    mov     eax, ebx
0x10b0ad:    and     eax, 0xffffffffc0
0x10b0b0:    or      eax, 0x38
0x10b0b3:    mov     byte ptr [rip + 0x20298], al
0x10b0b9:    mov     eax, dword ptr [rip + 0x1f8d9]
0x10b0bf:    lea     edi, [rax + 1]
0x10b0c2:    movsxd  rdi, edi
0x10b0c5:    call    0x11d670
0x10b0ca:    movsxd  rdx, dword ptr [rip + 0x1f8c7]
0x10b0d1:    mov     esi, 0x20
0x10b0d6:    mov     rdi, rax
0x10b0d9:    mov     rbp, rax
0x10b0dc:    call    0x104a50
0x10b0e1:    movsxd  rax, dword ptr [rip + 0x1f8b0]
0x10b0e8:    mov     rdi, rbp
0x10b0eb:    mov     byte ptr [rbp + rax], 0
0x10b0f0:    mov     esi, dword ptr [rip + 0x1f8a2]
0x10b0f6:    call    0x10a1a0
0x10b0fb:    mov     rdi, rbp
0x10b0fe:    lea     rbp, [rip + 0x1f8bb]
0x10b105:    call    0x118d10
0x10b10a:    mov     eax, dword ptr [rip + 0x1f88c]

```

```

0x10b110:    mov     rdi, r12
0x10b113:    mov     dword ptr [rip + 0x2023b], 0
0x10b11d:    sub     eax, 1
0x10b120:    mov     dword ptr [rip + 0x2022e], eax
0x10b126:    call    0x109fe0
0x10b12b:    mov     esi, 0x13
0x10b130:    lea     rdi, [rip + 0x15ad5]
0x10b137:    call    0x10a1a0
0x10b13c:    mov     rdi, rbp
0x10b13f:    call    0x1049b0
0x10b144:    mov     rdi, rbp
0x10b147:    mov     esi, eax
0x10b149:    call    0x10a1a0
0x10b14e:    mov     esi, 1
0x10b153:    lea     rdi, [rip + 0x155eb]
0x10b15a:    call    0x10a1a0
0x10b15f:    xor     eax, eax
0x10b161:    call    0x10a820
0x10b166:    mov     byte ptr [rip + 0x201e5], bl
0x10b16c:    pop     rbx
0x10b16d:    pop     rbp
0x10b16e:    pop     r12
0x10b170:    ret
0x10b171:    nop     word ptr cs:[rax + rax]
0x10b17c:    nop     dword ptr [rax]
0x10b180:    endbr64
0x10b184:    push    r15
0x10b186:    push    r14
0x10b188:    mov     r14, rdi
0x10b18b:    push    r13
0x10b18d:    push    r12
0x10b18f:    push    rbp
0x10b190:    lea     rbp, [rip + 0x1f969]
0x10b197:    push    rbx
0x10b198:    sub     rsp, 0x28
0x10b19c:    jmp     0x10b1ae
0x10b1ae:    xor     eax, eax
0x10b1b0:    call    0x118ab0
0x10b1b5:    mov     eax, dword ptr [rip + 0x1fd49]
0x10b1bb:    test    eax, eax
0x10b1bd:    je      0x10b1a0
0x10b1a0:    xor     eax, eax
0x10b1a2:    call    0x10ad50
0x10b1bf:    movsxd  rsi, dword ptr [rip + 0x1fd3a]
0x10b1c6:    movsx   ebx, byte ptr [rbp + rsi]
0x10b1cb:    mov     rdx, rsi
0x10b1ce:    mov     ecx, ebx
0x10b1d0:    test    bl, bl

```

```

0x10b1d2:    js      0x10b4a0
0x10b1a7:    mov     ebx, eax
0x10b1a9:    cmp     ebx, -1
0x10b1ac:    jne     0x10b1ef
0x10b4a0:    and     ecx, 0x7f
0x10b4a3:    mov     byte ptr [rbp + rsi], cl
0x10b4a7:    test    r14, r14
0x10b4aa:    je      0x10b4b3
0x10b1d8:    sub     eax, 1
0x10b1db:    add     edx, 1
0x10b1de:    mov     dword ptr [rip + 0x1fd20], eax
0x10b1e4:    mov     dword ptr [rip + 0x1fd16], edx
0x10b1ea:    cmp     ebx, -1
0x10b1ed:    je      0x10b1ae
0x10b1ef:    test    r14, r14
0x10b1f2:    je      0x10b1fb
0x10b4b3:    mov     eax, dword ptr [rip + 0x1ff43]
0x10b4b9:    mov     ebx, 0x1b
0x10b4be:    test    eax, eax
0x10b4c0:    jne     0x10b240
0x10b4ac:    mov     dword ptr [r14], 1
0x10b4b3:    mov     eax, dword ptr [rip + 0x1ff43]
0x10b4b9:    mov     ebx, 0x1b
0x10b4be:    test    eax, eax
0x10b4c0:    jne     0x10b240
0x10b1fb:    mov     eax, dword ptr [rip + 0x201fb]
0x10b201:    test    ebx, ebx
0x10b203:    jne     0x10b445
0x10b1f4:    mov     dword ptr [r14], 1
0x10b1fb:    mov     eax, dword ptr [rip + 0x201fb]
0x10b201:    test    ebx, ebx
0x10b203:    jne     0x10b445
0x10b4c6:    movsxd  rbp, ebx
0x10b4c9:    add     rbp, rbp
0x10b4cc:    cmp     ebx, 7
0x10b4cf:    sete    r12b
0x10b4d3:    jmp     0x10b21f
0x10b240:    mov     rax, qword ptr [rip + 0x1f739]
0x10b247:    xor     r12d, r12d
0x10b24a:    lea     r13, [rip + 0x1f8af]
0x10b251:    mov     dword ptr [rip + 0x1f751], 0
0x10b25b:    mov     dword ptr [rip + 0x1f743], 0
0x10b265:    mov     qword ptr [rip + 0x1f70c], rax
0x10b26c:    test    ebx, ebx
0x10b26e:    mov     eax, 0xffffffff
0x10b273:    movsxd  rbp, r12d
0x10b276:    cmovne  eax, ebx
0x10b279:    mov     rbx, qword ptr [rip + 0x200e0]

```

```

0x10b280:    mov     byte ptr [rbx + rbp], al
0x10b283:    add     rbp, 1
0x10b287:    mov     byte ptr [rbx + rbp], 0
0x10b28b:    cmp     qword ptr [rip + 0x1f6e5], 0
0x10b293:    je      0x10b2b3
0x10b209:    mov     ebx, 0xff
0x10b20e:    test    eax, eax
0x10b210:    jne     0x10b240
0x10b445:    test    eax, eax
0x10b447:    jne     0x10b240
0x10b21f:    call    0x104f00
0x10b2b3:    mov     ecx, dword ptr [rip + 0x1fc4b]
0x10b2b9:    test    ecx, ecx
0x10b2bb:    jne     0x10b2ee
0x10b295:    mov     rdi, rbx
0x10b298:    call    0x1099e0
0x10b212:    xor     r12d, r12d
0x10b215:    mov     ebp, 0xfe
0x10b21a:    mov     ebx, 0xff
0x10b21f:    call    0x104f00
0x10b44d:    cmp     ebx, 0xa
0x10b450:    sete    bpl
0x10b454:    cmp     ebx, 0xd
0x10b457:    sete    al
0x10b45a:    or      ebp, eax
0x10b45c:    call    0x104f00
0x10b224:    mov     rax, qword ptr [rax]
0x10b227:    movzx   eax, word ptr [rax + rbp]
0x10b22b:    test    ah, 0x40
0x10b22e:    jne     0x10b47d
0x10b2bd:    mov     edx, dword ptr [rip + 0x1f6e5]
0x10b2c3:    test    edx, edx
0x10b2c5:    je      0x10b380
0x10b2ee:    xor     eax, eax
0x10b2f0:    call    0x118ab0
0x10b29d:    mov     r15, rax
0x10b2a0:    cmp     rax, -1
0x10b2a4:    je      0x10b33a
0x10b461:    mov     rax, qword ptr [rax]
0x10b464:    test    bpl, bpl
0x10b467:    je      0x10b4c6
0x10b234:    test    r12b, r12b
0x10b237:    jne     0x10b47d
0x10b47d:    mov     rax, qword ptr [rip + 0x1f4dc]
0x10b484:    lea     r15, [rip + 0x1f4d5]
0x10b48b:    mov     byte ptr [rax], bl
0x10b48d:    mov     rax, qword ptr [rip + 0x1f4cc]
0x10b494:    mov     byte ptr [rax + 1], 0

```

```

0x10b498:      jmp     0x10b433
0x10b380:      mov     edi, 1
0x10b385:      call    0x104aa0
0x10b2cb:      mov     rdi, rbx
0x10b2ce:      call    0x10b080
0x10b2f5:      mov     eax, dword ptr [rip + 0x1fc09]
0x10b2fb:      test    eax, eax
0x10b2fd:      je       0x10b2e0
0x10b33a:      mov     edi, 1
0x10b33f:      xor     r15d, r15d
0x10b342:      call    0x104aa0
0x10b2aa:      test    rax, rax
0x10b2ad:      jne     0x10b38f
0x10b469:      mov     ebx, 0xa
0x10b46e:      test    byte ptr [rax + 0x15], 0x40
0x10b472:      je       0x10b240
0x10b23d:      nop     dword ptr [rax]
0x10b433:      add     rsp, 0x28
0x10b437:      mov     rax, r15
0x10b43a:      pop     rbx
0x10b43b:      pop     rbp
0x10b43c:      pop     r12
0x10b43e:      pop     r13
0x10b440:      pop     r14
0x10b442:      pop     r15
0x10b444:      ret
0x10b38a:      jmp     0x10b2d3
0x10b2d3:      mov     dword ptr [rip + 0x1f6cf], 1
0x10b2dd:      jmp     0x10b2ee
0x10b2e0:      xor     eax, eax
0x10b2e2:      call    0x10ad50
0x10b2ff:      movsxd  rsi, dword ptr [rip + 0x1fbfa]
0x10b306:      movsx   ebx, byte ptr [r13 + rsi]
0x10b30c:      mov     rdx, rsi
0x10b30f:      mov     ecx, ebx
0x10b311:      test    bl, bl
0x10b313:      js       0x10b360
0x10b347:      mov     dword ptr [rip + 0x1f65b], 0
0x10b351:      jmp     0x10b433
0x10b38f:      test    r14, r14
0x10b392:      je       0x10b41f
0x10b478:      mov     ebx, 0xa
0x10b47d:      mov     rax, qword ptr [rip + 0x1f4dc]
0x10b484:      lea     r15, [rip + 0x1f4d5]
0x10b48b:      mov     byte ptr [rax], bl
0x10b48d:      mov     rax, qword ptr [rip + 0x1f4cc]
0x10b494:      mov     byte ptr [rax + 1], 0
0x10b498:      jmp     0x10b433

```



```

0x10b2e7:    mov     ebx, eax
0x10b2e9:    cmp     ebx, -1
0x10b2ec:    jne     0x10b32c
0x10b360:    and     ecx, 0x7f
0x10b363:    add     r12d, 1
0x10b367:    mov     byte ptr [r13 + rsi], cl
0x10b36c:    cmp     r12d, 0xf
0x10b370:    je      0x10b33a
0x10b315:    sub     eax, 1
0x10b318:    add     edx, 1
0x10b31b:    mov     dword ptr [rip + 0x1fbe3], eax
0x10b321:    mov     dword ptr [rip + 0x1fbd9], edx
0x10b327:    cmp     ebx, -1
0x10b32a:    je      0x10b2ee
0x10b41f:    mov     edi, 1
0x10b424:    call    0x104aa0
0x10b398:    mov     ecx, dword ptr [rip + 0x1fb66]
0x10b39e:    cmp     r12d, ecx
0x10b3a1:    jge     0x10b41f
0x10b32c:    add     r12d, 1
0x10b330:    cmp     r12d, 0xf
0x10b334:    jne     0x10b26c
0x10b372:    mov     eax, 0x1b
0x10b377:    jmp     0x10b279
0x10b429:    mov     dword ptr [rip + 0x1f579], 0
0x10b433:    add     rsp, 0x28
0x10b437:    mov     rax, r15
0x10b43a:    pop     rbx
0x10b43b:    pop     rbp
0x10b43c:    pop     r12
0x10b43e:    pop     r13
0x10b440:    pop     r14
0x10b442:    pop     r15
0x10b444:    ret
0x10b3a3:    movsxd  rax, dword ptr [rip + 0x1fb56]
0x10b3aa:    lea     r9d, [r12 + 1]
0x10b3af:    lea     rdx, [rip + 0x1f74a]
0x10b3b6:    movsxd  rdi, r9d
0x10b3b9:    mov     qword ptr [rsp + 0x18], rdi
0x10b3be:    lea     r8d, [r9 + rax]
0x10b3c2:    lea     r13, [rax + rdx]
0x10b3c6:    jmp     0x10b3f1
0x10b26c:    test    ebx, ebx
0x10b26e:    mov     eax, 0xffffffff
0x10b273:    movsxd  rbp, r12d
0x10b276:    cmovne  eax, ebx
0x10b279:    mov     rbx, qword ptr [rip + 0x200e0]
0x10b280:    mov     byte ptr [rbx + rbp], al

```

```

0x10b283:    add     rbp, 1
0x10b287:    mov     byte ptr [rbx + rbp], 0
0x10b28b:    cmp     qword ptr [rip + 0x1f6e5], 0
0x10b293:    je      0x10b2b3
0x10b279:    mov     rbx, qword ptr [rip + 0x200e0]
0x10b280:    mov     byte ptr [rbx + rbp], al
0x10b283:    add     rbp, 1
0x10b287:    mov     byte ptr [rbx + rbp], 0
0x10b28b:    cmp     qword ptr [rip + 0x1f6e5], 0
0x10b293:    je      0x10b2b3
0x10b3f1:    mov     rdx, rbp
0x10b3f4:    mov     rsi, r13
0x10b3f7:    mov     rdi, rbx
0x10b3fa:    mov     dword ptr [rsp + 0x14], r8d
0x10b3ff:    mov     dword ptr [rsp + 0x10], r9d
0x10b404:    mov     dword ptr [rsp + 0xc], ecx
0x10b408:    call    0x104af0
0x10b40d:    mov     ecx, dword ptr [rsp + 0xc]
0x10b411:    mov     r9d, dword ptr [rsp + 0x10]
0x10b416:    test    eax, eax
0x10b418:    mov     r8d, dword ptr [rsp + 0x14]
0x10b41d:    je      0x10b3d0
0x10b3d0:    sub     ecx, r9d
0x10b3d3:    add     dword ptr [r14], 1
0x10b3d7:    add     r13, qword ptr [rsp + 0x18]
0x10b3dc:    mov     dword ptr [rip + 0x1fb1d], r8d
0x10b3e3:    add     r8d, r9d
0x10b3e6:    mov     dword ptr [rip + 0x1fb18], ecx
0x10b3ec:    cmp     r12d, ecx
0x10b3ef:    jge     0x10b41f
0x10b19e:    nop
0x10b2df:    nop
0x10b356:    nop     word ptr cs:[rax + rax]
0x10b37c:    nop     dword ptr [rax]
0x10b3c8:    nop     dword ptr [rax + rax]
0x10b49a:    nop     word ptr [rax + rax]
0x10b4d8:    nop     dword ptr [rax + rax]
0x10b4e0:    endbr64
0x10b4e4:    mov     eax, dword ptr [rip + 0x1f4c2]
0x10b4ea:    test    eax, eax
0x10b4ec:    jne     0x10b4f0
0x10b4ee:    ret
0x10b4f0:    sub     rsp, 8
0x10b4f4:    mov     rdi, qword ptr [rip + 0x1fe65]
0x10b4fb:    call    0x10b080
0x10b500:    mov     dword ptr [rip + 0x1f49e], 1
0x10b50a:    add     rsp, 8
0x10b50e:    ret

```

```

0x10b4ef:      nop
0x10b50f:      nop
0x10b510:      endbr64
0x10b514:      mov     rax, qword ptr [rip + 0x1fe45]
0x10b51b:      ret
0x10b51c:      nop     dword ptr [rax]
0x10b520:      endbr64
0x10b524:      push    r15
0x10b526:      mov     esi, 0x5413
0x10b52b:      mov     edi, 1
0x10b530:      push    r14
0x10b532:      push    r13
0x10b534:      push    r12
0x10b536:      push    rbp
0x10b537:      push    rbx
0x10b538:      sub     rsp, 0x58
0x10b53c:      mov     rax, qword ptr fs:[0x28]
0x10b545:      mov     qword ptr [rsp + 0x48], rax
0x10b54a:      xor     eax, eax
0x10b54c:      lea     rdx, [rsp + 0x18]
0x10b551:      mov     dword ptr [rsp + 0x10], 0
0x10b559:      mov     dword ptr [rsp + 0x14], 0
0x10b561:      call    0x104a70
0x10b566:      test    eax, eax
0x10b568:      jne     0x10b5a0
0x10b56a:      mov     ebx, eax
0x10b56c:      movzx   eax, word ptr [rsp + 0x1a]
0x10b571:      test    ax, ax
0x10b574:      je      0x10b790
0x10b5a0:      mov     dword ptr [rsp + 0xc], 0xffffffff
0x10b5a8:      mov     r12d, 0xffffffff
0x10b5ae:      xor     ebx, ebx
0x10b5b0:      xor     ebp, ebp
0x10b5b2:      lea     r13, [rip + 0x15667]
0x10b5b9:      lea     r15, [rip + 0x15a13]
0x10b5c0:      mov     rdi, r13
0x10b5c3:      call    0x1047e0
0x10b790:      mov     dword ptr [rsp + 0xc], 0xffffffff
0x10b798:      xor     ebp, ebp
0x10b79a:      mov     r12d, 0xffffffff
0x10b7a0:      jmp     0x10b5b2
0x10b57a:      movzx   ebp, word ptr [rsp + 0x18]
0x10b57f:      test    bp, bp
0x10b582:      je      0x10b790
0x10b5c8:      mov     rdi, rax
0x10b5cb:      test    rax, rax
0x10b5ce:      je      0x10b5df
0x10b5b2:      lea     r13, [rip + 0x15667]

```

```

0x10b5b9:    lea     r15, [rip + 0x15a13]
0x10b5c0:    mov     rdi, r13
0x10b5c3:    call    0x1047e0
0x10b588:    movzx   ebx, ax
0x10b58b:    lea     r12d, [rbp - 1]
0x10b58f:    lea     eax, [rbx - 1]
0x10b592:    mov     dword ptr [rsp + 0xc], eax
0x10b596:    jmp     0x10b5b2
0x10b5df:    lea     r14, [rip + 0x15642]
0x10b5e6:    mov     rdi, r14
0x10b5e9:    call    0x1047e0
0x10b5d0:    lea     rdx, [rsp + 0x14]
0x10b5d5:    mov     rsi, r15
0x10b5d8:    xor     eax, eax
0x10b5da:    call    0x104c60
0x10b5ee:    mov     rdi, rax
0x10b5f1:    test    rax, rax
0x10b5f4:    je      0x10b605
0x10b605:    cmp     dword ptr [rsp + 0xc], 0x3ff
0x10b60d:    mov     eax, dword ptr [rip + 0x1e8fd]
0x10b613:    jbe     0x10b628
0x10b5f6:    lea     rdx, [rsp + 0x10]
0x10b5fb:    mov     rsi, r15
0x10b5fe:    xor     eax, eax
0x10b600:    call    0x104c60
0x10b628:    mov     dword ptr [rip + 0x1f36a], ebx
0x10b62e:    cmp     r12d, 0x3ff
0x10b635:    jbe     0x10b64a
0x10b615:    mov     ebx, dword ptr [rsp + 0x14]
0x10b619:    lea     edx, [rbx - 1]
0x10b61c:    cmp     edx, 0x3ff
0x10b622:    ja      0x10b7c0
0x10b64a:    lea     r12, [rsp + 0x20]
0x10b64f:    mov     r8d, ebp
0x10b652:    mov     rcx, r15
0x10b655:    mov     edx, 0x20
0x10b65a:    mov     esi, 1
0x10b65f:    mov     rdi, r12
0x10b662:    xor     eax, eax
0x10b664:    mov     dword ptr [rip + 0x1f332], ebp
0x10b66a:    call    0x104f10
0x10b637:    mov     ebp, dword ptr [rsp + 0x10]
0x10b63b:    lea     edx, [rbp - 1]
0x10b63e:    cmp     edx, 0x3ff
0x10b644:    ja      0x10b7a8
0x10b7c0:    mov     ebx, dword ptr [rip + 0x1e72a]
0x10b7c6:    lea     edx, [rbx - 1]
0x10b7c9:    cmp     edx, 0x3ff

```

```

0x10b7cf:      mov     edx, 0x50
0x10b7d4:      cmova   ebx, edx
0x10b7d7:      jmp     0x10b628
0x10b66f:      mov     rsi, r12
0x10b672:      mov     rdi, r14
0x10b675:      call    0x118380
0x10b7a8:      lea     edx, [rax - 1]
0x10b7ab:      mov     ebp, 0x18
0x10b7b0:      cmp     edx, 0x400
0x10b7b6:      cmovb    ebp, eax
0x10b7b9:      jmp     0x10b64a
0x10b67a:      mov     rcx, r15
0x10b67d:      mov     edx, 0x20
0x10b682:      mov     rdi, r12
0x10b685:      mov     r8d, dword ptr [rip + 0x1f30c]
0x10b68c:      mov     esi, 1
0x10b691:      xor     eax, eax
0x10b693:      call    0x104f10
0x10b698:      mov     rdi, r13
0x10b69b:      mov     rsi, r12
0x10b69e:      call    0x118380
0x10b6a3:      mov     rdi, qword ptr [rip + 0x1fc8e]
0x10b6aa:      test    rdi, rdi
0x10b6ad:      je      0x10b6b4
0x10b6b4:      mov     rdi, qword ptr [rip + 0x1fc75]
0x10b6bb:      test    rdi, rdi
0x10b6be:      je      0x10b6c5
0x10b6af:      call    0x118d10
0x10b6c5:      mov     rdi, qword ptr [rip + 0x1fc5c]
0x10b6cc:      test    rdi, rdi
0x10b6cf:      je      0x10b6d6
0x10b6c0:      call    0x118d10
0x10b6d6:      mov     rdi, qword ptr [rip + 0x1fc43]
0x10b6dd:      test    rdi, rdi
0x10b6e0:      je      0x10b6e7
0x10b6d1:      call    0x118d10
0x10b6e7:      mov     edi, dword ptr [rip + 0x1f2ab]
0x10b6ed:      imul    edi, dword ptr [rip + 0x1f2a8]
0x10b6f4:      mov     esi, 1
0x10b6f9:      movsxd  rdi, edi
0x10b6fc:      call    0x11d740
0x10b6e2:      call    0x118d10
0x10b701:      mov     edi, dword ptr [rip + 0x1f291]
0x10b707:      mov     esi, 1
0x10b70c:      imul    edi, dword ptr [rip + 0x1f289]
0x10b713:      mov     qword ptr [rip + 0x1fc1e], rax
0x10b71a:      movsxd  rdi, edi
0x10b71d:      call    0x11d740

```

```

0x10b722:    mov     edi, dword ptr [rip + 0x1f270]
0x10b728:    mov     esi, 1
0x10b72d:    imul    edi, dword ptr [rip + 0x1f268]
0x10b734:    mov     qword ptr [rip + 0x1fbf5], rax
0x10b73b:    movsxd  rdi, edi
0x10b73e:    call    0x11d740
0x10b743:    mov     edi, dword ptr [rip + 0x1f24f]
0x10b749:    mov     esi, 1
0x10b74e:    imul    edi, dword ptr [rip + 0x1f247]
0x10b755:    mov     qword ptr [rip + 0x1fbcc], rax
0x10b75c:    movsxd  rdi, edi
0x10b75f:    call    0x11d740
0x10b764:    mov     qword ptr [rip + 0x1fbb5], rax
0x10b76b:    mov     rax, qword ptr [rsp + 0x48]
0x10b770:    sub     rax, qword ptr fs:[0x28]
0x10b779:    jne     0x10b7dc
0x10b77b:    add     rsp, 0x58
0x10b77f:    pop     rbx
0x10b780:    pop     rbp
0x10b781:    pop     r12
0x10b783:    pop     r13
0x10b785:    pop     r14
0x10b787:    pop     r15
0x10b789:    ret
0x10b7dc:    call    0x1049d0
0x10b598:    nop     dword ptr [rax + rax]
0x10b78a:    nop     word ptr [rax + rax]
0x10b7a5:    nop     dword ptr [rax]
0x10b7be:    nop
0x10b7e1:    nop     word ptr cs:[rax + rax]
0x10b7ec:    nop     dword ptr [rax]
0x10b7f0:    endbr64
0x10b7f4:    push    r15
0x10b7f6:    push    r14
0x10b7f8:    push    r13
0x10b7fa:    push    r12
0x10b7fc:    mov     r12, rdi
0x10b7ff:    mov     edi, 0x5858
0x10b804:    push    rbp
0x10b805:    push    rbx
0x10b806:    mov     ebx, esi
0x10b808:    sub     rsp, 0x38
0x10b80c:    mov     rax, qword ptr fs:[0x28]
0x10b815:    mov     qword ptr [rsp + 0x28], rax
0x10b81a:    movabs  rax, 0x7363762f7665642f
0x10b824:    mov     word ptr [rsp + 8], di
0x10b829:    mov     byte ptr [rsp + 0xa], 0
0x10b82e:    mov     dword ptr [rsp + 0x18], 0x585861

```

```

0x10b836:    mov     qword ptr [rsp], rax
0x10b83a:    mov     qword ptr [rsp + 0x10], rax
0x10b83f:    cmp     esi, 1
0x10b842:    je      0x10b978
0x10b978:    mov     r13, qword ptr [rip + 0xd661]
0x10b97f:    xor     r14d, r14d
0x10b982:    jmp     0x10b863
0x10b848:    mov     esi, dword ptr [rip + 0x1fbb2]
0x10b84e:    mov     r13, qword ptr [rip + 0xd783]
0x10b855:    mov     r14d, 1
0x10b85b:    test    esi, esi
0x10b85d:    jne     0x10b90c
0x10b863:    mov     rax, qword ptr [rip + 0x1f126]
0x10b86a:    lea     r15, [rsp + 0x10]
0x10b86f:    mov     esi, r14d
0x10b872:    mov     dword ptr [rip + 0x1fb84], 0
0x10b87c:    mov     rdi, r15
0x10b87f:    movzx   edx, byte ptr [rax + 8]
0x10b883:    mov     byte ptr [rsp + 0x19], dl
0x10b887:    movzx   eax, byte ptr [rax + 9]
0x10b88b:    mov     byte ptr [rsp + 0x1a], al
0x10b88f:    call    0x104930
0x10b90c:    mov     dword ptr [rip + 0x1faea], 1
0x10b916:    cmp     ebx, 2
0x10b919:    je      0x10ba30
0x10b894:    mov     ebp, eax
0x10b896:    cmp     eax, -1
0x10b899:    je      0x10b8f0
0x10ba30:    xor     eax, eax
0x10ba32:    mov     r15, rsp
0x10ba35:    call    0x10a300
0x10b91f:    mov     rax, qword ptr [rip + 0x1f06a]
0x10b926:    mov     r15, rsp
0x10b929:    mov     esi, r14d
0x10b92c:    mov     rdi, r15
0x10b92f:    movzx   edx, byte ptr [rax + 8]
0x10b933:    movzx   eax, byte ptr [rax + 9]
0x10b937:    mov     byte ptr [rsp + 8], dl
0x10b93b:    mov     byte ptr [rsp + 9], al
0x10b93f:    call    0x104930
0x10b8f0:    mov     ecx, 0x30
0x10b8f5:    mov     esi, r14d
0x10b8f8:    mov     rdi, r15
0x10b8fb:    mov     word ptr [rsp + 0x19], cx
0x10b900:    call    0x104930
0x10b89b:    mov     eax, dword ptr [rip + 0x1f0fb]
0x10b8a1:    imul    eax, dword ptr [rip + 0x1f0f0]
0x10b8a8:    mov     edi, ebp

```

```

0x10b8aa:    mov     rsi, r12
0x10b8ad:    lea     edx, [rax + rax + 4]
0x10b8b1:    xor     eax, eax
0x10b8b3:    call    r13
0x10ba3a:    mov     rax, qword ptr [rip + 0x1ef4f]
0x10ba41:    mov     esi, r14d
0x10ba44:    mov     rdi, r15
0x10ba47:    movzx   edx, byte ptr [rax + 8]
0x10ba4b:    movzx   eax, byte ptr [rax + 9]
0x10ba4f:    mov     byte ptr [rsp + 8], dl
0x10ba53:    mov     byte ptr [rsp + 9], al
0x10ba57:    call    0x104930
0x10b944:    mov     ebp, eax
0x10b946:    cmp     eax, -1
0x10b949:    je      0x10ba00
0x10b905:    mov     ebp, eax
0x10b907:    cmp     eax, -1
0x10b90a:    jne     0x10b89b
0x10b8b6:    mov     edi, ebp
0x10b8b8:    call    0x104ab0
0x10ba5c:    mov     ebp, eax
0x10ba5e:    cmp     eax, -1
0x10ba61:    je      0x10bae0
0x10ba00:    mov     edx, 0x30
0x10ba05:    mov     esi, r14d
0x10ba08:    mov     rdi, r15
0x10ba0b:    mov     word ptr [rsp + 8], dx
0x10ba10:    call    0x104930
0x10b94f:    mov     edx, dword ptr [rip + 0x1f043]
0x10b955:    mov     edi, ebp
0x10b957:    mov     rsi, r12
0x10b95a:    xor     eax, eax
0x10b95c:    imul    edx, dword ptr [rip + 0x1f039]
0x10b963:    add     edx, 4
0x10b966:    call    r13
0x10b8bd:    cmp     ebx, 2
0x10b8c0:    je      0x10b990
0x10bae0:    mov     eax, 0x30
0x10bae5:    mov     esi, r14d
0x10bae8:    mov     rdi, r15
0x10baeb:    mov     word ptr [rsp + 8], ax
0x10baf0:    call    0x104930
0x10ba63:    mov     edx, dword ptr [rip + 0x1ef2f]
0x10ba69:    mov     rsi, r12
0x10ba6c:    mov     edi, ebp
0x10ba6e:    xor     eax, eax
0x10ba70:    imul    edx, dword ptr [rip + 0x1ef25]
0x10ba77:    add     edx, 4

```



```

0x10ba7a:    call    r13
0x10ba15:    mov     ebp, eax
0x10ba17:    cmp     eax, -1
0x10ba1a:    jne     0x10b94f
0x10b969:    mov     edi, ebp
0x10b96b:    call    0x104ab0
0x10b990:    movsx   esi, byte ptr [r12 + 2]
0x10b996:    movsx   edi, byte ptr [r12 + 3]
0x10b99c:    call    0x10a7a0
0x10b8c6:    mov     eax, 1
0x10b8cb:    mov     rdx, qword ptr [rsp + 0x28]
0x10b8d0:    sub     rdx, qword ptr fs:[0x28]
0x10b8d9:    jne     0x10bb07
0x10baf5:    mov     ebp, eax
0x10baf7:    cmp     eax, -1
0x10bafa:    jne     0x10ba63
0x10ba7d:    mov     edi, ebp
0x10ba7f:    call    0x104ab0
0x10ba20:    xor     eax, eax
0x10ba22:    jmp     0x10b8cb
0x10b970:    jmp     0x10b8c6
0x10b9a1:    mov     edx, dword ptr [rip + 0x1f561]
0x10b9a7:    test    edx, edx
0x10b9a9:    jle     0x10b9e5
0x10b8df:    add     rsp, 0x38
0x10b8e3:    pop     rbx
0x10b8e4:    pop     rbp
0x10b8e5:    pop     r12
0x10b8e7:    pop     r13
0x10b8e9:    pop     r14
0x10b8eb:    pop     r15
0x10b8ed:    ret
0x10bb07:    call    0x1049d0
0x10bb00:    xor     eax, eax
0x10bb02:    jmp     0x10b8cb
0x10ba84:    mov     eax, dword ptr [rip + 0x1ef12]
0x10ba8a:    xor     esi, esi
0x10ba8c:    lea     edi, [rax - 1]
0x10ba8f:    call    0x10a7a0
0x10b8cb:    mov     rdx, qword ptr [rsp + 0x28]
0x10b8d0:    sub     rdx, qword ptr fs:[0x28]
0x10b8d9:    jne     0x10bb07
0x10b9e5:    mov     dword ptr [rip + 0x1f519], 0
0x10b9ef:    mov     eax, 1
0x10b9f4:    jmp     0x10b8cb
0x10b9ab:    xor     ebp, ebp
0x10b9ad:    lea     rbx, [rip + 0x1f56c]
0x10b9b4:    jmp     0x10b9cc

```

```

0x10ba94:    mov     edx, dword ptr [rip + 0x1f46e]
0x10ba9a:    test    edx, edx
0x10ba9c:    jle     0x10b9e5
0x10b9cc:    sub     edx, ebp
0x10b9ce:    movsxd  rsi, ebp
0x10b9d1:    mov     edi, 1
0x10b9d6:    add     rsi, rbx
0x10b9d9:    movsxd  rdx, edx
0x10b9dc:    call    0x118da0
0x10baa2:    xor     ebp, ebp
0x10baa4:    lea     rbx, [rip + 0x1f475]
0x10baab:    jmp     0x10bac0
0x10b9e1:    test    eax, eax
0x10b9e3:    jns     0x10b9c0
0x10bac0:    sub     edx, ebp
0x10bac2:    movsxd  rsi, ebp
0x10bac5:    mov     edi, 1
0x10baca:    add     rsi, rbx
0x10bacd:    movsxd  rdx, edx
0x10bad0:    call    0x118da0
0x10b9c0:    mov     edx, dword ptr [rip + 0x1f542]
0x10b9c6:    add     ebp, eax
0x10b9c8:    cmp     edx, ebp
0x10b9ca:    jle     0x10b9e5
0x10bad5:    test    eax, eax
0x10bad7:    jns     0x10bab0
0x10bad9:    jmp     0x10b9e5
0x10bab0:    mov     edx, dword ptr [rip + 0x1f452]
0x10bab6:    add     ebp, eax
0x10bab8:    cmp     edx, ebp
0x10baba:    jle     0x10b9e5
0x10b8ee:    nop
0x10b975:    nop     dword ptr [rax]
0x10b987:    nop     word ptr [rax + rax]
0x10b9b6:    nop     word ptr cs:[rax + rax]
0x10b9f9:    nop     dword ptr [rax]
0x10ba27:    nop     word ptr [rax + rax]
0x10baad:    nop     dword ptr [rax]
0x10bade:    nop
0x10bb0c:    nop     dword ptr [rax]
0x10bb10:    endbr64
0x10bb14:    mov     eax, dword ptr [rip + 0x1ee6e]
0x10bb1a:    test    eax, eax
0x10bb1c:    jne     0x10bb20
0x10bb1e:    ret
0x10bb20:    sub     rsp, 8
0x10bb24:    mov     esi, 1
0x10bb29:    xor     eax, eax

```

```

0x10bb2b:    call    0x10b7f0
0x10bb30:    mov     dword ptr [rip + 0x1f8ce], eax
0x10bb36:    add     rsp, 8
0x10bb3a:    ret
0x10bb1f:    nop
0x10bb3b:    nop     dword ptr [rax + rax]
0x10bb40:    endbr64
0x10bb44:    push    rbp
0x10bb45:    xor     eax, eax
0x10bb47:    mov     rbp, rdi
0x10bb4a:    call    0x10a6c0
0x10bb4f:    mov     edx, dword ptr [rip + 0x1ee33]
0x10bb55:    test    edx, edx
0x10bb57:    je      0x10bb94
0x10bb94:    xor     eax, eax
0x10bb96:    pop     rbp
0x10bb97:    jmp     0x10a300
0x10bb59:    mov     eax, dword ptr [rip + 0x1f8a5]
0x10bb5f:    test    eax, eax
0x10bb61:    je      0x10bb94
0x10bb63:    mov     edx, dword ptr [rip + 0x1ee33]
0x10bb69:    imul    edx, dword ptr [rip + 0x1ee28]
0x10bb70:    xor     esi, esi
0x10bb72:    mov     rdi, qword ptr [rip + 0x1f7af]
0x10bb79:    movsxd  rdx, edx
0x10bb7c:    call    0x104a50
0x10bb81:    xor     eax, eax
0x10bb83:    mov     esi, 2
0x10bb88:    mov     rdi, rbp
0x10bb8b:    call    0x10b7f0
0x10bb90:    test    eax, eax
0x10bb92:    jne     0x10bba0
0x10bba0:    mov     edx, dword ptr [rip + 0x1edf6]
0x10bba6:    imul    edx, dword ptr [rip + 0x1edeb]
0x10bbad:    xor     esi, esi
0x10bbaf:    mov     rdi, qword ptr [rip + 0x1f782]
0x10bbb6:    pop     rbp
0x10bbb7:    movsxd  rdx, edx
0x10bbbba:    jmp     0x104a50
0x10bb9c:    nop     dword ptr [rax]
0x10bbbf:    nop
0x10bbc0:    endbr64
0x10bbc4:    push    rbp
0x10bbc5:    mov     rbp, rdi
0x10bbc8:    push    rbx
0x10bbc9:    sub     rsp, 8
0x10bbcd:    cmp     dword ptr [rip + 0x1ed84], 1
0x10bbd4:    je      0x10bcd0

```

```

0x10bcd0:    lea     rdx, [rip + 0x1f6e9]
0x10bcd7:    mov     esi, 1
0x10bcdc:    mov     edi, 1
0x10bce1:    call    0x104d70
0x10bbda:    xor     eax, eax
0x10bbdc:    call    0x10a6c0
0x10bce6:    mov     dword ptr [rip + 0x1ec68], 0
0x10bcf0:    jmp     0x10bbda
0x10bbe1:    test    rbp, rbp
0x10bbe4:    je      0x10bbf4
0x10bbf4:    mov     eax, dword ptr [rip + 0x1eda2]
0x10bbfa:    mov     rdi, qword ptr [rip + 0x1e347]
0x10bc01:    lea     rdx, [rip - 0x2198]
0x10bc08:    lea     esi, [rax - 1]
0x10bc0b:    call    0x104820
0x10bbe6:    mov     eax, dword ptr [rip + 0x1ed9c]
0x10bbec:    test    eax, eax
0x10bbee:    jne     0x10bcc0
0x10bc10:    mov     edx, dword ptr [rip + 0x1f2f2]
0x10bc16:    test    edx, edx
0x10bc18:    jle     0x10bc4d
0x10bcc0:    mov     rdi, rbp
0x10bcc3:    call    0x10bb40
0x10bc4d:    mov     edi, dword ptr [rip + 0x1ed49]
0x10bc53:    xor     esi, esi
0x10bc55:    mov     dword ptr [rip + 0x1f2a9], 0
0x10bc5f:    call    0x10a7a0
0x10bc1a:    xor     ebp, ebp
0x10bc1c:    lea     rbx, [rip + 0x1f2fd]
0x10bc23:    jmp     0x10bc34
0x10bcc8:    jmp     0x10bbf4
0x10bc64:    mov     edx, dword ptr [rip + 0x1f29e]
0x10bc6a:    test    edx, edx
0x10bc6c:    jle     0x10bca5
0x10bc34:    sub     edx, ebp
0x10bc36:    movsxd  rsi, ebp
0x10bc39:    mov     edi, 1
0x10bc3e:    add     rsi, rbx
0x10bc41:    movsxd  rdx, edx
0x10bc44:    call    0x118da0
0x10bca5:    mov     dword ptr [rip + 0x1f259], 0
0x10bcaf:    add     rsp, 8
0x10bcb3:    mov     edi, 0xa
0x10bcb8:    pop     rbx
0x10bcb9:    pop     rbp
0x10bcba:    jmp     0x104810
0x10bc6e:    xor     ebp, ebp
0x10bc70:    lea     rbx, [rip + 0x1f2a9]

```

```

0x10bc77:      jmp      0x10bc8c
0x10bc49:      test     eax, eax
0x10bc4b:      jns       0x10bc28
0x10bc8c:      sub       edx, ebp
0x10bc8e:      movsxd    rsi, ebp
0x10bc91:      mov       edi, 1
0x10bc96:      add       rsi, rbx
0x10bc99:      movsxd    rdx, edx
0x10bc9c:      call      0x118da0
0x10bc28:      mov       edx, dword ptr [rip + 0x1f2da]
0x10bc2e:      add       ebp, eax
0x10bc30:      cmp       edx, ebp
0x10bc32:      jle       0x10bc4d
0x10bca1:      test     eax, eax
0x10bca3:      jns       0x10bc80
0x10bc80:      mov       edx, dword ptr [rip + 0x1f282]
0x10bc86:      add       ebp, eax
0x10bc88:      cmp       edx, ebp
0x10bc8a:      jle       0x10bca5
0x10bc25:      nop       dword ptr [rax]
0x10bc79:      nop       dword ptr [rax]
0x10bcbf:      nop
0x10bccd:      nop       dword ptr [rax]
0x10bcf5:      nop       word ptr cs:[rax + rax]
0x10bd00:      endbr64
0x10bd04:      push      r13
0x10bd06:      lea       r13, [rip + 0x1cc33]
0x10bd0d:      push      r12
0x10bd0f:      push      rbp
0x10bd10:      mov       rbp, rdi
0x10bd13:      lea       rdi, [rip + 0x14f14]
0x10bd1a:      push      rbx
0x10bd1b:      xor       ebx, ebx
0x10bd1d:      sub       rsp, 8
0x10bd21:      jmp       0x10bd2d
0x10bd2d:      mov       rsi, rbp
0x10bd30:      mov       r12d, ebx
0x10bd33:      call      0x104b30
0x10bd38:      test     eax, eax
0x10bd3a:      je        0x10bd60
0x10bd60:      lea       eax, [rbx - 8]
0x10bd63:      cmp       ebx, 7
0x10bd66:      cmovg     r12d, eax
0x10bd6a:      add       rsp, 8
0x10bd6e:      pop       rbx
0x10bd6f:      pop       rbp
0x10bd70:      mov       eax, r12d
0x10bd73:      pop       r12

```

```

0x10bd75:    pop     r13
0x10bd77:    ret
0x10bd3c:    add     rbx, 1
0x10bd40:    cmp     rbx, 0xa
0x10bd44:    jne     0x10bd28
0x10bd46:    add     rsp, 8
0x10bd4a:    mov     r12d, 0xffffffff
0x10bd50:    pop     rbx
0x10bd51:    mov     eax, r12d
0x10bd54:    pop     rbp
0x10bd55:    pop     r12
0x10bd57:    pop     r13
0x10bd59:    ret
0x10bd28:    mov     rdi, qword ptr [r13 + rbx*8]
0x10bd2d:    mov     rsi, rbp
0x10bd30:    mov     r12d, ebx
0x10bd33:    call    0x104b30
0x10bd23:    nop     dword ptr [rax + rax]
0x10bd5a:    nop     word ptr [rax + rax]
0x10bd78:    nop     dword ptr [rax + rax]
0x10bd80:    endbr64
0x10bd84:    push    r12
0x10bd86:    mov     r12, rdi
0x10bd89:    push    rbp
0x10bd8a:    lea     rbp, [rip + 0x1e1e7]
0x10bd91:    push    rbx
0x10bd92:    mov     ebx, 0x11
0x10bd97:    jmp     0x10bdac
0x10bdac:    mov     rdi, qword ptr [rbp]
0x10bdb0:    mov     rsi, r12
0x10bdb3:    call    0x104b30
0x10bdb8:    test    eax, eax
0x10bdba:    jne     0x10bda0
0x10bdbc:    movsxd  rbx, ebx
0x10bdbf:    lea     rax, [rip + 0x1df7a]
0x10bdc6:    shl     rbx, 5
0x10bdca:    mov     rax, qword ptr [rax + rbx + 8]
0x10bdcf:    pop     rbx
0x10bdd0:    pop     rbp
0x10bdd1:    pop     r12
0x10bdd3:    ret
0x10bda0:    add     ebx, 1
0x10bda3:    add     rbp, 0x20
0x10bda7:    cmp     ebx, 0x26
0x10bdaa:    je      0x10bdd8
0x10bdd8:    pop     rbx
0x10bdd9:    xor     eax, eax
0x10bddb:    pop     rbp

```

```

0x10bddc:    pop     r12
0x10bdde:    ret
0x10bd99:    nop     dword ptr [rax]
0x10bdd4:    nop     dword ptr [rax]
0x10bddf:    nop
0x10bde0:    endbr64
0x10bde4:    push    r14
0x10bde6:    lea     rdi, [rip + 0x1479d]
0x10bded:    push    r13
0x10bdef:    push    r12
0x10bdf1:    push    rbp
0x10bdf2:    push    rbx
0x10bdf3:    sub     rsp, 0xb0
0x10bdfa:    mov     rax, qword ptr fs:[0x28]
0x10be03:    mov     qword ptr [rsp + 0xa8], rax
0x10be0b:    xor     eax, eax
0x10be0d:    call    0x1047e0
0x10be12:    lea     rsi, [rsp + 0x10]
0x10be17:    mov     edi, 1
0x10be1c:    mov     rbp, rax
0x10be1f:    call    0x104eb0
0x10be24:    mov     rax, qword ptr [rsp + 0x38]
0x10be29:    shr     rax, 8
0x10be2d:    cmp     rax, 4
0x10be31:    sete    al
0x10be34:    movzx   eax, al
0x10be37:    mov     dword ptr [rip + 0x1eb4b], eax
0x10be3d:    test    rbp, rbp
0x10be40:    je      0x10c36d
0x10c36d:    mov     rcx, qword ptr [rip + 0x1e574]
0x10c374:    mov     rdi, qword ptr [rip + 0x1e505]
0x10c37b:    lea     rdx, [rip + 0x14a16]
0x10c382:    xor     eax, eax
0x10c384:    mov     esi, 1
0x10c389:    call    0x104e60
0x10be46:    mov     rdi, rbp
0x10be49:    call    0x1049b0
0x10c38e:    jmp     0x10c329
0x10be4e:    cmp     rax, 0x3f
0x10be52:    ja      0x10c308
0x10c329:    mov     rcx, qword ptr [rip + 0x1e550]
0x10c330:    mov     edx, 0x11
0x10c335:    mov     esi, 1
0x10c33a:    lea     rdi, [rip + 0x148f3]
0x10c341:    lea     rbp, [rip + 0x1d9d1]
0x10c348:    call    0x104e40
0x10be58:    lea     r12, [rip + 0x14ff1]
0x10be5f:    lea     rbx, [rip + 0x14d6f]

```

```

0x10be66:    lea     r14, [rip + 0x14e02]
0x10be6d:    lea     r13, [rip + 0x14e07]
0x10be74:    nop     dword ptr [rax]
0x10be78:    xor     edi, edi
0x10be7a:    mov     rsi, rbp
0x10be7d:    call    0x104e90
0x10c308:    mov     rcx, qword ptr [rip + 0x1e5d9]
0x10c30f:    mov     rdi, qword ptr [rip + 0x1e56a]
0x10c316:    lea     rdx, [rip + 0x14aab]
0x10c31d:    xor     eax, eax
0x10c31f:    mov     esi, 1
0x10c324:    call    0x104e60
0x10c34d:    jmp     0x10be58
0x10be82:    cmp     eax, -1
0x10be85:    je      0x10bf38
0x10bf38:    mov     rcx, qword ptr [rip + 0x1e9a9]
0x10bf3f:    mov     rdi, qword ptr [rip + 0x1e93a]
0x10bf46:    xor     eax, eax
0x10bf48:    lea     r8, [rip + 0x1ddc1]
0x10bf4f:    lea     rdx, [rip + 0x14ea2]
0x10bf56:    mov     esi, 1
0x10bf5b:    call    0x104e60
0x10be8b:    test    eax, eax
0x10be8d:    jne     0x10bff8
0x10bf60:    mov     rdi, qword ptr [rip + 0x1e919]
0x10bf67:    xor     eax, eax
0x10bf69:    mov     rcx, qword ptr [rip + 0x1e978]
0x10bf70:    lea     r8, [rip + 0x1dd89]
0x10bf77:    lea     rdx, [rip + 0x14ea2]
0x10bf7e:    mov     esi, 1
0x10bf83:    call    0x104e60
0x10be93:    mov     rdi, qword ptr [rip + 0x1e9e6]
0x10be9a:    mov     esi, 1
0x10be9f:    mov     r8, rbp
0x10bea2:    mov     rdx, r12
0x10bea5:    mov     rcx, qword ptr [rip + 0x1ea3c]
0x10beac:    lea     r9, [rip + 0x1de5d]
0x10beb3:    call    0x104e60
0x10bff8:    mov     rdi, rbp
0x10bffb:    lea     r13, [rsp + 8]
0x10c000:    call    0x11d7a0
0x10bf88:    mov     edi, 1
0x10bf8d:    call    0x104e20
0x10beb8:    mov     rsi, rbx
0x10bebb:    mov     rdi, rbp
0x10bebe:    call    0x104b30
0x10c005:    mov     edi, 0x800
0x10c00a:    mov     qword ptr [rip + 0x1e93f], rax

```



```

0x10c011:    call    0x11d670
0x10bec3:    test    eax, eax
0x10bec5:    je      0x10bf98
0x10c016:    mov     rdi, qword ptr [rip + 0x1de03]
0x10c01d:    mov     rsi, r13
0x10c020:    mov     qword ptr [rsp + 8], rax
0x10c025:    call    0x104ca0
0x10bf98:    mov     rcx, qword ptr [rip + 0x1e949]
0x10bf9f:    mov     rdi, qword ptr [rip + 0x1e8da]
0x10bfa6:    lea     rdx, [rip + 0x14c99]
0x10bfad:    mov     esi, 1
0x10bfb2:    lea     rbp, [rip + 0x14c21]
0x10bfb9:    call    0x104e60
0x10becb:    mov     rdi, rbp
0x10bece:    call    0x109940
0x10c02a:    xor     edx, edx
0x10c02c:    test    rax, rax
0x10c02f:    je      0x10c034
0x10bfbe:    jmp     0x10be78
0x10bed3:    test    eax, eax
0x10bed5:    jne     0x10bfc8
0x10c034:    mov     rdi, qword ptr [rip + 0x1de25]
0x10c03b:    mov     rsi, r13
0x10c03e:    mov     byte ptr [rip + 0x1e904], dl
0x10c044:    call    0x104ca0
0x10c031:    movzx   edx, byte ptr [rax]
0x10c034:    mov     rdi, qword ptr [rip + 0x1de25]
0x10c03b:    mov     rsi, r13
0x10c03e:    mov     byte ptr [rip + 0x1e904], dl
0x10c044:    call    0x104ca0
0x10be78:    xor     edi, edi
0x10be7a:    mov     rsi, rbp
0x10be7d:    call    0x104e90
0x10bedb:    mov     rsi, r14
0x10bede:    mov     rdi, rbp
0x10bee1:    call    0x104b30
0x10bfc8:    mov     rcx, qword ptr [rip + 0x1e919]
0x10bfcf:    mov     rdi, qword ptr [rip + 0x1e8aa]
0x10bfd6:    lea     rdx, [rip + 0x14c7d]
0x10bfdd:    xor     eax, eax
0x10bfdf:    mov     esi, 1
0x10bfe4:    lea     rbp, [rip + 0x14bf6]
0x10bfeb:    call    0x104e60
0x10c049:    mov     rdi, qword ptr [rip + 0x1ddf0]
0x10c050:    mov     rsi, r13
0x10c053:    mov     qword ptr [rip + 0x1e8e6], rax
0x10c05a:    call    0x104ca0
0x10bee6:    test    eax, eax

```

```

0x10bee8:    je      0x10bf01
0x10bff0:    jmp     0x10be78
0x10c05f:    cmp     qword ptr [rip + 0x1e8d9], 0
0x10c067:    mov     qword ptr [rip + 0x1e8ca], rax
0x10c06e:    je      0x10c2c0
0x10bf01:    mov     rdi, qword ptr [rip + 0x1e978]
0x10bf08:    mov     rcx, qword ptr [rip + 0x1e9d9]
0x10bf0f:    mov     esi, 1
0x10bf14:    xor     eax, eax
0x10bf16:    mov     rdx, r13
0x10bf19:    lea     rbp, [rip + 0x14cba]
0x10bf20:    call    0x104e60
0x10beea:    lea     rsi, [rip + 0x14d84]
0x10bef1:    mov     rdi, rbp
0x10bef4:    call    0x104b30
0x10c2c0:    mov     qword ptr [rip + 0x1e66d], 0
0x10c2cb:    mov     qword ptr [rip + 0x1e66a], 0
0x10c2d6:    jmp     0x10c07d
0x10c074:    test    rax, rax
0x10c077:    je      0x10c2c0
0x10bf25:    xor     edi, edi
0x10bf27:    mov     rsi, rbp
0x10bf2a:    call    0x104e90
0x10bef9:    test    eax, eax
0x10befb:    jne     0x10c3b4
0x10c07d:    mov     rdi, qword ptr [rip + 0x1dcbc]
0x10c084:    mov     rsi, r13
0x10c087:    call    0x104ca0
0x10bf2f:    cmp     eax, -1
0x10bf32:    jne     0x10be8b
0x10c3b4:    mov     edi, 1
0x10c3b9:    call    0x104e20
0x10c08c:    mov     rdi, qword ptr [rip + 0x1dcde]
0x10c093:    mov     rsi, r13
0x10c096:    mov     qword ptr [rip + 0x1dcab], rax
0x10c09d:    call    0x104ca0
0x10c0a2:    mov     rdi, qword ptr [rip + 0x1dcb7]
0x10c0a9:    mov     rsi, r13
0x10c0ac:    mov     qword ptr [rip + 0x1dcd5], rax
0x10c0b3:    call    0x104ca0
0x10c0b8:    cmp     qword ptr [rip + 0x1dc88], 0
0x10c0c0:    mov     qword ptr [rip + 0x1dca1], rax
0x10c0c7:    je      0x10c352
0x10c352:    mov     qword ptr [rip + 0x1da2b], 0
0x10c35d:    mov     qword ptr [rip + 0x1da00], 0
0x10c368:    jmp     0x10c0cd
0x10c0cd:    mov     rdi, qword ptr [rip + 0x1ddac]
0x10c0d4:    mov     rsi, r13

```

```

0x10c0d7:    call    0x104ca0
0x10c0dc:    xor     edx, edx
0x10c0de:    mov     qword ptr [rip + 0x1dda3], rax
0x10c0e5:    test    rax, rax
0x10c0e8:    je      0x10c0ff
0x10c0ff:    mov     rdi, qword ptr [rip + 0x1dd9a]
0x10c106:    mov     dword ptr [rip + 0x1ddc4], edx
0x10c10c:    call    0x104c50
0x10c0ea:    mov     rdi, qword ptr [rip + 0x1ddcf]
0x10c0f1:    mov     qword ptr [rip + 0x1dc70], rax
0x10c0f8:    call    0x104c90
0x10c111:    mov     dword ptr [rip + 0x1dd99], eax
0x10c117:    test    eax, eax
0x10c119:    js      0x10c13c
0x10c0fd:    mov     edx, eax
0x10c0ff:    mov     rdi, qword ptr [rip + 0x1dd9a]
0x10c106:    mov     dword ptr [rip + 0x1ddc4], edx
0x10c10c:    call    0x104c50
0x10c13c:    mov     rdi, qword ptr [rip + 0x1dd9d]
0x10c143:    call    0x104c50
0x10c11b:    mov     qword ptr [rip + 0x1dc62], 0
0x10c126:    mov     qword ptr [rip + 0x1dc37], 0
0x10c131:    mov     qword ptr [rip + 0x1dc0c], 0
0x10c13c:    mov     rdi, qword ptr [rip + 0x1dd9d]
0x10c143:    call    0x104c50
0x10c148:    mov     rdi, qword ptr [rip + 0x1ddb1]
0x10c14f:    mov     dword ptr [rip + 0x1dd9b], eax
0x10c155:    call    0x104c50
0x10c15a:    mov     rdi, qword ptr [rip + 0x1dc3f]
0x10c161:    mov     rsi, r13
0x10c164:    mov     dword ptr [rip + 0x1dda6], eax
0x10c16a:    call    0x104ca0
0x10c16f:    mov     rdi, qword ptr [rip + 0x1dc4a]
0x10c176:    mov     rsi, r13
0x10c179:    mov     qword ptr [rip + 0x1dc28], rax
0x10c180:    call    0x104ca0
0x10c185:    cmp     qword ptr [rip + 0x1dc1b], 0
0x10c18d:    mov     qword ptr [rip + 0x1dc34], rax
0x10c194:    je      0x10c2a0
0x10c2a0:    mov     qword ptr [rip + 0x1dafd], 0
0x10c2ab:    mov     qword ptr [rip + 0x1db12], 0
0x10c2b6:    jmp     0x10c1a3
0x10c19a:    test    rax, rax
0x10c19d:    je      0x10c2a0
0x10c1a3:    mov     rdi, qword ptr [rip + 0x1dc36]
0x10c1aa:    mov     rsi, r13
0x10c1ad:    lea     rbp, [rip + 0x1ddac]
0x10c1b4:    lea     rbx, [rbp - 0x220]

```

```

0x10c1bb:    call    0x104ca0
0x10c1c0:    mov     rdi, qword ptr [rip + 0x1dc39]
0x10c1c7:    mov     rsi, r13
0x10c1ca:    lea     r12, [rbx + 0x4c0]
0x10c1d1:    mov     qword ptr [rip + 0x1dc10], rax
0x10c1d8:    call    0x104ca0
0x10c1dd:    mov     rdi, qword ptr [rip + 0x1dd3c]
0x10c1e4:    mov     rsi, r13
0x10c1e7:    mov     qword ptr [rip + 0x1dc1a], rax
0x10c1ee:    call    0x104ca0
0x10c1f3:    mov     rdi, qword ptr [rip + 0x1dd46]
0x10c1fa:    mov     rsi, r13
0x10c1fd:    mov     qword ptr [rip + 0x1dd24], rax
0x10c204:    call    0x104ca0
0x10c209:    mov     qword ptr [rip + 0x1dd38], rax
0x10c210:    mov     rdi, qword ptr [rbp]
0x10c214:    mov     rsi, r13
0x10c217:    add     rbp, 0x20
0x10c21b:    call    0x104ca0
0x10c220:    mov     qword ptr [rbp - 0x18], rax
0x10c224:    cmp     rbp, r12
0x10c227:    jne     0x10c210
0x10c229:    xor     ebp, ebp
0x10c22b:    lea     r13, [rip + 0x161b3]
0x10c232:    lea     r14, [rip + 0x14c57]
0x10c239:    jmp     0x10c249
0x10c210:    mov     rdi, qword ptr [rbp]
0x10c214:    mov     rsi, r13
0x10c217:    add     rbp, 0x20
0x10c21b:    call    0x104ca0
0x10c249:    cmp     qword ptr [rbx + 8], 0
0x10c24e:    jne     0x10c240
0x10c250:    mov     eax, dword ptr [rbx + 0x14]
0x10c253:    test    eax, eax
0x10c255:    jne     0x10c2e0
0x10c240:    add     rbx, 0x20
0x10c244:    cmp     rbx, r12
0x10c247:    je      0x10c268
0x10c25b:    mov     qword ptr [rbx + 8], r13
0x10c25f:    add     rbx, 0x20
0x10c263:    cmp     rbx, r12
0x10c266:    jne     0x10c249
0x10c2e0:    mov     r8, qword ptr [rbx]
0x10c2e3:    mov     rcx, qword ptr [rip + 0x1e5fe]
0x10c2ea:    mov     rdx, r14
0x10c2ed:    xor     eax, eax
0x10c2ef:    mov     rdi, qword ptr [rip + 0x1e58a]
0x10c2f6:    mov     esi, 1

```

```

0x10c2fb:    add     ebp, 1
0x10c2fe:    call    0x104e60
0x10c268:    test    ebp, ebp
0x10c26a:    jne     0x10c390
0x10c303:    jmp     0x10c240
0x10c270:    mov     rax, qword ptr [rsp + 0xa8]
0x10c278:    sub     rax, qword ptr fs:[0x28]
0x10c281:    jne     0x10c3be
0x10c390:    mov     rcx, qword ptr [rip + 0x1e551]
0x10c397:    mov     rdi, qword ptr [rip + 0x1e4e2]
0x10c39e:    mov     r8d, ebp
0x10c3a1:    xor     eax, eax
0x10c3a3:    lea     rdx, [rip + 0x14b16]
0x10c3aa:    mov     esi, 1
0x10c3af:    call    0x104e60
0x10c287:    add     rsp, 0xb0
0x10c28e:    pop     rbx
0x10c28f:    pop     rbp
0x10c290:    pop     r12
0x10c292:    pop     r13
0x10c294:    pop     r14
0x10c296:    ret
0x10c3be:    call    0x1049d0
0x10bf92:    nop     word ptr [rax + rax]
0x10bfc3:    nop     dword ptr [rax + rax]
0x10bff5:    nop     dword ptr [rax]
0x10c23b:    nop     dword ptr [rax + rax]
0x10c297:    nop     word ptr [rax + rax]
0x10c2bb:    nop     dword ptr [rax + rax]
0x10c2db:    nop     dword ptr [rax + rax]
0x10c3c3:    nop     word ptr cs:[rax + rax]
0x10c3ce:    nop
0x10c3d0:    endbr64
0x10c3d4:    push    rbx
0x10c3d5:    mov     ebx, edi
0x10c3d7:    xor     edi, edi
0x10c3d9:    call    0x1048f0
0x10c3de:    test    eax, eax
0x10c3e0:    je      0x10c453
0x10c453:    mov     rdi, qword ptr [rip + 0x1e426]
0x10c45a:    mov     rcx, qword ptr [rip + 0x1e487]
0x10c461:    lea     rdx, [rip + 0x14a88]
0x10c468:    xor     eax, eax
0x10c46a:    mov     esi, 1
0x10c46f:    call    0x104e60
0x10c3e2:    mov     edi, 1
0x10c3e7:    call    0x1048f0
0x10c474:    mov     edi, 1

```

```

0x10c479:    call    0x104e20
0x10c3ec:    test    eax, eax
0x10c3ee:    je      0x10c453
0x10c3f0:    mov     edi, 1
0x10c3f5:    call    0x104cd0
0x10c3fa:    mov     qword ptr [rip + 0x1e58f], rax
0x10c401:    test    rax, rax
0x10c404:    je      0x10c47e
0x10c47e:    mov     rdi, qword ptr [rip + 0x1e3fb]
0x10c485:    mov     rcx, qword ptr [rip + 0x1e45c]
0x10c48c:    mov     esi, 1
0x10c491:    lea     rdx, [rip + 0x147f8]
0x10c498:    call    0x104e60
0x10c406:    lea     rsi, [rip + 0x1efb3]
0x10c40d:    mov     edi, 1
0x10c412:    call    0x104d60
0x10c49d:    mov     edi, 1
0x10c4a2:    call    0x104e20
0x10c417:    mov     edi, 0x40
0x10c41c:    call    0x11d670
0x10c421:    mov     dword ptr [rip + 0x1efd5], ebx
0x10c427:    pop     rbx
0x10c428:    mov     qword ptr [rip + 0x1ef31], rax
0x10c42f:    mov     qword ptr [rip + 0x1e52a], rax
0x10c436:    xor     eax, eax
0x10c438:    mov     qword ptr [rip + 0x1e52d], 0
0x10c443:    mov     qword ptr [rip + 0x1e51a], 0
0x10c44e:    jmp     0x10bde0
0x10c4a7:    nop     word ptr [rax + rax]
0x10c4b0:    endbr64
0x10c4b4:    push    r12
0x10c4b6:    push    rbp
0x10c4b7:    mov     rbp, rdi
0x10c4ba:    mov     rdi, qword ptr [rip + 0x1e48f]
0x10c4c1:    push    rbx
0x10c4c2:    call    0x109940
0x10c4c7:    test    eax, eax
0x10c4c9:    jne     0x10c4d0
0x10c4cb:    pop     rbx
0x10c4cc:    pop     rbp
0x10c4cd:    pop     r12
0x10c4cf:    ret
0x10c4d0:    mov     rdi, rbp
0x10c4d3:    call    0x1049b0
0x10c4d8:    lea     rdi, [rax + 0x81]
0x10c4df:    mov     rbx, rax
0x10c4e2:    call    0x11d670
0x10c4e7:    mov     rdi, rbp

```

```

0x10c4ea:    mov     r12, rax
0x10c4ed:    call   0x11d7a0
0x10c4f2:    mov     rsi, rbx
0x10c4f5:    add     ebx, 0x7f
0x10c4f8:    mov     rbp, rax
0x10c4fb:    mov     rdi, rax
0x10c4fe:    call   0x1184d0
0x10c503:    push    7
0x10c505:    lea     rcx, [rip + 0x147a2]
0x10c50c:    xor     eax, eax
0x10c50e:    push    rbp
0x10c50f:    mov     rdx, -1
0x10c516:    mov     esi, 1
0x10c51b:    mov     rdi, r12
0x10c51e:    lea     r9, [rip + 0x14040]
0x10c525:    mov     r8d, 0x1b
0x10c52b:    call   0x104f10
0x10c530:    mov     eax, dword ptr [rip + 0x1e462]
0x10c536:    pop     rdx
0x10c537:    pop     rcx
0x10c538:    add     eax, eax
0x10c53a:    cmp     ebx, eax
0x10c53c:    jl      0x10c555
0x10c555:    mov     rdi, r12
0x10c558:    call   0x1049b0
0x10c53e:    cdqe
0x10c540:    mov     byte ptr [r12 + rax], 7
0x10c545:    mov     eax, dword ptr [rip + 0x1e44d]
0x10c54b:    add     eax, eax
0x10c54d:    cdqe
0x10c54f:    mov     byte ptr [r12 + rax + 1], 0
0x10c555:    mov     rdi, r12
0x10c558:    call   0x1049b0
0x10c55d:    mov     rsi, r12
0x10c560:    mov     edi, 1
0x10c565:    mov     rdx, rax
0x10c568:    call   0x118da0
0x10c56d:    mov     rdi, rbp
0x10c570:    call   0x118d10
0x10c575:    mov     rdi, r12
0x10c578:    call   0x118d10
0x10c57d:    mov     rdi, qword ptr [rip + 0x1e2bc]
0x10c584:    pop     rbx
0x10c585:    pop     rbp
0x10c586:    pop     r12
0x10c588:    jmp     0x104c40
0x10c58d:    nop     dword ptr [rax]
0x10c590:    endbr64

```

```

0x10c594:    sub     rsp, 8
0x10c598:    mov     edi, 0x18
0x10c59d:    call    0x11d670
0x10c5a2:    mov     qword ptr [rax], 0
0x10c5a9:    mov     qword ptr [rax + 8], 0
0x10c5b1:    add     rsp, 8
0x10c5b5:    ret
0x10c5b6:    nop     word ptr cs:[rax + rax]
0x10c5c0:    endbr64
0x10c5c4:    test    rdi, rdi
0x10c5c7:    je      0x10c5d0
0x10c5d0:    ret
0x10c5c9:    jmp     0x118d10
0x10c5ce:    nop
0x10c5d1:    nop     word ptr cs:[rax + rax]
0x10c5dc:    nop     dword ptr [rax]
0x10c5e0:    endbr64
0x10c5e4:    mov     dword ptr [rdi], esi
0x10c5e6:    mov     dword ptr [rdi + 4], edx
0x10c5e9:    mov     dword ptr [rdi + 8], ecx
0x10c5ec:    mov     dword ptr [rdi + 0xc], r8d
0x10c5f0:    ret
0x10c5f1:    nop     word ptr cs:[rax + rax]
0x10c5fc:    nop     dword ptr [rax]
0x10c600:    endbr64
0x10c604:    mov     rax, rdi
0x10c607:    mov     rdi, rsi
0x10c60a:    mov     esi, edx
0x10c60c:    mov     ecx, dword ptr [rax + 0x10]
0x10c60f:    lea     r8d, [rcx + rdx]
0x10c613:    mov     edx, dword ptr [rax + 0xc]
0x10c616:    mov     dword ptr [rax + 0x10], r8d
0x10c61a:    cmp     edx, ecx
0x10c61c:    jle     0x10c640
0x10c640:    xor     eax, eax
0x10c642:    ret
0x10c61e:    mov     r9d, dword ptr [rax + 8]
0x10c622:    cmp     dword ptr [rax + 0x14], r9d
0x10c626:    jge     0x10c640
0x10c628:    cmp     r8d, edx
0x10c62b:    jle     0x10c631
0x10c631:    jmp     0x10a1a0
0x10c62d:    sub     edx, ecx
0x10c62f:    mov     esi, edx
0x10c631:    jmp     0x10a1a0
0x10c636:    nop     word ptr cs:[rax + rax]
0x10c643:    nop     word ptr cs:[rax + rax]
0x10c64e:    nop

```



```

0x10c650:    endbr64
0x10c654:    mov     ecx, dword ptr [rdi + 0x10]
0x10c657:    lea     edx, [rcx + 1]
0x10c65a:    mov     dword ptr [rdi + 0x10], edx
0x10c65d:    cmp     edx, dword ptr [rdi + 0xc]
0x10c660:    jg      0x10c66a
0x10c662:    mov     ecx, dword ptr [rdi + 8]
0x10c665:    cmp     dword ptr [rdi + 0x14], ecx
0x10c668:    jl      0x10c670
0x10c66a:    xor     eax, eax
0x10c66c:    ret
0x10c670:    mov     edi, esi
0x10c672:    jmp     0x10a230
0x10c66d:    nop     dword ptr [rax]
0x10c677:    nop     word ptr [rax + rax]
0x10c680:    endbr64
0x10c684:    mov     rax, rdi
0x10c687:    mov     edi, esi
0x10c689:    mov     dword ptr [rax + 0x10], edx
0x10c68c:    add     edx, dword ptr [rax]
0x10c68e:    mov     dword ptr [rax + 0x14], esi
0x10c691:    add     edi, dword ptr [rax + 4]
0x10c694:    mov     esi, edx
0x10c696:    jmp     0x10a4b0
0x10c69b:    nop     dword ptr [rax + rax]
0x10c6a0:    endbr64
0x10c6a4:    mov     eax, dword ptr [rdi]
0x10c6a6:    ret
0x10c6a7:    nop     word ptr [rax + rax]
0x10c6b0:    endbr64
0x10c6b4:    mov     eax, dword ptr [rdi + 4]
0x10c6b7:    ret
0x10c6b8:    nop     dword ptr [rax + rax]
0x10c6c0:    endbr64
0x10c6c4:    mov     eax, dword ptr [rdi + 8]
0x10c6c7:    ret
0x10c6c8:    nop     dword ptr [rax + rax]
0x10c6d0:    endbr64
0x10c6d4:    mov     eax, dword ptr [rdi + 0xc]
0x10c6d7:    ret
0x10c6d8:    nop     dword ptr [rax + rax]
0x10c6e0:    push    r14
0x10c6e2:    push    r13
0x10c6e4:    push    r12
0x10c6e6:    mov     r12, qword ptr [rip + 0x1ed1b]
0x10c6ed:    push    rbp
0x10c6ee:    push    rbx
0x10c6ef:    mov     r13, qword ptr [r12 + 0x20]

```

```

0x10c6f4:    mov     rbx, qword ptr [r12 + 0x18]
0x10c6f9:    cmp     r13, rbx
0x10c6fc:    je      0x10c761
0x10c761:    pop     rbx
0x10c762:    pop     rbp
0x10c763:    pop     r12
0x10c765:    pop     r13
0x10c767:    pop     r14
0x10c769:    ret
0x10c6fe:    mov     ebp, edi
0x10c700:    jbe     0x10c770
0x10c770:    mov     rax, rbx
0x10c773:    mov     rbx, r13
0x10c776:    mov     r13, rax
0x10c779:    test    bpl, 1
0x10c77d:    je      0x10c75c
0x10c702:    test    bpl, 1
0x10c706:    je      0x10c75c
0x10c75c:    and     ebp, 2
0x10c75f:    jne     0x10c788
0x10c77f:    jmp     0x10c708
0x10c708:    mov     rdi, qword ptr [r12 + 0x60]
0x10c70d:    test    rdi, rdi
0x10c710:    je      0x10c71e
0x10c788:    mov     rax, qword ptr [rip + 0x1ec79]
0x10c78f:    mov     rdx, rbx
0x10c792:    sub     rdx, r13
0x10c795:    add     qword ptr [rax + 0x48], rdx
0x10c799:    mov     rdi, qword ptr [rax + 0x58]
0x10c79d:    add     qword ptr [rax + 0x38], rdx
0x10c7a1:    mov     qword ptr [rax + 0x18], rbx
0x10c7a5:    lea     rsi, [rdi + r13]
0x10c7a9:    add     rdi, rbx
0x10c7ac:    mov     qword ptr [rax + 0x20], rbx
0x10c7b0:    call    0x1048b0
0x10c71e:    mov     r14, r13
0x10c721:    sub     r14, rbx
0x10c724:    lea     rdi, [r14 + 1]
0x10c728:    call    0x11d670
0x10c712:    call    0x118d10
0x10c7b5:    mov     rbx, qword ptr [rip + 0x1ec4c]
0x10c7bc:    mov     rax, qword ptr [rbx + 0x38]
0x10c7c0:    mov     rdi, qword ptr [rbx + 0x58]
0x10c7c4:    lea     rsi, [rax + 1]
0x10c7c8:    mov     qword ptr [rbx + 0x50], rsi
0x10c7cc:    call    0x11d690
0x10c72d:    mov     rdx, r14
0x10c730:    mov     qword ptr [r12 + 0x60], rax

```

```

0x10c735:    mov     rax, qword ptr [rip + 0x1eccc]
0x10c73c:    mov     rsi, qword ptr [rax + 0x58]
0x10c740:    mov     rdi, qword ptr [rax + 0x60]
0x10c744:    add     rsi, rbx
0x10c747:    call    0x104bb0
0x10c717:    mov     r12, qword ptr [rip + 0x1ecea]
0x10c71e:    mov     r14, r13
0x10c721:    sub     r14, rbx
0x10c724:    lea     rdi, [r14 + 1]
0x10c728:    call    0x11d670
0x10c7d1:    mov     qword ptr [rbx + 0x58], rax
0x10c7d5:    pop     rbx
0x10c7d6:    pop     rbp
0x10c7d7:    pop     r12
0x10c7d9:    pop     r13
0x10c7db:    pop     r14
0x10c7dd:    ret
0x10c74c:    mov     rax, qword ptr [rip + 0x1ecb5]
0x10c753:    mov     rax, qword ptr [rax + 0x60]
0x10c757:    mov     byte ptr [rax + r14], 0
0x10c75c:    and     ebp, 2
0x10c75f:    jne     0x10c788
0x10c76a:    nop     word ptr [rax + rax]
0x10c781:    nop     dword ptr [rax]
0x10c7de:    nop
0x10c7e0:    endbr64
0x10c7e4:    test    rdi, rdi
0x10c7e7:    je      0x10c818
0x10c818:    ret
0x10c7e9:    push    rbp
0x10c7ea:    mov     rbp, rdi
0x10c7ed:    mov     rdi, qword ptr [rdi + 0x58]
0x10c7f1:    test    rdi, rdi
0x10c7f4:    je      0x10c7fb
0x10c7fb:    mov     rdi, qword ptr [rbp + 0x60]
0x10c7ff:    test    rdi, rdi
0x10c802:    je      0x10c809
0x10c7f6:    call    0x118d10
0x10c809:    mov     rdi, rbp
0x10c80c:    pop     rbp
0x10c80d:    jmp     0x118d10
0x10c804:    call    0x118d10
0x10c812:    nop     word ptr [rax + rax]
0x10c819:    nop     dword ptr [rax]
0x10c820:    endbr64
0x10c824:    mov     rax, qword ptr [rip + 0x1ebdd]
0x10c82b:    push    rbp
0x10c82c:    mov     rdi, qword ptr [rax + 0x68]

```

```

0x10c830:    xor     eax, eax
0x10c832:    call    0x11aac0
0x10c837:    mov     rax, qword ptr [rip + 0x1ebca]
0x10c83e:    mov     rdi, qword ptr [rax]
0x10c841:    call    0x10c5c0
0x10c846:    mov     rbp, qword ptr [rip + 0x1ebbb]
0x10c84d:    test    rbp, rbp
0x10c850:    je      0x10c876
0x10c876:    mov     qword ptr [rip + 0x1eb87], 0
0x10c881:    pop     rbp
0x10c882:    ret
0x10c852:    mov     rdi, qword ptr [rbp + 0x58]
0x10c856:    test    rdi, rdi
0x10c859:    je      0x10c860
0x10c860:    mov     rdi, qword ptr [rbp + 0x60]
0x10c864:    test    rdi, rdi
0x10c867:    je      0x10c86e
0x10c85b:    call    0x118d10
0x10c86e:    mov     rdi, rbp
0x10c871:    call    0x118d10
0x10c869:    call    0x118d10
0x10c883:    nop     word ptr cs:[rax + rax]
0x10c88e:    nop
0x10c890:    endbr64
0x10c894:    mov     rax, qword ptr [rip + 0x1eb6d]
0x10c89b:    movsxd  rcx, edi
0x10c89e:    mov     edx, esi
0x10c8a0:    mov     r8, rcx
0x10c8a3:    mov     qword ptr [rax + 0x28], rcx
0x10c8a7:    movsxd  rcx, esi
0x10c8aa:    mov     rdi, qword ptr [rax]
0x10c8ad:    xor     esi, esi
0x10c8af:    mov     qword ptr [rax + 0x30], rcx
0x10c8b3:    mov     ecx, 1
0x10c8b8:    jmp     0x10c5e0
0x10c8bd:    nop     dword ptr [rax]
0x10c8c0:    endbr64
0x10c8c4:    push    r12
0x10c8c6:    mov     edi, 0x70
0x10c8cb:    push    rbx
0x10c8cc:    sub     rsp, 8
0x10c8d0:    call    0x11d670
0x10c8d5:    mov     rbx, qword ptr [rip + 0x1eb2c]
0x10c8dc:    mov     r12, rax
0x10c8df:    movdqu  xmm0, xmmword ptr [rbx]
0x10c8e3:    movups  xmmword ptr [rax], xmm0
0x10c8e6:    movdqu  xmm1, xmmword ptr [rbx + 0x10]
0x10c8eb:    movups  xmmword ptr [rax + 0x10], xmm1

```

```

0x10c8ef:    movdqu  xmm2, xmmword ptr [rbx + 0x20]
0x10c8f4:    movups  xmmword ptr [rax + 0x20], xmm2
0x10c8f8:    movdqu  xmm3, xmmword ptr [rbx + 0x30]
0x10c8fd:    movups  xmmword ptr [rax + 0x30], xmm3
0x10c901:    movdqu  xmm4, xmmword ptr [rbx + 0x40]
0x10c906:    movups  xmmword ptr [rax + 0x40], xmm4
0x10c90a:    movdqu  xmm5, xmmword ptr [rbx + 0x50]
0x10c90f:    movups  xmmword ptr [rax + 0x50], xmm5
0x10c913:    movdqu  xmm6, xmmword ptr [rbx + 0x60]
0x10c918:    cmp     qword ptr [rax + 0x58], 0
0x10c91d:    movups  xmmword ptr [rax + 0x60], xmm6
0x10c921:    je      0x10c94a
0x10c94a:    mov     rdi, qword ptr [r12 + 0x60]
0x10c94f:    test    rdi, rdi
0x10c952:    je      0x10c964
0x10c923:    mov     rdi, qword ptr [rax + 0x50]
0x10c927:    call    0x11d670
0x10c964:    add     rsp, 8
0x10c968:    mov     rax, r12
0x10c96b:    pop     rbx
0x10c96c:    pop     r12
0x10c96e:    ret
0x10c954:    mov     rbx, qword ptr [rip + 0x1eaad]
0x10c95b:    call    0x11d7a0
0x10c92c:    mov     rdx, qword ptr [r12 + 0x50]
0x10c931:    mov     qword ptr [rbx + 0x58], rax
0x10c935:    mov     rax, qword ptr [rip + 0x1eacc]
0x10c93c:    mov     rsi, qword ptr [r12 + 0x58]
0x10c941:    mov     rdi, qword ptr [rax + 0x58]
0x10c945:    call    0x104bb0
0x10c960:    mov     qword ptr [rbx + 0x60], rax
0x10c964:    add     rsp, 8
0x10c968:    mov     rax, r12
0x10c96b:    pop     rbx
0x10c96c:    pop     r12
0x10c96e:    ret
0x10c96f:    nop
0x10c970:    endbr64
0x10c974:    test    rdi, rdi
0x10c977:    je      0x10c9f0
0x10c9f0:    ret
0x10c979:    push    r13
0x10c97b:    push    r12
0x10c97d:    push    rbp
0x10c97e:    push    rbx
0x10c97f:    mov     rbx, rdi
0x10c982:    sub     rsp, 8
0x10c986:    mov     rbp, qword ptr [rip + 0x1ea7b]

```

```

0x10c98d:    mov     rdi, qword ptr [rbp + 0x58]
0x10c991:    mov     r13, qword ptr [rbp + 0x28]
0x10c995:    mov     r12, qword ptr [rbp + 0x30]
0x10c999:    test    rdi, rdi
0x10c99c:    je      0x10c9a3
0x10c9a3:    mov     rdi, qword ptr [rbp + 0x60]
0x10c9a7:    test    rdi, rdi
0x10c9aa:    je      0x10c9b1
0x10c99e:    call    0x118d10
0x10c9b1:    mov     rdi, rbp
0x10c9b4:    call    0x118d10
0x10c9ac:    call    0x118d10
0x10c9b9:    movsxd  rax, r13d
0x10c9bc:    mov     edx, r12d
0x10c9bf:    movsxd  r12, r12d
0x10c9c2:    mov     qword ptr [rbx + 0x28], rax
0x10c9c6:    mov     rdi, qword ptr [rbx]
0x10c9c9:    mov     r8d, r13d
0x10c9cc:    mov     ecx, 1
0x10c9d1:    mov     qword ptr [rbx + 0x30], r12
0x10c9d5:    xor     esi, esi
0x10c9d7:    mov     qword ptr [rip + 0x1ea2a], rbx
0x10c9de:    add     rsp, 8
0x10c9e2:    pop     rbx
0x10c9e3:    pop     rbp
0x10c9e4:    pop     r12
0x10c9e6:    pop     r13
0x10c9e8:    jmp     0x10c5e0
0x10c9ed:    nop     dword ptr [rax]
0x10c9f1:    nop     word ptr cs:[rax + rax]
0x10c9fc:    nop     dword ptr [rax]
0x10ca00:    endbr64
0x10ca04:    mov     rdx, qword ptr [rip + 0x1e9fd]
0x10ca0b:    mov     rax, qword ptr [rdx + 0x18]
0x10ca0f:    sub     rax, qword ptr [rdx + 0x40]
0x10ca13:    ret
0x10ca14:    nop     word ptr cs:[rax + rax]
0x10ca1f:    nop
0x10ca20:    endbr64
0x10ca24:    mov     rax, qword ptr [rip + 0x1e9dd]
0x10ca2b:    mov     r8d, dword ptr [rax + 8]
0x10ca2f:    mov     dword ptr [rax + 8], edi
0x10ca32:    mov     eax, r8d
0x10ca35:    ret
0x10ca36:    nop     word ptr cs:[rax + rax]
0x10ca40:    endbr64
0x10ca44:    mov     rax, qword ptr [rip + 0x1e9bd]
0x10ca4b:    cmp     qword ptr [rax + 0x48], 0

```

```

0x10ca50:    sete    al
0x10ca53:    movzx   eax, al
0x10ca56:    ret
0x10ca57:    nop     word ptr [rax + rax]
0x10ca60:    endbr64
0x10ca64:    mov     rax, qword ptr [rip + 0x1e99d]
0x10ca6b:    mov     rdx, qword ptr [rax + 0x18]
0x10ca6f:    mov     qword ptr [rax + 0x20], rdx
0x10ca73:    ret
0x10ca74:    nop     word ptr cs:[rax + rax]
0x10ca7f:    nop
0x10ca80:    endbr64
0x10ca84:    sub     rsp, 8
0x10ca88:    xor     eax, eax
0x10ca8a:    mov     edi, 3
0x10ca8f:    call    0x10c6e0
0x10ca94:    mov     rax, qword ptr [rip + 0x1e96d]
0x10ca9b:    mov     dword ptr [rax + 0x10], 0x18
0x10caa2:    add     rsp, 8
0x10caa6:    ret
0x10caa7:    nop     word ptr [rax + rax]
0x10cab0:    endbr64
0x10cab4:    sub     rsp, 8
0x10cab8:    xor     eax, eax
0x10caba:    mov     edi, 1
0x10cabf:    call    0x10c6e0
0x10cac4:    mov     rax, qword ptr [rip + 0x1e93d]
0x10cacb:    mov     dword ptr [rax + 0x10], 0x19
0x10cad2:    add     rsp, 8
0x10cad6:    ret
0x10cad7:    nop     word ptr [rax + rax]
0x10cae0:    endbr64
0x10cae4:    mov     rax, qword ptr [rip + 0x1e91d]
0x10caeb:    mov     rdx, qword ptr [rax + 0x18]
0x10caef:    mov     rcx, qword ptr [rax + 0x20]
0x10caf3:    movsxd  rdx, edx
0x10caf6:    mov     qword ptr [rax + 0x18], rcx
0x10cafa:    mov     qword ptr [rax + 0x20], rdx
0x10cafe:    ret
0x10caff:    nop
0x10cb00:    endbr64
0x10cb04:    mov     rax, qword ptr [rip + 0x1e8fd]
0x10cb0b:    mov     rdx, qword ptr [rax + 0x18]
0x10cb0f:    cmp     rdx, qword ptr [rax + 0x40]
0x10cb13:    jbe     0x10cb24
0x10cb24:    ret
0x10cb15:    sub     rdx, 1
0x10cb19:    mov     dword ptr [rax + 0x10], 1

```

```

0x10cb20:    mov     qword ptr [rax + 0x18], rdx
0x10cb24:    ret
0x10cb25:    nop     word ptr cs:[rax + rax]
0x10cb30:    endbr64
0x10cb34:    mov     rax, qword ptr [rip + 0x1e8cd]
0x10cb3b:    mov     rdx, qword ptr [rax + 0x18]
0x10cb3f:    cmp     rdx, qword ptr [rax + 0x38]
0x10cb43:    jae     0x10cb54
0x10cb45:    add     rdx, 1
0x10cb49:    mov     dword ptr [rax + 0x10], 2
0x10cb50:    mov     qword ptr [rax + 0x18], rdx
0x10cb54:    ret
0x10cb54:    ret
0x10cb55:    nop     word ptr cs:[rax + rax]
0x10cb60:    endbr64
0x10cb64:    mov     rdx, qword ptr [rip + 0x1e89d]
0x10cb6b:    mov     rax, qword ptr [rdx + 0x18]
0x10cb6f:    mov     rdi, qword ptr [rdx + 0x40]
0x10cb73:    cmp     rax, rdi
0x10cb76:    jbe     0x10cc20
0x10cc20:    ret
0x10cb7c:    mov     r8, qword ptr [rdx + 0x58]
0x10cb80:    jmp     0x10cba9
0x10cba9:    sub     rax, 1
0x10cbad:    movzx   esi, byte ptr [r8 + rax]
0x10cbb2:    mov     ecx, esi
0x10cbb4:    and     ecx, 0xffffffff
0x10cbb7:    sub     ecx, 0x41
0x10cbba:    cmp     cl, 0x19
0x10cbbd:    ja      0x10cb88
0x10cbbf:    mov     rax, qword ptr [rdx + 0x18]
0x10cbc3:    cmp     rdi, rax
0x10cbc6:    jae     0x10cbef
0x10cb88:    lea     ecx, [rsi - 0x30]
0x10cb8b:    cmp     cl, 9
0x10cb8e:    jbe     0x10cbbf
0x10cbc8:    sub     rax, 1
0x10cbcc:    movzx   esi, byte ptr [r8 + rax]
0x10cbd1:    mov     ecx, esi
0x10cbd3:    and     ecx, 0xffffffff
0x10cbd6:    sub     ecx, 0x41
0x10cbd9:    cmp     cl, 0x19
0x10cbdc:    jbe     0x10cc00
0x10cbef:    mov     dword ptr [rdx + 0x10], 3
0x10cbf6:    ret
0x10cb90:    sub     esi, 0x24
0x10cb93:    cmp     sil, 1
0x10cb97:    jbe     0x10cbbf

```



```

0x10cc00:    mov     qword ptr [rdx + 0x18], rax
0x10cc04:    mov     dword ptr [rdx + 0x10], 1
0x10cc0b:    cmp     rdi, rax
0x10cc0e:    jb      0x10cbc8
0x10cbde:    lea     ecx, [rsi - 0x30]
0x10cbe1:    cmp     cl, 9
0x10cbe4:    jbe     0x10cc00
0x10cb99:    mov     qword ptr [rdx + 0x18], rax
0x10cb9d:    mov     dword ptr [rdx + 0x10], 1
0x10cba4:    cmp     rdi, rax
0x10cba7:    jae     0x10cbef
0x10cc10:    mov     dword ptr [rdx + 0x10], 3
0x10cc17:    ret
0x10cbe6:    sub     esi, 0x24
0x10cbe9:    cmp     sil, 1
0x10cbef:    jbe     0x10cc00
0x10cb82:    nop     word ptr [rax + rax]
0x10cbf7:    nop     word ptr [rax + rax]
0x10cc18:    nop     dword ptr [rax + rax]
0x10cc21:    nop     word ptr cs:[rax + rax]
0x10cc2c:    nop     dword ptr [rax]
0x10cc30:    endbr64
0x10cc34:    mov     rdx, qword ptr [rip + 0x1e7cd]
0x10cc3b:    mov     rax, qword ptr [rdx + 0x18]
0x10cc3f:    mov     rdi, qword ptr [rdx + 0x38]
0x10cc43:    cmp     rax, rdi
0x10cc46:    jae     0x10cce8
0x10cc4c:    mov     r8, qword ptr [rdx + 0x58]
0x10cc50:    jmp     0x10cc7d
0x10cce8:    ret
0x10cc7d:    movzx   esi, byte ptr [r8 + rax]
0x10cc82:    mov     ecx, esi
0x10cc84:    and     ecx, 0xffffffff
0x10cc87:    sub     ecx, 0x41
0x10cc8a:    cmp     cl, 0x19
0x10cc8d:    ja      0x10cc58
0x10cc8f:    mov     rax, qword ptr [rdx + 0x18]
0x10cc93:    cmp     rdi, rax
0x10cc96:    jbe     0x10ccb9
0x10cc58:    lea     ecx, [rsi - 0x30]
0x10cc5b:    cmp     cl, 9
0x10cc5e:    jbe     0x10cc8f
0x10ccb9:    mov     dword ptr [rdx + 0x10], 4
0x10ccc2:    ret
0x10cc98:    movzx   esi, byte ptr [r8 + rax]
0x10cc9d:    mov     ecx, esi
0x10cc9f:    and     ecx, 0xffffffff
0x10cca2:    sub     ecx, 0x41

```

```

0x10cca5:    cmp     cl, 0x19
0x10cca8:    jbe     0x10ccc8
0x10cc60:    sub     esi, 0x24
0x10cc63:    cmp     sil, 1
0x10cc67:    jbe     0x10cc8f
0x10ccc8:    add     rax, 1
0x10cccc:    mov     dword ptr [rdx + 0x10], 2
0x10ccd3:    mov     qword ptr [rdx + 0x18], rax
0x10ccd7:    cmp     rdi, rax
0x10ccda:    jne     0x10cc98
0x10ccaa:    lea     ecx, [rsi - 0x30]
0x10ccad:    cmp     cl, 9
0x10ccb0:    jbe     0x10ccc8
0x10cc69:    add     rax, 1
0x10cc6d:    mov     dword ptr [rdx + 0x10], 2
0x10cc74:    mov     qword ptr [rdx + 0x18], rax
0x10cc78:    cmp     rdi, rax
0x10cc7b:    je      0x10ccbb
0x10ccdc:    mov     dword ptr [rdx + 0x10], 4
0x10cce3:    ret
0x10ccb2:    sub     esi, 0x24
0x10ccb5:    cmp     sil, 1
0x10ccb9:    jbe     0x10ccc8
0x10cc52:    nop     word ptr [rax + rax]
0x10ccc3:    nop     dword ptr [rax + rax]
0x10cce4:    nop     dword ptr [rax]
0x10cce9:    nop     dword ptr [rax]
0x10ccf0:    endbr64
0x10ccf4:    mov     rax, qword ptr [rip + 0x1e70d]
0x10ccfb:    mov     rdx, qword ptr [rax + 0x40]
0x10ccff:    mov     dword ptr [rax + 0x10], 5
0x10cd06:    mov     qword ptr [rax + 0x18], rdx
0x10cd0a:    ret
0x10cd0b:    nop     dword ptr [rax + rax]
0x10cd10:    endbr64
0x10cd14:    mov     rax, qword ptr [rip + 0x1e6ed]
0x10cd1b:    mov     rdx, qword ptr [rax + 0x38]
0x10cd1f:    mov     dword ptr [rax + 0x10], 6
0x10cd26:    mov     qword ptr [rax + 0x18], rdx
0x10cd2a:    ret
0x10cd2b:    nop     dword ptr [rax + rax]
0x10cd30:    endbr64
0x10cd34:    push    rbp
0x10cd35:    push    rbx
0x10cd36:    mov     ebx, edi
0x10cd38:    sub     rsp, 8
0x10cd3c:    call    0x104f00
0x10cd41:    movsxd  rdx, ebx

```

```

0x10cd44:    mov     rax, qword ptr [rax]
0x10cd47:    test   byte ptr [rax + rdx*2 + 1], 0x40
0x10cd4c:    je      0x10cdab
0x10cdab:    add     rsp, 8
0x10cdaf:    pop     rbx
0x10cdb0:    pop     rbp
0x10cdb1:    ret
0x10cd4e:    mov     rbp, qword ptr [rip + 0x1e6b3]
0x10cd55:    mov     rsi, qword ptr [rbp + 0x38]
0x10cd59:    lea     rdx, [rsi + 1]
0x10cd5d:    cmp     rdx, qword ptr [rbp + 0x50]
0x10cd61:    jae     0x10cdb8
0x10cd63:    mov     rax, qword ptr [rbp + 0x18]
0x10cd67:    mov     rcx, qword ptr [rbp + 0x58]
0x10cd6b:    sub     rdx, rax
0x10cd6e:    lea     rsi, [rcx + rax]
0x10cd72:    lea     rdi, [rcx + rax + 1]
0x10cd77:    call    0x104d40
0x10cdb8:    add     rsi, 0x21
0x10cdbc:    mov     rdi, qword ptr [rbp + 0x58]
0x10cdc0:    mov     qword ptr [rbp + 0x50], rsi
0x10cdc4:    call    0x11d690
0x10cd7c:    mov     rax, qword ptr [rip + 0x1e685]
0x10cd83:    mov     rdx, qword ptr [rax + 0x58]
0x10cd87:    mov     rax, qword ptr [rax + 0x18]
0x10cd8b:    mov     byte ptr [rdx + rax], bl
0x10cd8e:    mov     rax, qword ptr [rip + 0x1e673]
0x10cd95:    add     qword ptr [rax + 0x18], 1
0x10cd9a:    add     qword ptr [rax + 0x38], 1
0x10cd9f:    add     qword ptr [rax + 0x48], 1
0x10cda4:    mov     dword ptr [rax + 0x10], 7
0x10cdab:    add     rsp, 8
0x10cdaf:    pop     rbx
0x10cdb0:    pop     rbp
0x10cdb1:    ret
0x10cdc9:    mov     qword ptr [rbp + 0x58], rax
0x10cdcd:    mov     rbp, qword ptr [rip + 0x1e634]
0x10cdd4:    mov     rax, qword ptr [rbp + 0x38]
0x10cdd8:    lea     rdx, [rax + 1]
0x10cddc:    jmp     0x10cd63
0x10cdb2:    nop     word ptr [rax + rax]
0x10cdde:    nop
0x10cde0:    endbr64
0x10cde4:    mov     rax, qword ptr [rip + 0x1e61d]
0x10cdeb:    mov     rcx, qword ptr [rax + 0x18]
0x10cdef:    mov     rdx, qword ptr [rax + 0x38]
0x10cdf3:    cmp     rcx, rdx
0x10cdf6:    jb      0x10ce00

```

```

0x10ce00:    push    rbx
0x10ce01:    mov     rax, qword ptr [rax + 0x58]
0x10ce05:    add     rdx, 1
0x10ce09:    sub     rdx, rcx
0x10ce0c:    lea     rsi, [rax + rcx + 1]
0x10ce11:    lea     rdi, [rax + rcx]
0x10ce15:    call    0x104bb0
0x10cdf8:    ret
0x10ce1a:    mov     rbx, qword ptr [rip + 0x1e5e7]
0x10ce21:    mov     rsi, qword ptr [rbx + 0x38]
0x10ce25:    sub     qword ptr [rbx + 0x48], 1
0x10ce2a:    lea     rax, [rsi - 1]
0x10ce2e:    mov     qword ptr [rbx + 0x38], rax
0x10ce32:    test    al, 0xf
0x10ce34:    je      0x10ce40
0x10ce40:    mov     qword ptr [rbx + 0x50], rsi
0x10ce44:    mov     rdi, qword ptr [rbx + 0x58]
0x10ce48:    call    0x11d690
0x10ce36:    mov     dword ptr [rbx + 0x10], 8
0x10ce3d:    pop     rbx
0x10ce3e:    ret
0x10ce4d:    mov     qword ptr [rbx + 0x58], rax
0x10ce51:    mov     rbx, qword ptr [rip + 0x1e5b0]
0x10ce58:    mov     dword ptr [rbx + 0x10], 8
0x10ce5f:    pop     rbx
0x10ce60:    ret
0x10cdf9:    nop     dword ptr [rax]
0x10ce3f:    nop
0x10ce61:    nop     word ptr cs:[rax + rax]
0x10ce6c:    nop     dword ptr [rax]
0x10ce70:    endbr64
0x10ce74:    mov     rax, qword ptr [rip + 0x1e58d]
0x10ce7b:    mov     rcx, qword ptr [rax + 0x18]
0x10ce7f:    cmp     rcx, qword ptr [rax + 0x40]
0x10ce83:    ja      0x10ce90
0x10ce85:    ret
0x10ce90:    push    rbx
0x10ce91:    mov     rdx, qword ptr [rax + 0x38]
0x10ce95:    mov     rsi, qword ptr [rax + 0x58]
0x10ce99:    add     rdx, 1
0x10ce9d:    lea     rdi, [rsi + rcx - 1]
0x10cea2:    sub     rdx, rcx
0x10cea5:    add     rsi, rcx
0x10cea8:    call    0x104bb0
0x10cead:    mov     rbx, qword ptr [rip + 0x1e554]
0x10ceb4:    mov     rsi, qword ptr [rbx + 0x38]
0x10ceb8:    sub     qword ptr [rbx + 0x18], 1
0x10cebd:    sub     qword ptr [rbx + 0x48], 1

```

```

0x10cec2:    lea     rax, [rsi - 1]
0x10cec6:    mov     qword ptr [rbx + 0x38], rax
0x10ceca:    test    al, 0xf
0x10cecc:    je      0x10cee0
0x10cee0:    mov     qword ptr [rbx + 0x50], rsi
0x10cee4:    mov     rdi, qword ptr [rbx + 0x58]
0x10cee8:    call    0x11d690
0x10cece:    mov     dword ptr [rbx + 0x10], 9
0x10ced5:    pop     rbx
0x10ced6:    ret
0x10ceed:    mov     qword ptr [rbx + 0x58], rax
0x10cef1:    mov     rbx, qword ptr [rip + 0x1e510]
0x10cef8:    mov     dword ptr [rbx + 0x10], 9
0x10ceff:    pop     rbx
0x10cf00:    ret
0x10ce86:    nop     word ptr cs:[rax + rax]
0x10ced7:    nop     word ptr [rax + rax]
0x10cf01:    nop     word ptr cs:[rax + rax]
0x10cf0c:    nop     dword ptr [rax]
0x10cf10:    endbr64
0x10cf14:    push    rbp
0x10cf15:    push    rbx
0x10cf16:    sub     rsp, 8
0x10cf1a:    mov     r9, qword ptr [rip + 0x1e4e7]
0x10cf21:    mov     rax, qword ptr [r9 + 0x18]
0x10cf25:    mov     rbx, qword ptr [r9 + 0x20]
0x10cf29:    mov     qword ptr [r9 + 0x20], rax
0x10cf2d:    xor     eax, eax
0x10cf2f:    call    0x10cc30
0x10cf34:    xor     eax, eax
0x10cf36:    mov     edi, 3
0x10cf3b:    mov     rbp, qword ptr [r9 + 0x18]
0x10cf3f:    call    0x10c6e0
0x10cf44:    mov     rax, qword ptr [rip + 0x1e4bd]
0x10cf4b:    mov     rdx, qword ptr [rax + 0x18]
0x10cf4f:    cmp     rdx, rbx
0x10cf52:    jae     0x10cf65
0x10cf54:    lea     rcx, [rdx + rbx]
0x10cf58:    sub     rcx, rbp
0x10cf5b:    cmp     rbx, rbp
0x10cf5e:    cmovae  rdx, rcx
0x10cf62:    mov     rbx, rdx
0x10cf65:    mov     qword ptr [rax + 0x20], rbx
0x10cf69:    mov     dword ptr [rax + 0x10], 0xa
0x10cf70:    add     rsp, 8
0x10cf74:    pop     rbx
0x10cf75:    pop     rbp
0x10cf76:    ret

```

```

0x10cf65:    mov     qword ptr [rax + 0x20], rbx
0x10cf69:    mov     dword ptr [rax + 0x10], 0xa
0x10cf70:    add     rsp, 8
0x10cf74:    pop     rbx
0x10cf75:    pop     rbp
0x10cf76:    ret
0x10cf77:    nop     word ptr [rax + rax]
0x10cf80:    endbr64
0x10cf84:    push    rbp
0x10cf85:    push    rbx
0x10cf86:    sub     rsp, 8
0x10cf8a:    mov     rax, qword ptr [rip + 0x1e477]
0x10cf91:    mov     rbp, qword ptr [rax + 0x18]
0x10cf95:    mov     rbx, qword ptr [rax + 0x20]
0x10cf99:    mov     qword ptr [rax + 0x20], rbp
0x10cf9d:    xor     eax, eax
0x10cf9f:    call    0x10cb60
0x10cfa4:    xor     eax, eax
0x10cfa6:    mov     edi, 3
0x10cfab:    call    0x10c6e0
0x10cfb0:    mov     rax, qword ptr [rip + 0x1e451]
0x10cfb7:    mov     rdx, qword ptr [rax + 0x18]
0x10cfbb:    cmp     rdx, rbx
0x10cfbe:    jae     0x10cfd1
0x10cfc0:    lea     rcx, [rdx + rbx]
0x10cfc4:    sub     rcx, rbp
0x10cfc7:    cmp     rbx, rbp
0x10cfca:    cmovae  rdx, rcx
0x10cfce:    mov     rbx, rdx
0x10cfd1:    mov     qword ptr [rax + 0x20], rbx
0x10cfd5:    mov     dword ptr [rax + 0x10], 0xb
0x10cfdc:    add     rsp, 8
0x10cfe0:    pop     rbx
0x10cfe1:    pop     rbp
0x10cfe2:    ret
0x10cfd1:    mov     qword ptr [rax + 0x20], rbx
0x10cfd5:    mov     dword ptr [rax + 0x10], 0xb
0x10cfdc:    add     rsp, 8
0x10cfe0:    pop     rbx
0x10cfe1:    pop     rbp
0x10cfe2:    ret
0x10cfe3:    nop     word ptr cs:[rax + rax]
0x10cfef:    nop
0x10cff0:    endbr64
0x10cff4:    push    rbx
0x10cff5:    mov     rbx, qword ptr [rip + 0x1e40c]
0x10cffc:    mov     esi, 1
0x10d001:    mov     qword ptr [rbx + 0x18], 0

```

```

0x10d009:      mov     rdi, qword ptr [rbx + 0x58]
0x10d00d:      mov     qword ptr [rbx + 0x20], 0
0x10d015:      mov     qword ptr [rbx + 0x38], 0
0x10d01d:      mov     qword ptr [rbx + 0x40], 0
0x10d025:      mov     qword ptr [rbx + 0x48], 0
0x10d02d:      mov     qword ptr [rbx + 0x50], 1
0x10d035:      call    0x11d690
0x10d03a:      mov     qword ptr [rbx + 0x58], rax
0x10d03e:      mov     rax, qword ptr [rip + 0x1e3c3]
0x10d045:      mov     rax, qword ptr [rax + 0x58]
0x10d049:      mov     byte ptr [rax], 0
0x10d04c:      mov     rax, qword ptr [rip + 0x1e3b5]
0x10d053:      mov     dword ptr [rax + 0x10], 0xc
0x10d05a:      pop     rbx
0x10d05b:      ret
0x10d05c:      nop     dword ptr [rax]
0x10d060:      endbr64
0x10d064:      push    rbx
0x10d065:      mov     edi, 0x70
0x10d06a:      sub     rsp, 0x10
0x10d06e:      mov     rax, qword ptr fs:[0x28]
0x10d077:      mov     qword ptr [rsp + 8], rax
0x10d07c:      xor     eax, eax
0x10d07e:      call    0x11d670
0x10d083:      mov     qword ptr [rax + 8], 1
0x10d08b:      mov     qword ptr [rax + 0x58], 0
0x10d093:      mov     qword ptr [rax + 0x60], 0
0x10d09b:      mov     qword ptr [rax + 0x28], 0
0x10d0a3:      mov     qword ptr [rax + 0x30], 0
0x10d0ab:      mov     qword ptr [rip + 0x1e356], rax
0x10d0b2:      xor     eax, eax
0x10d0b4:      call    0x10cff0
0x10d0b9:      mov     rbx, qword ptr [rip + 0x1e348]
0x10d0c0:      xor     eax, eax
0x10d0c2:      call    0x10c590
0x10d0c7:      lea     rdi, [rip + 0x135e4]
0x10d0ce:      mov     qword ptr [rbx], rax
0x10d0d1:      call    0x117c50
0x10d0d6:      mov     ecx, 1
0x10d0db:      mov     rsi, rsp
0x10d0de:      mov     edx, 1
0x10d0e3:      lea     rdi, [rip + 0x13e42]
0x10d0ea:      call    0x117320
0x10d0ef:      mov     rdi, qword ptr [rsp]
0x10d0f3:      lea     rax, [rip + 0x13e24]
0x10d0fa:      mov     rbx, qword ptr [rip + 0x1e307]
0x10d101:      test    rdi, rdi
0x10d104:      cmovne  rdi, rax

```

```

0x10d108:    xor     eax, eax
0x10d10a:    call    0x119940
0x10d10f:    lea     rdi, [rip + 0x1cb32]
0x10d116:    mov     qword ptr [rbx + 0x68], rax
0x10d11a:    mov     eax, dword ptr [rip + 0x1cb40]
0x10d120:    test    eax, eax
0x10d122:    lea     rax, [rip + 0x1cb07]
0x10d129:    cmov     rdi, rax
0x10d12d:    call    0x117c50
0x10d132:    lea     rdi, [rip + 0x1d0c7]
0x10d139:    mov     edx, 6
0x10d13e:    lea     rsi, [rip + 0x1d0db]
0x10d145:    call    0x117f70
0x10d14a:    xor     eax, eax
0x10d14c:    call    0x11a310
0x10d151:    mov     rax, qword ptr [rip + 0x1e2b0]
0x10d158:    mov     rdi, qword ptr [rax + 0x68]
0x10d15c:    xor     eax, eax
0x10d15e:    call    0x11a900
0x10d163:    nop     dword ptr [rax + rax]
0x10d168:    xor     eax, eax
0x10d16a:    call    0x11ab40
0x10d16f:    test    rax, rax
0x10d172:    jne     0x10d168
0x10d174:    mov     edi, 0x200
0x10d179:    call    0x11a620
0x10d17e:    mov     rax, qword ptr [rsp + 8]
0x10d183:    sub     rax, qword ptr fs:[0x28]
0x10d18c:    jne     0x10d194
0x10d18e:    add     rsp, 0x10
0x10d192:    pop     rbx
0x10d193:    ret
0x10d194:    call    0x1049d0
0x10d199:    nop     dword ptr [rax]
0x10d1a0:    endbr64
0x10d1a4:    sub     rsp, 8
0x10d1a8:    mov     rax, qword ptr [rip + 0x1e259]
0x10d1af:    or      edi, 2
0x10d1b2:    mov     rdx, qword ptr [rax + 0x40]
0x10d1b6:    mov     dword ptr [rax + 0x10], 6
0x10d1bd:    mov     qword ptr [rax + 0x20], rdx
0x10d1c1:    mov     rdx, qword ptr [rax + 0x38]
0x10d1c5:    mov     qword ptr [rax + 0x18], rdx
0x10d1c9:    xor     eax, eax
0x10d1cb:    call    0x10c6e0
0x10d1d0:    mov     rax, qword ptr [rip + 0x1e231]
0x10d1d7:    mov     rdx, qword ptr [rax + 0x18]
0x10d1db:    mov     dword ptr [rax + 0x10], 0xd

```



```

0x10d1e2:    mov     qword ptr [rax + 0x20], rdx
0x10d1e6:    add     rsp, 8
0x10d1ea:    ret
0x10d1eb:    nop     dword ptr [rax + rax]
0x10d1f0:    endbr64
0x10d1f4:    mov     rax, qword ptr [rip + 0x1e20d]
0x10d1fb:    push    rbx
0x10d1fc:    mov     edi, 3
0x10d201:    mov     rdx, qword ptr [rax + 0x18]
0x10d205:    mov     rsi, qword ptr [rax + 0x20]
0x10d209:    mov     dword ptr [rax + 0x10], 5
0x10d210:    mov     rcx, qword ptr [rax + 0x40]
0x10d214:    mov     qword ptr [rax + 0x20], rdx
0x10d218:    mov     rbx, rsi
0x10d21b:    mov     qword ptr [rax + 0x18], rcx
0x10d21f:    sub     rbx, rdx
0x10d222:    cmp     rsi, rdx
0x10d225:    cmovbe  rbx, rcx
0x10d229:    xor     eax, eax
0x10d22b:    call    0x10c6e0
0x10d230:    mov     rdx, qword ptr [rip + 0x1e1d1]
0x10d237:    mov     rax, qword ptr [rdx + 0x38]
0x10d23b:    mov     dword ptr [rdx + 0x10], 0xe
0x10d242:    cmp     rax, rbx
0x10d245:    cmova   rax, rbx
0x10d249:    mov     qword ptr [rdx + 0x20], rax
0x10d24d:    pop     rbx
0x10d24e:    ret
0x10d24f:    nop
0x10d250:    endbr64
0x10d254:    mov     rax, qword ptr [rip + 0x1e1ad]
0x10d25b:    push    rbx
0x10d25c:    mov     edi, 3
0x10d261:    mov     rdx, qword ptr [rax + 0x18]
0x10d265:    mov     rbx, qword ptr [rax + 0x20]
0x10d269:    mov     dword ptr [rax + 0x10], 6
0x10d270:    mov     qword ptr [rax + 0x20], rdx
0x10d274:    mov     rdx, qword ptr [rax + 0x38]
0x10d278:    mov     qword ptr [rax + 0x18], rdx
0x10d27c:    xor     eax, eax
0x10d27e:    call    0x10c6e0
0x10d283:    mov     rdx, qword ptr [rip + 0x1e17e]
0x10d28a:    mov     rax, qword ptr [rdx + 0x38]
0x10d28e:    mov     dword ptr [rdx + 0x10], 0xf
0x10d295:    cmp     rax, rbx
0x10d298:    cmova   rax, rbx
0x10d29c:    mov     qword ptr [rdx + 0x20], rax
0x10d2a0:    pop     rbx

```

```

0x10d2a1:    ret
0x10d2a2:    nop        word ptr cs:[rax + rax]
0x10d2ad:    nop        dword ptr [rax]
0x10d2b0:    endbr64
0x10d2b4:    mov        rax, qword ptr [rip + 0x1e14d]
0x10d2bb:    mov        rdx, qword ptr [rax + 0x58]
0x10d2bf:    mov        rax, qword ptr [rax + 0x18]
0x10d2c3:    cmp        byte ptr [rdx + rax], 0x20
0x10d2c7:    je         0x10d2d0
0x10d2d0:    sub        rsp, 8
0x10d2d4:    nop        dword ptr [rax]
0x10d2d8:    xor        eax, eax
0x10d2da:    call       0x10cde0
0x10d2c9:    ret
0x10d2df:    mov        rax, qword ptr [rip + 0x1e122]
0x10d2e6:    mov        rdx, qword ptr [rax + 0x58]
0x10d2ea:    mov        rcx, qword ptr [rax + 0x18]
0x10d2ee:    cmp        byte ptr [rdx + rcx], 0x20
0x10d2f2:    je         0x10d2d8
0x10d2d8:    xor        eax, eax
0x10d2da:    call       0x10cde0
0x10d2f4:    cmp        qword ptr [rax + 0x48], 0
0x10d2f9:    jne        0x10d31d
0x10d2fb:    jmp        0x10d324
0x10d31d:    cmp        byte ptr [rdx + rcx - 1], 0x20
0x10d322:    je         0x10d300
0x10d324:    mov        dword ptr [rax + 0x10], 0x11
0x10d32b:    add        rsp, 8
0x10d32f:    ret
0x10d300:    xor        eax, eax
0x10d302:    call       0x10ce70
0x10d307:    mov        rax, qword ptr [rip + 0x1e0fa]
0x10d30e:    cmp        qword ptr [rax + 0x48], 0
0x10d313:    je         0x10d324
0x10d315:    mov        rdx, qword ptr [rax + 0x58]
0x10d319:    mov        rcx, qword ptr [rax + 0x18]
0x10d31d:    cmp        byte ptr [rdx + rcx - 1], 0x20
0x10d322:    je         0x10d300
0x10d2ca:    nop        word ptr [rax + rax]
0x10d2fd:    nop        dword ptr [rax]
0x10d330:    endbr64
0x10d334:    mov        rax, qword ptr [rip + 0x1e0cd]
0x10d33b:    mov        rdx, qword ptr [rax + 0x58]
0x10d33f:    mov        rax, qword ptr [rax + 0x18]
0x10d343:    cmp        byte ptr [rdx + rax], 0x20
0x10d347:    je         0x10d350
0x10d350:    sub        rsp, 8
0x10d354:    xor        eax, eax

```

```

0x10d356:    call    0x10d2b0
0x10d349:    ret
0x10d35b:    mov     edi, 0x20
0x10d360:    call    0x10cd30
0x10d365:    mov     rax, qword ptr [rip + 0x1e09c]
0x10d36c:    mov     dword ptr [rax + 0x10], 0x10
0x10d373:    add     rsp, 8
0x10d377:    ret
0x10d34a:    nop     word ptr [rax + rax]
0x10d378:    nop     dword ptr [rax + rax]
0x10d380:    endbr64
0x10d384:    push    rbp
0x10d385:    push    rbx
0x10d386:    sub     rsp, 8
0x10d38a:    mov     rbp, qword ptr [rip + 0x1e077]
0x10d391:    mov     rbx, qword ptr [rbp + 0x18]
0x10d395:    cmp     rbx, qword ptr [rbp + 0x38]
0x10d399:    jb      0x10d3a8
0x10d3a8:    xor     eax, eax
0x10d3aa:    call    0x10cc30
0x10d39b:    add     rsp, 8
0x10d39f:    pop     rbx
0x10d3a0:    pop     rbp
0x10d3a1:    ret
0x10d3af:    cmp     rbx, qword ptr [rbp + 0x18]
0x10d3b3:    jae     0x10d3e4
0x10d3b5:    call    0x104ef0
0x10d3e4:    mov     dword ptr [rbp + 0x10], 0x12
0x10d3eb:    add     rsp, 8
0x10d3ef:    pop     rbx
0x10d3f0:    pop     rbp
0x10d3f1:    ret
0x10d3ba:    nop     word ptr [rax + rax]
0x10d3c0:    mov     rdx, qword ptr [rbp + 0x58]
0x10d3c4:    mov     rcx, qword ptr [rax]
0x10d3c7:    add     rdx, rbx
0x10d3ca:    add     rbx, 1
0x10d3ce:    movsx   rsi, byte ptr [rdx]
0x10d3d2:    mov     ecx, dword ptr [rcx + rsi*4]
0x10d3d5:    mov     byte ptr [rdx], cl
0x10d3d7:    mov     rbp, qword ptr [rip + 0x1e02a]
0x10d3de:    cmp     qword ptr [rbp + 0x18], rbx
0x10d3e2:    ja      0x10d3c0
0x10d3a2:    nop     word ptr [rax + rax]
0x10d3f2:    nop     word ptr cs:[rax + rax]
0x10d3fd:    nop     dword ptr [rax]
0x10d400:    endbr64
0x10d404:    push    rbp

```

```

0x10d405:    push    rbx
0x10d406:    sub     rsp, 8
0x10d40a:    mov     rbp, qword ptr [rip + 0x1dff7]
0x10d411:    mov     rbx, qword ptr [rbp + 0x18]
0x10d415:    cmp     rbx, qword ptr [rbp + 0x38]
0x10d419:    jb      0x10d428
0x10d428:    xor     eax, eax
0x10d42a:    call    0x10cc30
0x10d41b:    add     rsp, 8
0x10d41f:    pop     rbx
0x10d420:    pop     rbp
0x10d421:    ret
0x10d42f:    cmp     rbx, qword ptr [rbp + 0x18]
0x10d433:    jae     0x10d464
0x10d435:    call    0x1047c0
0x10d464:    mov     dword ptr [rbp + 0x10], 0x13
0x10d46b:    add     rsp, 8
0x10d46f:    pop     rbx
0x10d470:    pop     rbp
0x10d471:    ret
0x10d43a:    nop     word ptr [rax + rax]
0x10d440:    mov     rdx, qword ptr [rbp + 0x58]
0x10d444:    mov     rcx, qword ptr [rax]
0x10d447:    add     rdx, rbx
0x10d44a:    add     rbx, 1
0x10d44e:    movsx   rsi, byte ptr [rdx]
0x10d452:    mov     ecx, dword ptr [rcx + rsi*4]
0x10d455:    mov     byte ptr [rdx], cl
0x10d457:    mov     rbp, qword ptr [rip + 0x1dfaa]
0x10d45e:    cmp     qword ptr [rbp + 0x18], rbx
0x10d462:    ja      0x10d440
0x10d422:    nop     word ptr [rax + rax]
0x10d472:    nop     word ptr cs:[rax + rax]
0x10d47d:    nop     dword ptr [rax]
0x10d480:    endbr64
0x10d484:    push    r14
0x10d486:    push    r13
0x10d488:    push    r12
0x10d48a:    push    rbp
0x10d48b:    mov     rbp, qword ptr [rip + 0x1df76]
0x10d492:    push    rbx
0x10d493:    mov     rbx, qword ptr [rbp + 0x18]
0x10d497:    cmp     rbx, qword ptr [rbp + 0x38]
0x10d49b:    jb      0x10d4b0
0x10d4b0:    xor     eax, eax
0x10d4b2:    call    0x10cc30
0x10d49d:    pop     rbx
0x10d49e:    pop     rbp

```

```

0x10d49f:    pop     r12
0x10d4a1:    pop     r13
0x10d4a3:    pop     r14
0x10d4a5:    ret
0x10d4b7:    cmp     rbx, qword ptr [rbp + 0x18]
0x10d4bb:    jae     0x10d528
0x10d4bd:    call    0x104f00
0x10d528:    mov     dword ptr [rbp + 0x10], 0x14
0x10d52f:    pop     rbx
0x10d530:    pop     rbp
0x10d531:    pop     r12
0x10d533:    pop     r13
0x10d535:    pop     r14
0x10d537:    ret
0x10d4c2:    mov     r14d, 1
0x10d4c8:    mov     r13, rax
0x10d4cb:    jmp     0x10d4f3
0x10d4f3:    mov     rax, qword ptr [rbp + 0x58]
0x10d4f7:    add     rax, rbx
0x10d4fa:    movsx   r12, byte ptr [rax]
0x10d4fe:    mov     rbp, rax
0x10d501:    mov     rax, qword ptr [r13]
0x10d505:    test    byte ptr [rax + r12*2], 8
0x10d50a:    je      0x10d4e2
0x10d4e2:    mov     rbp, qword ptr [rip + 0x1df1f]
0x10d4e9:    add     rbx, 1
0x10d4ed:    cmp     qword ptr [rbp + 0x18], rbx
0x10d4f1:    jbe     0x10d528
0x10d50c:    test    r14d, r14d
0x10d50f:    jne     0x10d4d0
0x10d511:    call    0x104ef0
0x10d4d0:    call    0x1047c0
0x10d516:    mov     rax, qword ptr [rax]
0x10d519:    mov     eax, dword ptr [rax + r12*4]
0x10d51d:    mov     byte ptr [rbp], al
0x10d520:    jmp     0x10d4e2
0x10d4d5:    xor     r14d, r14d
0x10d4d8:    mov     rax, qword ptr [rax]
0x10d4db:    mov     eax, dword ptr [rax + r12*4]
0x10d4df:    mov     byte ptr [rbp], al
0x10d4e2:    mov     rbp, qword ptr [rip + 0x1df1f]
0x10d4e9:    add     rbx, 1
0x10d4ed:    cmp     qword ptr [rbp + 0x18], rbx
0x10d4f1:    jbe     0x10d528
0x10d4a6:    nop     word ptr cs:[rax + rax]
0x10d4cd:    nop     dword ptr [rax]
0x10d522:    nop     word ptr [rax + rax]
0x10d538:    nop     dword ptr [rax + rax]

```

```

0x10d540:      endbr64
0x10d544:      push    r12
0x10d546:      mov     r12, rdi
0x10d549:      push    rbp
0x10d54a:      push    rbx
0x10d54b:      call    0x1049b0
0x10d550:      mov     rbp, qword ptr [rip + 0x1deb1]
0x10d557:      mov     rbx, rax
0x10d55a:      mov     rsi, qword ptr [rbp + 0x40]
0x10d55e:      sub     rax, rsi
0x10d561:      add     qword ptr [rbp + 0x18], rax
0x10d565:      add     qword ptr [rbp + 0x20], rax
0x10d569:      mov     rax, qword ptr [rbp + 0x48]
0x10d56d:      lea     rdx, [rax + 1]
0x10d571:      lea     r8, [rbx + rdx]
0x10d575:      cmp     r8, qword ptr [rbp + 0x50]
0x10d579:      ja      0x10d5d8
0x10d57b:      mov     rdi, qword ptr [rbp + 0x58]
0x10d57f:      add     rsi, rdi
0x10d582:      add     rdi, rbx
0x10d585:      call    0x104d40
0x10d5d8:      mov     qword ptr [rbp + 0x50], r8
0x10d5dc:      mov     rdi, qword ptr [rbp + 0x58]
0x10d5e0:      mov     rsi, r8
0x10d5e3:      call    0x11d690
0x10d58a:      mov     rax, qword ptr [rip + 0x1de77]
0x10d591:      mov     rdx, rbx
0x10d594:      mov     rsi, r12
0x10d597:      mov     rdi, qword ptr [rax + 0x58]
0x10d59b:      call    0x104bb0
0x10d5e8:      mov     qword ptr [rbp + 0x58], rax
0x10d5ec:      mov     rbp, qword ptr [rip + 0x1de15]
0x10d5f3:      mov     rax, qword ptr [rbp + 0x48]
0x10d5f7:      mov     rsi, qword ptr [rbp + 0x40]
0x10d5fb:      lea     rdx, [rax + 1]
0x10d5ff:      jmp     0x10d57b
0x10d5a0:      mov     rax, qword ptr [rip + 0x1de61]
0x10d5a7:      mov     rsi, rbx
0x10d5aa:      mov     rdi, qword ptr [rax + 0x58]
0x10d5ae:      call    0x1184d0
0x10d5b3:      mov     rax, qword ptr [rip + 0x1de4e]
0x10d5ba:      mov     qword ptr [rax + 0x40], rbx
0x10d5be:      add     rbx, qword ptr [rax + 0x48]
0x10d5c2:      mov     qword ptr [rax + 0x38], rbx
0x10d5c6:      mov     dword ptr [rax + 0x10], 0x15
0x10d5cd:      pop     rbx
0x10d5ce:      pop     rbp
0x10d5cf:      pop     r12

```

```

0x10d5d1:    ret
0x10d5d2:    nop     word ptr [rax + rax]
0x10d604:    nop     word ptr cs:[rax + rax]
0x10d60f:    nop
0x10d610:    endbr64
0x10d614:    test    rdi, rdi
0x10d617:    je      0x10d6f0
0x10d6f0:    ret
0x10d61d:    push    r12
0x10d61f:    push    rbp
0x10d620:    mov     rbp, rdi
0x10d623:    push    rbx
0x10d624:    call    0x1049b0
0x10d629:    mov     r12, qword ptr [rip + 0x1ddd8]
0x10d630:    mov     rbx, rax
0x10d633:    mov     rax, qword ptr [r12 + 0x38]
0x10d638:    lea     rdx, [rax + 1]
0x10d63c:    mov     rax, qword ptr [r12 + 0x50]
0x10d641:    lea     rcx, [rbx + rdx]
0x10d645:    cmp     rcx, rax
0x10d648:    ja      0x10d6c0
0x10d64a:    mov     rax, qword ptr [r12 + 0x18]
0x10d64f:    mov     rdi, qword ptr [r12 + 0x58]
0x10d654:    sub     rdx, rax
0x10d657:    lea     rsi, [rdi + rax]
0x10d65b:    add     rax, rbx
0x10d65e:    add     rdi, rax
0x10d661:    call    0x104d40
0x10d6c0:    lea     rsi, [rax + rbx + 0x21]
0x10d6c5:    mov     rdi, qword ptr [r12 + 0x58]
0x10d6ca:    mov     qword ptr [r12 + 0x50], rsi
0x10d6cf:    call    0x11d690
0x10d666:    mov     rax, qword ptr [rip + 0x1dd9b]
0x10d66d:    mov     rdx, rbx
0x10d670:    mov     rsi, rbp
0x10d673:    mov     rdi, qword ptr [rax + 0x18]
0x10d677:    add     rdi, qword ptr [rax + 0x58]
0x10d67b:    call    0x104bb0
0x10d6d4:    mov     qword ptr [r12 + 0x58], rax
0x10d6d9:    mov     r12, qword ptr [rip + 0x1dd28]
0x10d6e0:    mov     rax, qword ptr [r12 + 0x38]
0x10d6e5:    lea     rdx, [rax + 1]
0x10d6e9:    jmp     0x10d64a
0x10d680:    mov     rax, qword ptr [rip + 0x1dd81]
0x10d687:    mov     rsi, rbx
0x10d68a:    mov     rdi, qword ptr [rax + 0x18]
0x10d68e:    add     rdi, qword ptr [rax + 0x58]
0x10d692:    call    0x1184d0

```

```

0x10d697:    mov     rax, qword ptr [rip + 0x1dd6a]
0x10d69e:    add     qword ptr [rax + 0x18], rbx
0x10d6a2:    add     qword ptr [rax + 0x38], rbx
0x10d6a6:    add     qword ptr [rax + 0x48], rbx
0x10d6aa:    mov     dword ptr [rax + 0x10], 0x16
0x10d6b1:    pop     rbx
0x10d6b2:    pop     rbp
0x10d6b3:    pop     r12
0x10d6b5:    ret
0x10d6b6:    nop     word ptr cs:[rax + rax]
0x10d6ee:    nop
0x10d6f1:    nop     word ptr cs:[rax + rax]
0x10d6fc:    nop     dword ptr [rax]
0x10d700:    endbr64
0x10d704:    mov     rax, qword ptr [rip + 0x1dcfd]
0x10d70b:    mov     rdi, qword ptr [rax + 0x60]
0x10d70f:    test    rdi, rdi
0x10d712:    je      0x10d730
0x10d730:    ret
0x10d714:    sub     rsp, 8
0x10d718:    call    0x10d610
0x10d71d:    mov     rax, qword ptr [rip + 0x1dce4]
0x10d724:    mov     dword ptr [rax + 0x10], 0x1a
0x10d72b:    add     rsp, 8
0x10d72f:    ret
0x10d731:    nop     word ptr cs:[rax + rax]
0x10d73c:    nop     dword ptr [rax]
0x10d740:    endbr64
0x10d744:    mov     rax, qword ptr [rip + 0x1dcbd]
0x10d74b:    push    r12
0x10d74d:    push    rbp
0x10d74e:    push    rbx
0x10d74f:    mov     rbx, qword ptr [rax + 0x40]
0x10d753:    mov     r12, qword ptr [rax + 0x28]
0x10d757:    mov     eax, dword ptr [rax + 0xc]
0x10d75a:    lea     rbp, [rbx + 3]
0x10d75e:    test    eax, eax
0x10d760:    je      0x10d7f0
0x10d7f0:    mov     edx, dword ptr [rip + 0x1ca0e]
0x10d7f6:    mov     esi, dword ptr [rip + 0x1ca04]
0x10d7fc:    mov     edi, dword ptr [rip + 0x1ca06]
0x10d802:    jmp     0x10d778
0x10d766:    mov     edx, dword ptr [rip + 0x1caa4]
0x10d76c:    mov     esi, dword ptr [rip + 0x1ca9a]
0x10d772:    mov     edi, dword ptr [rip + 0x1ca9c]
0x10d778:    call    0x10a570
0x10d778:    call    0x10a570
0x10d77d:    mov     rcx, qword ptr [rip + 0x1dc84]

```



```

0x10d784:    cmp     r12, rbp
0x10d787:    jb      0x10d810
0x10d810:    mov     rbx, qword ptr [rcx + 0x40]
0x10d814:    mov     rsi, qword ptr [rcx + 0x18]
0x10d818:    mov     qword ptr [rcx + 0x40], 0
0x10d820:    xor     edx, edx
0x10d822:    add     qword ptr [rcx + 0x58], rbx
0x10d826:    sub     rsi, rbx
0x10d829:    sub     qword ptr [rcx + 0x38], rbx
0x10d82d:    sub     qword ptr [rcx + 0x20], rbx
0x10d831:    movsxd  rbx, ebx
0x10d834:    mov     qword ptr [rcx + 0x18], rsi
0x10d838:    jmp     0x10d797
0x10d78d:    mov     rsi, qword ptr [rcx + 0x18]
0x10d791:    mov     rdx, qword ptr [rcx + 0x40]
0x10d795:    xor     ebx, ebx
0x10d797:    mov     rdi, qword ptr [rcx + 0x28]
0x10d79b:    mov     rax, rdi
0x10d79e:    sub     rax, rdx
0x10d7a1:    mov     edx, 1
0x10d7a6:    mov     r8, rax
0x10d7a9:    shr     r8, 2
0x10d7ad:    cmp     rax, 3
0x10d7b1:    cmova   edx, r8d
0x10d7b5:    xor     eax, eax
0x10d7b7:    cmp     rdi, rsi
0x10d7ba:    ja      0x10d7d3
0x10d797:    mov     rdi, qword ptr [rcx + 0x28]
0x10d79b:    mov     rax, rdi
0x10d79e:    sub     rax, rdx
0x10d7a1:    mov     edx, 1
0x10d7a6:    mov     r8, rax
0x10d7a9:    shr     r8, 2
0x10d7ad:    cmp     rax, 3
0x10d7b1:    cmova   edx, r8d
0x10d7b5:    xor     eax, eax
0x10d7b7:    cmp     rdi, rsi
0x10d7ba:    ja      0x10d7d3
0x10d7bc:    mov     rax, rsi
0x10d7bf:    sub     rax, rdi
0x10d7c2:    movsxd  rdi, edx
0x10d7c5:    lea     r8d, [rdx + rax]
0x10d7c9:    xor     edx, edx
0x10d7cb:    div     rdi
0x10d7ce:    mov     eax, r8d
0x10d7d1:    sub     eax, edx
0x10d7d3:    mov     edx, esi
0x10d7d5:    mov     rdi, qword ptr [rcx]

```

```

0x10d7d8:    xor     esi, esi
0x10d7da:    sub     edx, eax
0x10d7dc:    call    0x10c680
0x10d7d3:    mov     edx, esi
0x10d7d5:    mov     rdi, qword ptr [rcx]
0x10d7d8:    xor     esi, esi
0x10d7da:    sub     edx, eax
0x10d7dc:    call    0x10c680
0x10d7e1:    cmp     r12, rbp
0x10d7e4:    jb      0x10d840
0x10d840:    mov     rax, qword ptr [rip + 0x1dbc1]
0x10d847:    movsxd  rbx, ebx
0x10d84a:    mov     qword ptr [rax + 0x40], rbx
0x10d84e:    sub     qword ptr [rax + 0x58], rbx
0x10d852:    add     qword ptr [rax + 0x38], rbx
0x10d856:    add     qword ptr [rax + 0x18], rbx
0x10d85a:    add     qword ptr [rax + 0x20], rbx
0x10d85e:    pop     rbx
0x10d85f:    pop     rbp
0x10d860:    pop     r12
0x10d862:    ret
0x10d7e6:    pop     rbx
0x10d7e7:    pop     rbp
0x10d7e8:    pop     r12
0x10d7ea:    ret
0x10d7eb:    nop     dword ptr [rax + rax]
0x10d807:    nop     word ptr [rax + rax]
0x10d83d:    nop     dword ptr [rax]
0x10d863:    nop     word ptr cs:[rax + rax]
0x10d86e:    nop
0x10d870:    endbr64
0x10d874:    push    r15
0x10d876:    push    r14
0x10d878:    push    r13
0x10d87a:    push    r12
0x10d87c:    push    rbp
0x10d87d:    push    rbx
0x10d87e:    sub     rsp, 0x28
0x10d882:    mov     rax, qword ptr fs:[0x28]
0x10d88b:    mov     qword ptr [rsp + 0x18], rax
0x10d890:    mov     rax, qword ptr [rip + 0x1db71]
0x10d897:    lea     r14, [rsp + 0x17]
0x10d89c:    mov     rdi, r14
0x10d89f:    mov     rbx, qword ptr [rax + 0x40]
0x10d8a3:    mov     r13, qword ptr [rax + 0x28]
0x10d8a7:    call    0x10a6a0
0x10d8ac:    mov     rax, qword ptr [rip + 0x1db55]
0x10d8b3:    add     rbx, 3

```

```

0x10d8b7:    mov     ecx, dword ptr [rax + 0xc]
0x10d8ba:    test    ecx, ecx
0x10d8bc:    je      0x10da20
0x10da20:    mov     edx, dword ptr [rip + 0x1c7de]
0x10da26:    mov     esi, dword ptr [rip + 0x1c7d4]
0x10da2c:    mov     edi, dword ptr [rip + 0x1c7d6]
0x10da32:    jmp     0x10d8d4
0x10d8c2:    mov     edx, dword ptr [rip + 0x1c948]
0x10d8c8:    mov     esi, dword ptr [rip + 0x1c93e]
0x10d8ce:    mov     edi, dword ptr [rip + 0x1c940]
0x10d8d4:    call    0x10a570
0x10d8d4:    call    0x10a570
0x10d8d9:    mov     rdx, qword ptr [rip + 0x1db28]
0x10d8e0:    cmp     r13, rbx
0x10d8e3:    jb      0x10da90
0x10da90:    mov     r12, qword ptr [rdx + 0x40]
0x10da94:    mov     rax, qword ptr [rdx + 0x18]
0x10da98:    mov     qword ptr [rdx + 0x40], 0
0x10daa0:    add     qword ptr [rdx + 0x58], r12
0x10daa4:    movsxd  rcx, r12d
0x10daa7:    sub     rax, r12
0x10daaa:    sub     qword ptr [rdx + 0x38], r12
0x10daae:    mov     qword ptr [rsp + 8], rcx
0x10dab3:    xor     ecx, ecx
0x10dab5:    sub     qword ptr [rdx + 0x20], r12
0x10dab9:    mov     qword ptr [rdx + 0x18], rax
0x10dabd:    jmp     0x10d8fa
0x10d8e9:    mov     qword ptr [rsp + 8], 0
0x10d8f2:    mov     rax, qword ptr [rdx + 0x18]
0x10d8f6:    mov     rcx, qword ptr [rdx + 0x40]
0x10d8fa:    mov     rdi, qword ptr [rdx + 0x28]
0x10d8fe:    mov     r15, rdi
0x10d901:    sub     r15, rcx
0x10d904:    cmp     r15, 3
0x10d908:    jbe     0x10da80
0x10d8fa:    mov     rdi, qword ptr [rdx + 0x28]
0x10d8fe:    mov     r15, rdi
0x10d901:    sub     r15, rcx
0x10d904:    cmp     r15, 3
0x10d908:    jbe     0x10da80
0x10da80:    mov     ecx, 1
0x10da85:    mov     r15d, 1
0x10da8b:    jmp     0x10d915
0x10d90e:    shr     r15, 2
0x10d912:    movsxd  rcx, r15d
0x10d915:    cmp     rdi, rax
0x10d918:    ja      0x10da70
0x10d915:    cmp     rdi, rax

```

```

0x10d918:    ja     0x10da70
0x10d91e:    mov    edx, eax
0x10d920:    sub    rax, rdi
0x10d923:    sub    edx, edi
0x10d925:    add    r15d, edx
0x10d928:    xor    edx, edx
0x10d92a:    div    rcx
0x10d92d:    sub    r15d, edx
0x10d930:    mov    r12d, r15d
0x10d933:    call   0x11d670
0x10da70:    xor    r12d, r12d
0x10da73:    xor    r15d, r15d
0x10da76:    jmp    0x10d933
0x10d938:    mov    esi, 0x20
0x10d93d:    mov    rbp, rax
0x10d940:    mov    rax, qword ptr [rip + 0x1dac1]
0x10d947:    mov    rdi, rbp
0x10d94a:    mov    rdx, qword ptr [rax + 0x28]
0x10d94e:    call   0x104a50
0x10d933:    call   0x11d670
0x10d953:    mov    rax, qword ptr [rip + 0x1daae]
0x10d95a:    mov    edx, dword ptr [rax + 8]
0x10d95d:    test   edx, edx
0x10d95f:    je     0x10da40
0x10da40:    mov    rsi, qword ptr [rax + 0x40]
0x10da44:    mov    rdx, qword ptr [rax + 0x38]
0x10da48:    mov    rdi, rbp
0x10da4b:    mov    rax, qword ptr [rax + 0x28]
0x10da4f:    sub    rdx, rsi
0x10da52:    sub    rdx, r12
0x10da55:    sub    rax, rsi
0x10da58:    mov    esi, 0x2a
0x10da5d:    cmp    rdx, rax
0x10da60:    cmova  rdx, rax
0x10da64:    call   0x104a50
0x10d965:    mov    rdx, qword ptr [rax + 0x38]
0x10d969:    sub    rdx, qword ptr [rax + 0x40]
0x10d96d:    mov    rdi, rbp
0x10d970:    sub    rdx, r12
0x10d973:    mov    rsi, qword ptr [rax + 0x28]
0x10d977:    sub    rsi, qword ptr [rax + 0x40]
0x10d97b:    cmp    rdx, rsi
0x10d97e:    cmova  rdx, rsi
0x10d982:    mov    rsi, qword ptr [rax + 0x40]
0x10d986:    add    rsi, r12
0x10d989:    add    rsi, qword ptr [rax + 0x58]
0x10d98d:    call   0x104bb0
0x10da69:    jmp    0x10d992

```

```

0x10d992:    mov     rax, qword ptr [rip + 0x1da6f]
0x10d999:    xor     edx, edx
0x10d99b:    xor     esi, esi
0x10d99d:    mov     rdi, qword ptr [rax]
0x10d9a0:    call    0x10c680
0x10d9a5:    cmp     r13, rbx
0x10d9a8:    jae     0x10daf0
0x10d9ae:    mov     rax, qword ptr [rip + 0x1da53]
0x10d9b5:    mov     rsi, rbp
0x10d9b8:    mov     rdi, qword ptr [rax]
0x10d9bb:    mov     rdx, qword ptr [rax + 0x28]
0x10d9bf:    sub     edx, dword ptr [rax + 0x40]
0x10d9c2:    call    0x10c600
0x10daf0:    mov     rax, qword ptr [rip + 0x1d911]
0x10daf7:    mov     rsi, qword ptr [rax + 0x58]
0x10dafb:    mov     edx, dword ptr [rax + 0x40]
0x10dafe:    mov     rdi, qword ptr [rax]
0x10db01:    call    0x10c600
0x10d9c7:    mov     rax, qword ptr [rip + 0x1da3a]
0x10d9ce:    xor     esi, esi
0x10d9d0:    mov     edx, dword ptr [rax + 0x18]
0x10d9d3:    mov     rdi, qword ptr [rax]
0x10d9d6:    sub     edx, r15d
0x10d9d9:    call    0x10c680
0x10db06:    jmp     0x10d9ae
0x10d9de:    cmp     r13, rbx
0x10d9e1:    jb      0x10dac8
0x10dac8:    mov     rax, qword ptr [rip + 0x1d939]
0x10dacf:    movsxd  r12, dword ptr [rsp + 8]
0x10dad4:    sub     qword ptr [rax + 0x58], r12
0x10dad8:    add     qword ptr [rax + 0x38], r12
0x10dadc:    add     qword ptr [rax + 0x18], r12
0x10dae0:    add     qword ptr [rax + 0x20], r12
0x10dae4:    mov     qword ptr [rax + 0x40], r12
0x10dae8:    jmp     0x10d9e7
0x10d9e7:    mov     rdi, rbp
0x10d9ea:    call    0x118d10
0x10d9ef:    mov     rdi, r14
0x10d9f2:    call    0x10a6b0
0x10d9f7:    mov     rax, qword ptr [rsp + 0x18]
0x10d9fc:    sub     rax, qword ptr fs:[0x28]
0x10da05:    jne     0x10db0b
0x10da0b:    add     rsp, 0x28
0x10da0f:    pop     rbx
0x10da10:    pop     rbp
0x10da11:    pop     r12
0x10da13:    pop     r13
0x10da15:    pop     r14

```

```

0x10da17:    pop     r15
0x10da19:    ret
0x10db0b:    call    0x1049d0
0x10da1a:    nop     word ptr [rax + rax]
0x10da37:    nop     word ptr [rax + rax]
0x10da6e:    nop
0x10da7b:    nop     dword ptr [rax + rax]
0x10dac2:    nop     word ptr [rax + rax]
0x10daed:    nop     dword ptr [rax]
0x10db10:    endbr64
0x10db14:    mov     rax, qword ptr [rip + 0x1d8ed]
0x10db1b:    push    rbx
0x10db1c:    mov     rbx, rdi
0x10db1f:    mov     rdi, qword ptr [rdi]
0x10db22:    mov     rsi, qword ptr [rax + 0x48]
0x10db26:    add     rsi, 1
0x10db2a:    call    0x11d690
0x10db2f:    mov     qword ptr [rbx], rax
0x10db32:    mov     rdi, rax
0x10db35:    mov     rax, qword ptr [rip + 0x1d8cc]
0x10db3c:    mov     rcx, qword ptr [rax + 0x48]
0x10db40:    mov     rsi, qword ptr [rax + 0x40]
0x10db44:    add     rsi, qword ptr [rax + 0x58]
0x10db48:    lea     rdx, [rcx + 1]
0x10db4c:    call    0x104bb0
0x10db51:    mov     rax, qword ptr [rip + 0x1d8b0]
0x10db58:    pop     rbx
0x10db59:    mov     eax, dword ptr [rax + 0x48]
0x10db5c:    ret
0x10db5d:    nop     dword ptr [rax]
0x10db60:    endbr64
0x10db64:    push    rbp
0x10db65:    xor     eax, eax
0x10db67:    mov     rbp, rdi
0x10db6a:    call    0x10cff0
0x10db6f:    test    rbp, rbp
0x10db72:    lea     rax, [rip + 0x133bf]
0x10db79:    cmov     rbp, rax
0x10db7d:    mov     rdi, rbp
0x10db80:    call    0x10d540
0x10db85:    xor     eax, eax
0x10db87:    call    0x10d870
0x10db8c:    xor     eax, eax
0x10db8e:    pop     rbp
0x10db8f:    jmp     0x10d740
0x10db94:    nop     word ptr cs:[rax + rax]
0x10db9f:    nop
0x10dba0:    endbr64

```

```

0x10dba4:    mov     rax, qword ptr [rip + 0x1d85d]
0x10dbab:    mov     dword ptr [rax + 0xc], edi
0x10dbae:    ret
0x10dbaf:    nop
0x10dbb0:    endbr64
0x10dbb4:    push    r12
0x10dbb6:    push    rbp
0x10dbb7:    push    rbx
0x10dbb8:    cmp     edi, 1
0x10dbbb:    je      0x10dbd0
0x10dbd0:    mov     ecx, dword ptr [rip + 0x1d83e]
0x10dbd6:    test    ecx, ecx
0x10dbd8:    je      0x10dd38
0x10dbbd:    cmp     edi, 2
0x10dbc0:    je      0x10dc90
0x10dd38:    mov     dword ptr [rip + 0x1d6d2], 1
0x10dd42:    xor     eax, eax
0x10dd44:    call    0x11a570
0x10dbde:    xor     eax, eax
0x10dbe0:    call    0x11ab10
0x10dc90:    mov     edx, dword ptr [rip + 0x1d77e]
0x10dc96:    test    edx, edx
0x10dc98:    je      0x10dcf8
0x10dbc6:    test    edi, edi
0x10dbc8:    je      0x10dc40
0x10dd49:    mov     dword ptr [rip + 0x1d6c1], eax
0x10dd4f:    jmp     0x10dbde
0x10dbe5:    mov     rbp, rax
0x10dbe8:    test    rax, rax
0x10dbeb:    je      0x10dbca
0x10dcf8:    mov     dword ptr [rip + 0x1d712], 1
0x10dd02:    xor     eax, eax
0x10dd04:    call    0x11a570
0x10dc9a:    xor     eax, eax
0x10dc9c:    call    0x11ab40
0x10dc40:    mov     eax, dword ptr [rip + 0x1d7ce]
0x10dc46:    test    eax, eax
0x10dc48:    jne     0x10dd18
0x10dbca:    pop     rbx
0x10dbcb:    pop     rbp
0x10dbcc:    pop     r12
0x10dbce:    ret
0x10dbed:    mov     r12, qword ptr [rbp]
0x10dbf1:    mov     rbx, qword ptr [rip + 0x1d810]
0x10dbf8:    mov     rdi, r12
0x10dbfb:    call    0x1049b0
0x10dd09:    mov     dword ptr [rip + 0x1d701], eax
0x10dd0f:    jmp     0x10dc9a

```

```

0x10dca1:    mov     rbp, rax
0x10dca4:    test    rax, rax
0x10dca7:    jne     0x10dbed
0x10dc4e:    xor     eax, eax
0x10dc50:    call    0x11ab10
0x10dd18:    mov     edi, dword ptr [rip + 0x1d6f2]
0x10dd1e:    xor     eax, eax
0x10dd20:    call    0x11ac30
0x10dc00:    mov     rdi, qword ptr [rbx + 0x40]
0x10dc04:    mov     qword ptr [rbx + 0x48], rax
0x10dc08:    lea     rsi, [rax + rdi]
0x10dc0c:    mov     qword ptr [rbx + 0x38], rsi
0x10dc10:    mov     qword ptr [rbx + 0x18], rsi
0x10dc14:    add     rsi, 1
0x10dc18:    cmp     rsi, qword ptr [rbx + 0x50]
0x10dc1c:    ja      0x10dd70
0x10dcad:    mov     rax, qword ptr [rip + 0x1d754]
0x10dcb4:    mov     edi, 2
0x10dcb9:    mov     rdx, qword ptr [rax + 0x40]
0x10dcbd:    mov     dword ptr [rax + 0x10], 6
0x10dcc4:    mov     qword ptr [rax + 0x20], rdx
0x10dcc8:    mov     rdx, qword ptr [rax + 0x38]
0x10dccc:    mov     qword ptr [rax + 0x18], rdx
0x10dcd0:    xor     eax, eax
0x10dcd2:    call    0x10c6e0
0x10dc55:    test    rax, rax
0x10dc58:    je      0x10dd98
0x10dd25:    mov     dword ptr [rip + 0x1d6e5], 0
0x10dd2f:    jmp     0x10dc4e
0x10dc22:    add     rdi, qword ptr [rbx + 0x58]
0x10dc26:    mov     rsi, r12
0x10dc29:    call    0x1048b0
0x10dd70:    mov     qword ptr [rbx + 0x50], rsi
0x10dd74:    mov     rdi, qword ptr [rbx + 0x58]
0x10dd78:    call    0x11d690
0x10dcd7:    mov     rax, qword ptr [rip + 0x1d72a]
0x10dcde:    mov     rdx, qword ptr [rax + 0x18]
0x10dce2:    mov     dword ptr [rax + 0x10], 0xd
0x10dce9:    mov     qword ptr [rax + 0x20], rdx
0x10dced:    jmp     0x10dc2e
0x10dd98:    mov     rax, qword ptr [rip + 0x1d669]
0x10dd9f:    mov     rdi, qword ptr [rax + 0x40]
0x10dda3:    add     rdi, qword ptr [rax + 0x58]
0x10dda7:    xor     eax, eax
0x10dda9:    call    0x11a380
0x10dc5e:    mov     rdx, qword ptr [rip + 0x1d7a3]
0x10dc65:    mov     rsi, qword ptr [rax]
0x10dc68:    mov     rbp, qword ptr [rdx + 0x40]

```



```

0x10dc6c:      add     rbp, qword ptr [rdx + 0x58]
0x10dc70:      mov     rdi, rbp
0x10dc73:      call    0x104b30
0x10dc2e:      pop     rbx
0x10dc2f:      xor     eax, eax
0x10dc31:      pop     rbp
0x10dc32:      pop     r12
0x10dc34:      jmp     0x10d870
0x10dd7d:      mov     qword ptr [rbx + 0x58], rax
0x10dd81:      mov     rbx, qword ptr [rip + 0x1d680]
0x10dd88:      mov     r12, qword ptr [rbp]
0x10dd8c:      mov     rdi, qword ptr [rbx + 0x40]
0x10dd90:      jmp     0x10dc22
0x10ddae:      jmp     0x10dc80
0x10dc78:      test    eax, eax
0x10dc7a:      jne     0x10dd58
0x10dc80:      pop     rbx
0x10dc81:      xor     eax, eax
0x10dc83:      pop     rbp
0x10dc84:      pop     r12
0x10dc86:      jmp     0x11ab40
0x10dd58:      xor     eax, eax
0x10dd5a:      mov     rdi, rbp
0x10dd5d:      call    0x11a380
0x10dd62:      xor     eax, eax
0x10dd64:      call    0x11ab40
0x10dd69:      jmp     0x10dc80
0x10dbcf:      nop
0x10dc39:      nop     dword ptr [rax]
0x10dc8b:      nop     dword ptr [rax + rax]
0x10dcf2:      nop     word ptr [rax + rax]
0x10dd11:      nop     dword ptr [rax]
0x10dd34:      nop     dword ptr [rax]
0x10dd54:      nop     dword ptr [rax]
0x10dd6e:      nop
0x10dd95:      nop     dword ptr [rax]
0x10ddb3:      nop     word ptr cs:[rax + rax]
0x10dbbd:      nop     dword ptr [rax]
0x10ddc0:      endbr64
0x10ddc4:      mov     eax, dword ptr [rip + 0x1be96]
0x10ddca:      push    rbp
0x10ddcb:      mov     rbp, rdi
0x10ddce:      lea     rdi, [rip + 0x1be73]
0x10ddd5:      test    eax, eax
0x10ddd7:      lea     rax, [rip + 0x1be52]
0x10dddde:      cmove   rdi, rax
0x10dde2:      call    0x117c50
0x10dde7:      mov     edx, 9

```

```

0x10ddec:    lea     rsi, [rip + 0x1c4ad]
0x10ddf3:    lea     rdi, [rip + 0x1c466]
0x10ddfa:    call    0x117f70
0x10ddff:    mov     rdi, rbp
0x10de02:    call    0x11d7a0
0x10de07:    mov     rdi, rax
0x10de0a:    mov     rbp, rax
0x10de0d:    mov     qword ptr [rip + 0x1d794], rax
0x10de14:    call    0x1049b0
0x10de19:    mov     rdi, rbp
0x10de1c:    mov     rsi, rax
0x10de1f:    call    0x1184d0
0x10de24:    xor     eax, eax
0x10de26:    call    0x10c590
0x10de2b:    lea     rdi, [rip + 0x1d5ee]
0x10de32:    pop     rbp
0x10de33:    mov     qword ptr [rip + 0x1d78e], rax
0x10de3a:    jmp     0x104a30
0x10de3f:    nop
0x10de40:    endbr64
0x10de44:    mov     rdi, qword ptr [rip + 0x1d77d]
0x10de4b:    jmp     0x10c5c0
0x10de50:    endbr64
0x10de54:    push    rbp
0x10de55:    mov     rbp, rdi
0x10de58:    push    rbx
0x10de59:    mov     rbx, rsi
0x10de5c:    sub     rsp, 8
0x10de60:    mov     rdi, qword ptr [rip + 0x1d749]
0x10de67:    test    rdi, rdi
0x10de6a:    je      0x10de71
0x10de71:    mov     rdi, rbp
0x10de74:    call    0x11d670
0x10de7c:    call    0x118d10
0x10de79:    mov     rdi, qword ptr [rip + 0x1d748]
0x10de80:    mov     edx, ebx
0x10de82:    xor     esi, esi
0x10de84:    mov     qword ptr [rip + 0x1d725], rax
0x10de8b:    add     rsp, 8
0x10de8f:    mov     r8d, ebp
0x10de92:    mov     ecx, 1
0x10de97:    pop     rbx
0x10de98:    pop     rbp
0x10de99:    jmp     0x10c5e0
0x10de9e:    nop
0x10dea0:    endbr64
0x10dea4:    push    rbp
0x10dea5:    mov     rbp, rsp

```

```

0x10dea8:    push    r15
0x10deaa:    push    r14
0x10deac:    push    r13
0x10deae:    push    r12
0x10deb0:    push    rbx
0x10deb1:    sub     rsp, 0x68
0x10deb5:    mov     rax, qword ptr fs:[0x28]
0x10debe:    mov     qword ptr [rbp - 0x38], rax
0x10dec2:    xor     eax, eax
0x10dec4:    lea     rax, [rbp - 0x61]
0x10dec8:    mov     rdi, rax
0x10decb:    mov     qword ptr [rbp - 0x80], rax
0x10decf:    call    0x10a6a0
0x10ded4:    mov     rax, qword ptr [rip + 0x1d6ed]
0x10dedb:    mov     esi, 0x20
0x10dee0:    mov     rdi, qword ptr [rip + 0x1d6c9]
0x10dee7:    movsxd  rdx, dword ptr [rax + 0xc]
0x10deeb:    call    0x104a50
0x10def0:    mov     rdi, qword ptr [rip + 0x1d6c9]
0x10def7:    call    0x1049b0
0x10defc:    lea     r14, [rax + 1]
0x10df00:    mov     rdi, r14
0x10df03:    call    0x11d670
0x10df08:    mov     rcx, qword ptr [rip + 0x1d6b1]
0x10df0f:    mov     r13, rax
0x10df12:    movzx   eax, byte ptr [rcx]
0x10df15:    test    al, al
0x10df17:    je      0x10e430
0x10e430:    mov     rax, r13
0x10e433:    jmp     0x10dfc7
0x10df1d:    xor     esi, esi
0x10df1f:    xor     ebx, ebx
0x10df21:    lea     r15, [rip + 0x13188]
0x10df28:    xor     edx, edx
0x10df2a:    mov     dword ptr [rbp - 0x74], esi
0x10df2d:    jmp     0x10df48
0x10dfc7:    mov     byte ptr [rax], 0
0x10dfca:    mov     rdi, r13
0x10dfcd:    call    0x1049b0
0x10df48:    movsxd  r12, dword ptr [rbp - 0x74]
0x10df4c:    add     ebx, 1
0x10df4f:    cmp     al, 0x5c
0x10df51:    jne     0x10df30
0x10dfd2:    cmp     byte ptr [rip + 0x1d5df], 0
0x10dfd9:    mov     rdi, qword ptr [rip + 0x1d5d0]
0x10dfe0:    mov     rdx, rax
0x10dfe3:    mov     rax, qword ptr [rip + 0x1d5de]
0x10dfea:    mov     ecx, edx

```

```

0x10dfec:    jne     0x10e350
0x10df53:    lea     rdx, [rcx + rdx + 1]
0x10df58:    movzx   eax, byte ptr [rdx]
0x10df5b:    test    al, al
0x10df5d:    je      0x10e320
0x10df30:    cmp     al, 9
0x10df32:    je      0x10df80
0x10dff2:    movsxd  rax, dword ptr [rax + 0xc]
0x10dff6:    cmp     edx, eax
0x10dff8:    jl      0x10e120
0x10e350:    mov     eax, dword ptr [rax + 0xc]
0x10e353:    jmp     0x10df80
0x10e320:    add     dword ptr [rbp - 0x74], 1
0x10e324:    mov     byte ptr [r13 + r12], 0x5c
0x10e32a:    jmp     0x10df3d
0x10df63:    sub     eax, 0x5c
0x10df66:    cmp     al, 0x17
0x10df68:    ja      0x10e0d8
0x10df80:    add     r14, 8
0x10df84:    mov     rdi, r13
0x10df87:    mov     rsi, r14
0x10df8a:    call    0x11d690
0x10df34:    add     dword ptr [rbp - 0x74], 1
0x10df38:    mov     byte ptr [r13 + r12], al
0x10df3d:    movsxd  rdx, ebx
0x10df40:    movzx   eax, byte ptr [rcx + rdx]
0x10df44:    test    al, al
0x10df46:    je      0x10dfc0
0x10e120:    sub     rax, rdx
0x10e123:    mov     rsi, r13
0x10e126:    shr     rax, 1
0x10e129:    add     rdi, rax
0x10e12c:    call    0x104bb0
0x10df8e:    cmp     ecx, eax
0x10e000:    mov     rsi, r13
0x10e003:    cmovle  eax, ecx
0x10e006:    movsxd  rdx, eax
0x10e009:    call    0x104bb0
0x10df3d:    movsxd  rdx, ebx
0x10df40:    movzx   eax, byte ptr [rcx + rdx]
0x10df44:    test    al, al
0x10df46:    je      0x10dfc0
0x10df6e:    movzx   eax, al
0x10df71:    movsxd  rax, dword ptr [r15 + rax*4]
0x10df75:    add     rax, r15
0x10df78:    jmp     rax
0x10e0d8:    mov     edi, dword ptr [rbp - 0x74]
0x10e0db:    mov     byte ptr [r13 + r12], 0x5c

```

```

0x10e0e1:      add     ebx, 1
0x10e0e4:      movzx   edx, byte ptr [rdx]
0x10e0e7:      lea     eax, [rdi + 1]
0x10e0ea:      add     edi, 2
0x10e0ed:      cdqe
0x10e0ef:      mov     dword ptr [rbp - 0x74], edi
0x10e0f2:      mov     byte ptr [r13 + rax], dl
0x10e0f7:      jmp     0x10df3d
0x10df8f:      mov     rcx, qword ptr [rip + 0x1d62a]
0x10df96:      movsxd   rdx, ebx
0x10df99:      add     dword ptr [rbp - 0x74], 8
0x10df9d:      mov     r13, rax
0x10dfa0:      movabs   rax, 0x2020202020202020
0x10dfaa:      mov     qword ptr [r13 + r12], rax
0x10dfaf:      movzx   eax, byte ptr [rcx + rdx]
0x10dfb3:      test    al, al
0x10dfb5:      jne     0x10df48
0x10dfc0:      movsxd   rax, dword ptr [rbp - 0x74]
0x10dfc4:      add     rax, r13
0x10dfc7:      mov     byte ptr [rax], 0
0x10dfca:      mov     rdi, r13
0x10dfcd:      call    0x1049b0
0x10e131:      jmp     0x10e00e
0x10e00e:      mov     rdi, r13
0x10e011:      call    0x118d10
0x10e1c0:      movzx   eax, byte ptr [rip + 0x1d35d]
0x10e1c7:      lea     rcx, [rip + 0x1d356]
0x10e1ce:      test    al, al
0x10e1d0:      jne     0x10e155
0x10e140:      lea     rax, [rip + 0x1d2d9]
0x10e147:      lea     rcx, [rip + 0x1d2d2]
0x10e14e:      movzx   eax, byte ptr [rax]
0x10e151:      test    al, al
0x10e153:      je      0x10e1d2
0x10e308:      movzx   eax, byte ptr [rip + 0x1d152]
0x10e30f:      lea     rcx, [rip + 0x1d14b]
0x10e316:      jmp     0x10e151
0x10e2f0:      add     dword ptr [rbp - 0x74], 1
0x10e2f4:      add     ebx, 1
0x10e2f7:      mov     byte ptr [r13 + r12], 0x5c
0x10e2fd:      jmp     0x10df3d
0x10e218:      xor     eax, eax
0x10e21a:      call    0x118330
0x10dfb7:      nop     word ptr [rax + rax]
0x10e016:      mov     rsi, qword ptr [rip + 0x1d5ab]
0x10e01d:      mov     rcx, qword ptr [rip + 0x1d58c]
0x10e024:      xor     eax, eax
0x10e026:      mov     edx, dword ptr [rsi + 0xc]

```

```

0x10e029:    test    edx, edx
0x10e02b:    jg      0x10e039
0x10e1d2:    add     r14, 6
0x10e1d6:    mov     rdi, r13
0x10e1d9:    add     ebx, 1
0x10e1dc:    mov     rsi, r14
0x10e1df:    call    0x11d690
0x10e155:    mov     rdi, rcx
0x10e158:    mov     qword ptr [rbp - 0x90], rcx
0x10e15f:    call    0x1049b0
0x10e151:    test    al, al
0x10e153:    je      0x10e1d2
0x10e21f:    lea     r10, [rbp - 0x60]
0x10e223:    mov     edi, dword ptr [rax + 0x14]
0x10e226:    movsxd  rsi, dword ptr [rax + 0x10]
0x10e22a:    movsxd  rcx, dword ptr [rax + 0x18]
0x10e22e:    lea     edx, [rdi + 0x76c]
0x10e234:    mov     rdi, r10
0x10e237:    push    rdx
0x10e238:    mov     eax, dword ptr [rax + 0xc]
0x10e23b:    mov     edx, 0x20
0x10e240:    push    rax
0x10e241:    lea     rax, [rip + 0x1c4b8]
0x10e248:    mov     r9, qword ptr [rax + rsi*8]
0x10e24c:    lea     rax, [rip + 0x1c50d]
0x10e253:    mov     esi, 1
0x10e258:    mov     r8, qword ptr [rax + rcx*8]
0x10e25c:    lea     rcx, [rip + 0x12d65]
0x10e263:    xor     eax, eax
0x10e265:    mov     qword ptr [rbp - 0x88], r10
0x10e26c:    call    0x104f10
0x10e02d:    jmp     0x10e054
0x10e039:    movzx   edx, byte ptr [rcx + rax]
0x10e03d:    cmp     dl, 0xd
0x10e040:    je      0x10e047
0x10e1e4:    mov     edi, 0x2965
0x10e1e9:    mov     edx, 6
0x10e1ee:    add     dword ptr [rbp - 0x74], edx
0x10e1f1:    add     r12, rax
0x10e1f4:    mov     rcx, qword ptr [rip + 0x1d3c5]
0x10e1fb:    mov     r13, rax
0x10e1fe:    mov     dword ptr [r12], 0x6e6f6e28
0x10e206:    mov     word ptr [r12 + 4], di
0x10e20c:    jmp     0x10df3d
0x10e164:    mov     rdi, r13
0x10e167:    add     r14, rax
0x10e16a:    mov     qword ptr [rbp - 0x88], rax
0x10e171:    mov     rsi, r14

```

```

0x10e174:    call    0x11d690
0x10e271:    mov     r10, qword ptr [rbp - 0x88]
0x10e278:    mov     rdi, r10
0x10e27b:    mov     qword ptr [rbp - 0x90], r10
0x10e282:    call    0x1049b0
0x10e054:    movzx   eax, byte ptr [rip + 0x1d55e]
0x10e05b:    cmp     al, 1
0x10e05d:    je      0x10e330
0x10e047:    mov     byte ptr [rcx + rax], 0x20
0x10e04b:    add     rax, 1
0x10e04f:    cmp     dword ptr [rsi + 0xc], eax
0x10e052:    jg      0x10e039
0x10e042:    cmp     dl, 0xa
0x10e045:    jne     0x10e030
0x10e179:    mov     rdx, qword ptr [rbp - 0x88]
0x10e180:    mov     rcx, qword ptr [rbp - 0x90]
0x10e187:    add     r12, rax
0x10e18a:    mov     r13, rax
0x10e18d:    cmp     edx, 8
0x10e190:    jae     0x10e360
0x10e287:    mov     rdi, r13
0x10e28a:    add     r14, rax
0x10e28d:    mov     qword ptr [rbp - 0x88], rax
0x10e294:    mov     rsi, r14
0x10e297:    call    0x11d690
0x10e330:    mov     edx, dword ptr [rip + 0x1bf3a]
0x10e336:    mov     esi, dword ptr [rip + 0x1bf30]
0x10e33c:    mov     edi, dword ptr [rip + 0x1bf32]
0x10e342:    call    0x10a570
0x10e063:    cmp     al, 2
0x10e065:    jne     0x10e100
0x10e030:    add     rax, 1
0x10e034:    cmp     dword ptr [rsi + 0xc], eax
0x10e037:    jle     0x10e054
0x10e196:    test    dl, 4
0x10e199:    jne     0x10e438
0x10e360:    mov     rax, qword ptr [rcx]
0x10e363:    lea     rdi, [r12 + 8]
0x10e368:    add     ebx, 1
0x10e36b:    and     rdi, 0xfffffffffffffff8
0x10e36f:    mov     qword ptr [r12], rax
0x10e373:    mov     eax, edx
0x10e375:    mov     rsi, qword ptr [rcx + rax - 8]
0x10e37a:    mov     qword ptr [r12 + rax - 8], rsi
0x10e37f:    sub     r12, rdi
0x10e382:    mov     rsi, rcx
0x10e385:    sub     rsi, r12
0x10e388:    add     r12d, edx

```

```

0x10e38b:    shr     r12d, 3
0x10e38f:    mov     ecx, r12d
0x10e392:    rep movsq      qword ptr [rdi], qword ptr [rsi]
0x10e29c:    mov     rdx, qword ptr [rbp - 0x88]
0x10e2a3:    mov     r10, qword ptr [rbp - 0x90]
0x10e2aa:    add     r12, rax
0x10e2ad:    mov     r13, rax
0x10e2b0:    cmp     edx, 8
0x10e2b3:    jae     0x10e3a8
0x10e347:    jmp     0x10e082
0x10e06b:    mov     edx, dword ptr [rip + 0x1c20b]
0x10e071:    mov     esi, dword ptr [rip + 0x1c201]
0x10e077:    mov     edi, dword ptr [rip + 0x1c203]
0x10e07d:    call    0x10a570
0x10e100:    mov     edx, dword ptr [rip + 0x1c15e]
0x10e106:    mov     esi, dword ptr [rip + 0x1c154]
0x10e10c:    mov     edi, dword ptr [rip + 0x1c156]
0x10e112:    call    0x10a570
0x10e19f:    test    edx, edx
0x10e1a1:    jne     0x10e408
0x10e438:    mov     eax, dword ptr [rcx]
0x10e43a:    mov     dword ptr [r12], eax
0x10e43e:    mov     eax, edx
0x10e440:    mov     ecx, dword ptr [rcx + rax - 4]
0x10e444:    mov     dword ptr [r12 + rax - 4], ecx
0x10e449:    jmp     0x10e1a7
0x10e395:    add     dword ptr [rbp - 0x74], edx
0x10e398:    mov     rcx, qword ptr [rip + 0x1d221]
0x10e39f:    jmp     0x10df3d
0x10e392:    rep movsq      qword ptr [rdi], qword ptr [rsi]
0x10e2b9:    test    dl, 4
0x10e2bc:    jne     0x10e44e
0x10e3a8:    mov     rax, qword ptr [rbp - 0x60]
0x10e3ac:    lea     rsi, [r12 + 8]
0x10e3b1:    and     rsi, 0xfffffffffffffff8
0x10e3b5:    mov     qword ptr [r12], rax
0x10e3b9:    mov     eax, edx
0x10e3bb:    mov     rcx, qword ptr [r10 + rax - 8]
0x10e3c0:    mov     qword ptr [r12 + rax - 8], rcx
0x10e3c5:    sub     r12, rsi
0x10e3c8:    sub     r10, r12
0x10e3cb:    add     r12d, edx
0x10e3ce:    and     r12d, 0xffffffff8
0x10e3d2:    cmp     r12d, 8
0x10e3d6:    jb     0x10e2d7
0x10e082:    mov     rdi, qword ptr [rip + 0x1d53f]
0x10e089:    xor     edx, edx
0x10e08b:    xor     esi, esi

```



```

0x10e08d:    call    0x10c680
0x10e117:    jmp     0x10e082
0x10e1a7:    add     dword ptr [rbp - 0x74], edx
0x10e1aa:    mov     rcx, qword ptr [rip + 0x1d40f]
0x10e1b1:    add     ebx, 1
0x10e1b4:    jmp     0x10df3d
0x10e408:    movzx   eax, byte ptr [rcx]
0x10e40b:    mov     byte ptr [r12], al
0x10e40f:    test    dl, 2
0x10e412:    je      0x10e1a7
0x10e2c2:    test    edx, edx
0x10e2c4:    je      0x10e2d7
0x10e44e:    mov     eax, dword ptr [rbp - 0x60]
0x10e451:    mov     dword ptr [r12], eax
0x10e455:    mov     eax, edx
0x10e457:    mov     ecx, dword ptr [r10 + rax - 4]
0x10e45c:    mov     dword ptr [r12 + rax - 4], ecx
0x10e461:    jmp     0x10e2d7
0x10e2d7:    pop     rcx
0x10e2d8:    add     dword ptr [rbp - 0x74], edx
0x10e2db:    add     ebx, 1
0x10e2de:    mov     rcx, qword ptr [rip + 0x1d2db]
0x10e2e5:    pop     rsi
0x10e2e6:    jmp     0x10df3d
0x10e3dc:    mov     r11d, dword ptr [rbp - 0x74]
0x10e3e0:    and     r12d, 0xffffffff8
0x10e3e4:    xor     eax, eax
0x10e3e6:    mov     ecx, eax
0x10e3e8:    add     eax, 8
0x10e3eb:    mov     rdi, qword ptr [r10 + rcx]
0x10e3ef:    mov     qword ptr [rsi + rcx], rdi
0x10e3f3:    cmp     eax, r12d
0x10e3f6:    jb      0x10e3e6
0x10e092:    mov     rdi, qword ptr [rip + 0x1d52f]
0x10e099:    mov     rsi, qword ptr [rip + 0x1d510]
0x10e0a0:    mov     edx, dword ptr [rdi + 0xc]
0x10e0a3:    call    0x10c600
0x10e418:    mov     eax, edx
0x10e41a:    movzx   ecx, word ptr [rcx + rax - 2]
0x10e41f:    mov     word ptr [r12 + rax - 2], cx
0x10e425:    jmp     0x10e1a7
0x10e2c6:    movzx   eax, byte ptr [rbp - 0x60]
0x10e2ca:    mov     byte ptr [r12], al
0x10e2ce:    test    dl, 2
0x10e2d1:    jne     0x10e466
0x10e3e6:    mov     ecx, eax
0x10e3e8:    add     eax, 8
0x10e3eb:    mov     rdi, qword ptr [r10 + rcx]

```

```

0x10e3ef:    mov     qword ptr [rsi + rcx], rdi
0x10e3f3:    cmp     eax, r12d
0x10e3f6:    jb      0x10e3e6
0x10e3f8:    mov     dword ptr [rbp - 0x74], r11d
0x10e3fc:    jmp     0x10e2d7
0x10e0a8:    mov     rdi, qword ptr [rbp - 0x80]
0x10e0ac:    call    0x10a6b0
0x10e466:    mov     eax, edx
0x10e468:    movzx   ecx, word ptr [r10 + rax - 2]
0x10e46e:    mov     word ptr [r12 + rax - 2], cx
0x10e474:    jmp     0x10e2d7
0x10e0b1:    mov     rax, qword ptr [rbp - 0x38]
0x10e0b5:    sub     rax, qword ptr fs:[0x28]
0x10e0be:    jne     0x10e479
0x10e0c4:    lea     rsp, [rbp - 0x28]
0x10e0c8:    pop     rbx
0x10e0c9:    pop     r12
0x10e0cb:    pop     r13
0x10e0cd:    pop     r14
0x10e0cf:    pop     r15
0x10e0d1:    pop     rbp
0x10e0d2:    ret
0x10e479:    call    0x1049d0
0x10df2f:    nop
0x10df7b:    nop     dword ptr [rax + rax]
0x10e02f:    nop
0x10e0d3:    nop     dword ptr [rax + rax]
0x10e0fc:    nop     dword ptr [rax]
0x10e11c:    nop     dword ptr [rax]
0x10e136:    nop     word ptr cs:[rax + rax]
0x10e1b9:    nop     dword ptr [rax]
0x10e211:    nop     dword ptr [rax]
0x10e2eb:    nop     dword ptr [rax + rax]
0x10e302:    nop     word ptr [rax + rax]
0x10e31b:    nop     dword ptr [rax + rax]
0x10e32f:    nop
0x10e34c:    nop     dword ptr [rax]
0x10e358:    nop     dword ptr [rax + rax]
0x10e3a4:    nop     dword ptr [rax]
0x10e401:    nop     dword ptr [rax]
0x10e42a:    nop     word ptr [rax + rax]
0x10e47e:    nop
0x10e480:    endbr64
0x10e484:    push    r12
0x10e486:    mov     r12d, esi
0x10e489:    push    rbp
0x10e48a:    mov     rbp, rdi
0x10e48d:    mov     rdi, qword ptr [rip + 0x1d12c]

```

```

0x10e494:    push    rbx
0x10e495:    mov     ebx, edx
0x10e497:    test    rdi, rdi
0x10e49a:    je      0x10e4a1
0x10e4a1:    mov     rdi, rbp
0x10e4a4:    call    0x11d7a0
0x10e49c:    call    0x118d10
0x10e4a9:    mov     rdi, rax
0x10e4ac:    mov     rbp, rax
0x10e4af:    mov     qword ptr [rip + 0x1d10a], rax
0x10e4b6:    call    0x1049b0
0x10e4bb:    mov     rdi, rbp
0x10e4be:    mov     rsi, rax
0x10e4c1:    call    0x1184d0
0x10e4c6:    mov     byte ptr [rip + 0x1d0ec], r12b
0x10e4cd:    xor     eax, eax
0x10e4cf:    mov     byte ptr [rip + 0x1d0e3], bl
0x10e4d5:    pop     rbx
0x10e4d6:    pop     rbp
0x10e4d7:    pop     r12
0x10e4d9:    jmp     0x10dea0
0x10e4de:    nop
0x10e4e0:    endbr64
0x10e4e4:    push    rbp
0x10e4e5:    mov     rdi, qword ptr [rip + 0x1d0bc]
0x10e4ec:    call    0x11d7a0
0x10e4f1:    mov     rdi, qword ptr [rip + 0x1d0c8]
0x10e4f8:    mov     rbp, rax
0x10e4fb:    test    rdi, rdi
0x10e4fe:    je      0x10e505
0x10e505:    mov     rdi, rbp
0x10e508:    call    0x11d7a0
0x10e500:    call    0x118d10
0x10e50d:    mov     rdi, rax
0x10e510:    mov     rbp, rax
0x10e513:    mov     qword ptr [rip + 0x1d0a6], rax
0x10e51a:    call    0x1049b0
0x10e51f:    mov     rdi, rbp
0x10e522:    mov     rsi, rax
0x10e525:    call    0x1184d0
0x10e52a:    xor     eax, eax
0x10e52c:    pop     rbp
0x10e52d:    mov     byte ptr [rip + 0x1d085], 0
0x10e534:    mov     byte ptr [rip + 0x1d07d], 0
0x10e53b:    jmp     0x10dea0
0x10e540:    endbr64
0x10e544:    mov     rax, qword ptr [rsi]
0x10e547:    mov     rdx, qword ptr [rdi]

```

```

0x10e54a:    mov     rax, qword ptr [rax + 8]
0x10e54e:    sub     eax, dword ptr [rdx + 8]
0x10e551:    ret
0x10e552:    nop     word ptr cs:[rax + rax]
0x10e55d:    nop     dword ptr [rax]
0x10e560:    push    rbp
0x10e561:    mov     rbp, rdi
0x10e564:    mov     rdi, rdx
0x10e567:    push    rbx
0x10e568:    mov     ebx, esi
0x10e56a:    lea     rsi, [rip + 0x1bdff]
0x10e571:    sub     rsp, 8
0x10e575:    call    0x1048b0
0x10e57a:    movsxd  rsi, ebx
0x10e57d:    mov     rcx, rax
0x10e580:    lea     rsi, [rsi + rsi*4]
0x10e584:    mov     rax, qword ptr [rbp + 0x38]
0x10e588:    shl     rsi, 4
0x10e58c:    mov     edx, dword ptr [rax + rsi + 0x10]
0x10e590:    mov     eax, 0x2d
0x10e595:    mov     edi, edx
0x10e597:    and     edi, 0xf000
0x10e59d:    cmp     edi, 0x8000
0x10e5a3:    je      0x10e600
0x10e600:    mov     byte ptr [rcx], al
0x10e602:    lea     rax, [rcx + 9]
0x10e606:    nop     word ptr cs:[rax + rax]
0x10e610:    test    dl, 1
0x10e613:    jne     0x10e618
0x10e5a5:    mov     eax, 0x64
0x10e5aa:    cmp     edi, 0x4000
0x10e5b0:    je      0x10e600
0x10e615:    mov     byte ptr [rax], 0x2d
0x10e618:    sub     rax, 1
0x10e61c:    shr     edx, 1
0x10e61e:    cmp     rax, rcx
0x10e621:    jne     0x10e610
0x10e618:    sub     rax, 1
0x10e61c:    shr     edx, 1
0x10e61e:    cmp     rax, rcx
0x10e621:    jne     0x10e610
0x10e5b2:    mov     eax, 0x63
0x10e5b7:    cmp     edi, 0x2000
0x10e5bd:    je      0x10e600
0x10e623:    mov     rax, qword ptr [rbp + 0x38]
0x10e627:    mov     eax, dword ptr [rax + rsi + 0x10]
0x10e62b:    test    ah, 8
0x10e62e:    je      0x10e644

```

```

0x10e610:    test    dl, 1
0x10e613:    jne     0x10e618
0x10e5bf:    mov     eax, 0x62
0x10e5c4:    cmp     edi, 0x6000
0x10e5ca:    je      0x10e600
0x10e644:    test    ah, 4
0x10e647:    je      0x10e65d
0x10e630:    cmp     byte ptr [rcx + 3], 0x78
0x10e634:    mov     edx, 0x73
0x10e639:    mov     esi, 0x53
0x10e63e:    cmovne  edx, esi
0x10e641:    mov     byte ptr [rcx + 3], dl
0x10e644:    test    ah, 4
0x10e647:    je      0x10e65d
0x10e5cc:    mov     eax, 0x70
0x10e5d1:    cmp     edi, 0x1000
0x10e5d7:    je      0x10e600
0x10e65d:    test    ah, 2
0x10e660:    je      0x10e676
0x10e649:    cmp     byte ptr [rcx + 6], 0x78
0x10e64d:    mov     edx, 0x73
0x10e652:    mov     esi, 0x53
0x10e657:    cmovne  edx, esi
0x10e65a:    mov     byte ptr [rcx + 6], dl
0x10e65d:    test    ah, 2
0x10e660:    je      0x10e676
0x10e5d9:    mov     eax, 0x73
0x10e5de:    cmp     edi, 0xc000
0x10e5e4:    je      0x10e600
0x10e676:    add     rsp, 8
0x10e67a:    pop     rbx
0x10e67b:    pop     rbp
0x10e67c:    ret
0x10e662:    cmp     byte ptr [rcx + 9], 0x78
0x10e666:    mov     eax, 0x74
0x10e66b:    mov     edx, 0x54
0x10e670:    cmovne  eax, edx
0x10e673:    mov     byte ptr [rcx + 9], al
0x10e676:    add     rsp, 8
0x10e67a:    pop     rbx
0x10e67b:    pop     rbp
0x10e67c:    ret
0x10e5e6:    cmp     edi, 0xa000
0x10e5ec:    mov     eax, 0x6c
0x10e5f1:    mov     edi, 0x3f
0x10e5f6:    cmovne  eax, edi
0x10e5f9:    nop     dword ptr [rax]
0x10e600:    mov     byte ptr [rcx], al

```

```

0x10e602:    lea     rax, [rcx + 9]
0x10e606:    nop     word ptr cs:[rax + rax]
0x10e610:    test    dl, 1
0x10e613:    jne     0x10e618
0x10e67d:    nop     dword ptr [rax]
0x10e680:    endbr64
0x10e684:    push    r13
0x10e686:    push    r12
0x10e688:    push    rbp
0x10e689:    push    rbx
0x10e68a:    sub     rsp, 8
0x10e68e:    movzx   eax, byte ptr [rdi + 0x48]
0x10e692:    test    al, al
0x10e694:    sete    cl
0x10e697:    cmp     byte ptr [rsi + 0x48], 0
0x10e69b:    sete    dl
0x10e69e:    cmp     cl, dl
0x10e6a0:    je      0x10e6c0
0x10e6c0:    mov     r12d, dword ptr [rip + 0x1cf61]
0x10e6c7:    mov     rbx, rdi
0x10e6ca:    mov     rbp, rsi
0x10e6cd:    cmp     r12d, 8
0x10e6d1:    ja      0x10e6e7
0x10e6a2:    cmp     al, 1
0x10e6a4:    sbb     r12d, r12d
0x10e6a7:    or      r12d, 1
0x10e6ab:    add     rsp, 8
0x10e6af:    mov     eax, r12d
0x10e6b2:    pop     rbx
0x10e6b3:    pop     rbp
0x10e6b4:    pop     r12
0x10e6b6:    pop     r13
0x10e6b8:    ret
0x10e6d3:    lea     rdx, [rip + 0x131c6]
0x10e6da:    mov     eax, r12d
0x10e6dd:    movsxd  rax, dword ptr [rdx + rax*4]
0x10e6e1:    add     rax, rdx
0x10e6e4:    jmp     rax
0x10e6e7:    lea     rdi, [rip + 0x12a22]
0x10e6ee:    call    0x108b00
0x10e760:    mov     r12d, dword ptr [rdi + 0x10]
0x10e764:    mov     eax, dword ptr [rsi + 0x10]
0x10e767:    cmp     r12d, eax
0x10e76a:    jne     0x10e734
0x10e7a8:    mov     r13, qword ptr [rdi]
0x10e7ab:    mov     esi, 0x2e
0x10e7b0:    mov     rdi, r13
0x10e7b3:    call    0x104a20

```

```

0x10e788:    mov     rdx, qword ptr [rdi + 8]
0x10e78c:    mov     rax, qword ptr [rsi + 8]
0x10e790:    mov     r12d, edx
0x10e793:    sub     r12d, eax
0x10e796:    cmp     rdx, rax
0x10e799:    jne     0x10e6ab
0x10e748:    mov     r12d, dword ptr [rdi + 0x14]
0x10e74c:    mov     eax, dword ptr [rsi + 0x14]
0x10e74f:    cmp     r12d, eax
0x10e752:    jne     0x10e734
0x10e728:    mov     r12d, dword ptr [rdi + 0x18]
0x10e72c:    mov     eax, dword ptr [rsi + 0x18]
0x10e72f:    cmp     r12d, eax
0x10e732:    je      0x10e6f3
0x10e770:    mov     r12, qword ptr [rdi + 0x20]
0x10e774:    mov     rax, qword ptr [rsi + 0x20]
0x10e778:    cmp     r12, rax
0x10e77b:    jne     0x10e734
0x10e710:    mov     eax, dword ptr [rsi + 0x14]
0x10e713:    cmp     dword ptr [rdi + 0x14], eax
0x10e716:    je      0x10e6f3
0x10e6f3:    mov     rbp, qword ptr [rbp]
0x10e6f7:    mov     r13, qword ptr [rbx]
0x10e6fa:    mov     rsi, rbp
0x10e6fd:    mov     rdi, r13
0x10e700:    add     rsp, 8
0x10e704:    pop     rbx
0x10e705:    pop     rbp
0x10e706:    pop     r12
0x10e708:    pop     r13
0x10e70a:    jmp     0x104b30
0x10e7f8:    mov     eax, dword ptr [rsi + 0x18]
0x10e7fb:    cmp     dword ptr [rdi + 0x18], eax
0x10e7fe:    je      0x10e6f3
0x10e76c:    jmp     0x10e6f3
0x10e734:    add     rsp, 8
0x10e738:    sub     r12d, eax
0x10e73b:    pop     rbx
0x10e73c:    mov     eax, r12d
0x10e73f:    pop     rbp
0x10e740:    pop     r12
0x10e742:    pop     r13
0x10e744:    ret
0x10e7b8:    mov     rbp, qword ptr [rbp]
0x10e7bc:    mov     esi, 0x2e
0x10e7c1:    mov     rbx, rax
0x10e7c4:    mov     rdi, rbp
0x10e7c7:    call    0x104a20

```

```

0x10e79f:      jmp      0x10e6f3
0x10e6ab:      add     rsp, 8
0x10e6af:      mov     eax, r12d
0x10e6b2:      pop     rbx
0x10e6b3:      pop     rbp
0x10e6b4:      pop     r12
0x10e6b6:      pop     r13
0x10e6b8:      ret
0x10e754:      jmp     0x10e6f3
0x10e77d:      jmp     0x10e6f3
0x10e718:      mov     rsi, qword ptr [rsi + 0x28]
0x10e71c:      mov     rdi, qword ptr [rdi + 0x28]
0x10e720:      jmp     0x10e700
0x10e804:      mov     rsi, qword ptr [rsi + 0x30]
0x10e808:      mov     rdi, qword ptr [rdi + 0x30]
0x10e80c:      jmp     0x10e700
0x10e7cc:      test    rbx, rbx
0x10e7cf:      je      0x10e818
0x10e700:      add     rsp, 8
0x10e704:      pop     rbx
0x10e705:      pop     rbp
0x10e706:      pop     r12
0x10e708:      pop     r13
0x10e70a:      jmp     0x104b30
0x10e818:      or      rax, rbx
0x10e81b:      je      0x10e6fa
0x10e7d1:      test    rax, rax
0x10e7d4:      je      0x10e818
0x10e6fa:      mov     rsi, rbp
0x10e6fd:      mov     rdi, r13
0x10e700:      add     rsp, 8
0x10e704:      pop     rbx
0x10e705:      pop     rbp
0x10e706:      pop     r12
0x10e708:      pop     r13
0x10e70a:      jmp     0x104b30
0x10e821:      test    rbx, rbx
0x10e824:      mov     eax, 0xffffffff
0x10e829:      cmovne  r12d, eax
0x10e82d:      jmp     0x10e6ab
0x10e7d6:      lea     rsi, [rax + 1]
0x10e7da:      lea     rdi, [rbx + 1]
0x10e7de:      call    0x104b30
0x10e7e3:      mov     r12d, eax
0x10e7e6:      test    eax, eax
0x10e7e8:      je      0x10e6fa
0x10e7ee:      jmp     0x10e6ab
0x10e6b9:      nop     dword ptr [rax]

```



```

0x10e70f:      nop
0x10e722:      nop      word ptr [rax + rax]
0x10e745:      nop      dword ptr [rax]
0x10e756:      nop      word ptr cs:[rax + rax]
0x10e76e:      nop
0x10e782:      nop      word ptr [rax + rax]
0x10e7a4:      nop      dword ptr [rax]
0x10e7f3:      nop      dword ptr [rax + rax]
0x10e811:      nop      dword ptr [rax]
0x10e832:      nop      word ptr cs:[rax + rax]
0x10e83d:      nop      dword ptr [rax]
0x10e840:      push     r13
0x10e842:      mov      ecx, 0x12
0x10e847:      push     r12
0x10e849:      push     rbp
0x10e84a:      mov      rbp, rdi
0x10e84d:      push     rbx
0x10e84e:      sub      rsp, 0x138
0x10e855:      mov      rax, qword ptr fs:[0x28]
0x10e85e:      mov      qword ptr [rsp + 0x128], rax
0x10e866:      xor      eax, eax
0x10e868:      mov      r8, rsp
0x10e86b:      mov      rdi, r8
0x10e86e:      rep stosq      qword ptr [rdi], rax
0x10e871:      movsxd   rax, esi
0x10e874:      mov      rsi, r8
0x10e877:      lea      rbx, [rax + rax*4]
0x10e87b:      mov      rax, qword ptr [rbp + 0x38]
0x10e87f:      shl      rbx, 4
0x10e883:      mov      rdi, qword ptr [rax + rbx]
0x10e887:      call     0x118f90
0x10e86e:      rep stosq      qword ptr [rdi], rax
0x10e88c:      mov      rax, qword ptr [rbp + 0x38]
0x10e890:      mov      edx, dword ptr [rsp + 0x18]
0x10e894:      mov      edi, dword ptr [rsp + 0x1c]
0x10e898:      mov      ecx, dword ptr [rsp + 0x20]
0x10e89c:      add      rax, rbx
0x10e89f:      cmp      qword ptr [rsp + 8], 0
0x10e8a5:      mov      dword ptr [rax + 0x10], edx
0x10e8a8:      mov      dword ptr [rax + 0x14], edi
0x10e8ab:      mov      dword ptr [rax + 0x18], ecx
0x10e8ae:      je       0x10eab0
0x10eab0:      mov      byte ptr [rax + 0x48], 2
0x10eab4:      mov      rdi, qword ptr [rax]
0x10eab7:      call     0x118e80
0x10e8b4:      mov      ecx, edx
0x10e8b6:      and      ecx, 0xf000
0x10e8bc:      cmp      ecx, 0x4000

```

```

0x10e8c2:    je      0x10e908
0x10eabc:    xor     edx, edx
0x10eabe:    mov     edi, dword ptr [rsp + 0x1c]
0x10eac2:    cmp     eax, -1
0x10eac5:    cmovne  eax, edx
0x10eac8:    mov     rdx, qword ptr [rbp + 0x38]
0x10eacc:    cdqe
0x10eace:    mov     qword ptr [rdx + rbx + 8], rax
0x10ead3:    call    0x11d310
0x10e908:    mov     byte ptr [rax + 0x48], 0
0x10e90c:    mov     rdx, qword ptr [rsp + 0x30]
0x10e911:    mov     qword ptr [rax + 8], rdx
0x10e915:    call    0x11d310
0x10e8c4:    cmp     ecx, 0x8000
0x10e8ca:    je      0x10eb88
0x10ead8:    mov     r12, rax
0x10eadb:    test    rax, rax
0x10eade:    jne     0x10e926
0x10e91a:    mov     r12, rax
0x10e91d:    test    rax, rax
0x10e920:    je      0x10eae4
0x10eb88:    xor     ecx, ecx
0x10eb8a:    and     edx, 0x49
0x10eb8d:    mov     byte ptr [rax + 0x48], 1
0x10eb91:    je      0x10eb9b
0x10e8d0:    mov     byte ptr [rax + 0x4a], 0
0x10e8d4:    cmp     ecx, 0x1000
0x10e8da:    je      0x10eb30
0x10eae4:    mov     r12, qword ptr [rbp + 0x38]
0x10eae8:    mov     edi, 0x20
0x10eaeD:    call    0x11d670
0x10e926:    mov     rdi, rax
0x10e929:    call    0x1049b0
0x10eb9b:    mov     byte ptr [rax + 0x4a], cl
0x10eb9e:    mov     rdx, qword ptr [rsp + 0x30]
0x10eba3:    jmp     0x10e911
0x10eb93:    mov     edx, dword ptr [rbp + 0x10]
0x10eb96:    test    edx, edx
0x10eb98:    sete    cl
0x10eb9b:    mov     byte ptr [rax + 0x4a], cl
0x10eb9e:    mov     rdx, qword ptr [rsp + 0x30]
0x10eba3:    jmp     0x10e911
0x10eb30:    mov     byte ptr [rax + 0x48], 3
0x10eb34:    mov     rdx, qword ptr [rsp + 0x30]
0x10eb39:    jmp     0x10e911
0x10e8e0:    cmp     ecx, 0xc000
0x10e8e6:    je      0x10ebb0
0x10eaf2:    mov     r8d, dword ptr [rsp + 0x1c]

```

```

0x10eaf7:    mov     esi, 1
0x10eafc:    lea     rcx, [rip + 0x12622]
0x10eb03:    add     r12, rbx
0x10eb06:    mov     rdx, -1
0x10eb0d:    mov     qword ptr [r12 + 0x28], rax
0x10eb12:    mov     rax, qword ptr [rbp + 0x38]
0x10eb16:    mov     rdi, qword ptr [rax + rbx + 0x28]
0x10eb1b:    xor     eax, eax
0x10eb1d:    call    0x104f10
0x10e92e:    mov     r13, qword ptr [rbp + 0x38]
0x10e932:    cmp     rax, 7
0x10e936:    lea     rdi, [rax + 1]
0x10e93a:    mov     eax, 9
0x10e93f:    cmovbe  rdi, rax
0x10e943:    add     r13, rbx
0x10e946:    call    0x11d670
0x10e911:    mov     qword ptr [rax + 8], rdx
0x10e915:    call    0x11d310
0x10ebb0:    mov     byte ptr [rax + 0x48], 4
0x10ebb4:    mov     rdx, qword ptr [rsp + 0x30]
0x10ebb9:    jmp     0x10e911
0x10e8ec:    cmp     ecx, 0xa000
0x10e8f2:    je      0x10ebc0
0x10eb22:    jmp     0x10e975
0x10e94b:    mov     r8, r12
0x10e94e:    mov     esi, 1
0x10e953:    lea     rcx, [rip + 0x127c6]
0x10e95a:    mov     qword ptr [r13 + 0x28], rax
0x10e95e:    mov     rax, qword ptr [rbp + 0x38]
0x10e962:    mov     rdx, -1
0x10e969:    mov     rdi, qword ptr [rax + rbx + 0x28]
0x10e96e:    xor     eax, eax
0x10e970:    call    0x104f10
0x10ebc0:    mov     rdi, qword ptr [rax]
0x10ebc3:    lea     rsi, [rsp + 0x90]
0x10ebcb:    call    0x118f40
0x10e8f8:    mov     byte ptr [rax + 0x48], 1
0x10e8fc:    mov     rdx, qword ptr [rsp + 0x30]
0x10e901:    jmp     0x10e911
0x10e975:    mov     edi, dword ptr [rsp + 0x20]
0x10e979:    call    0x11d4c0
0x10ebd0:    cmp     eax, -1
0x10ebd3:    je      0x10ec12
0x10e97e:    mov     r12, rax
0x10e981:    test    rax, rax
0x10e984:    je      0x10eb40
0x10ec12:    mov     rax, qword ptr [rbp + 0x38]
0x10ec16:    add     rax, rbx

```

```

0x10ec19:    mov     byte ptr [rax + 0x48], 2
0x10ec1d:    mov     rdi, qword ptr [rax]
0x10ec20:    call    0x118e80
0x10ebd5:    mov     ecx, dword ptr [rsp + 0xa8]
0x10ebdc:    mov     rax, qword ptr [rbp + 0x38]
0x10ebe0:    mov     edx, ecx
0x10ebe2:    add     rax, rbx
0x10ebe5:    and     edx, 0xf000
0x10ebeb:    cmp     edx, 0x4000
0x10ebf1:    mov     rdx, qword ptr [rsp + 0xc0]
0x10ebf9:    setne   byte ptr [rax + 0x48]
0x10ebfd:    and     ecx, 0x49
0x10ec00:    mov     qword ptr [rsp + 0x30], rdx
0x10ec05:    setne   byte ptr [rax + 0x4a]
0x10ec09:    mov     edi, dword ptr [rsp + 0x1c]
0x10ec0d:    jmp     0x10e911
0x10eb40:    mov     r12, qword ptr [rbp + 0x38]
0x10eb44:    mov     edi, 0x20
0x10eb49:    call    0x11d670
0x10e98a:    mov     rdi, rax
0x10e98d:    call    0x1049b0
0x10ec25:    xor     edx, edx
0x10ec27:    cmp     eax, -1
0x10ec2a:    cmovbe  eax, edx
0x10ec2d:    movsxd  rdx, eax
0x10ec30:    mov     rax, qword ptr [rbp + 0x38]
0x10ec34:    mov     qword ptr [rsp + 0x30], rdx
0x10ec39:    add     rax, rbx
0x10ec3c:    jmp     0x10ec09
0x10eb4e:    mov     r8d, dword ptr [rsp + 0x20]
0x10eb53:    mov     esi, 1
0x10eb58:    lea     rcx, [rip + 0x125c6]
0x10eb5f:    add     r12, rbx
0x10eb62:    mov     rdx, -1
0x10eb69:    mov     qword ptr [r12 + 0x30], rax
0x10eb6e:    mov     rax, qword ptr [rbp + 0x38]
0x10eb72:    mov     rdi, qword ptr [rax + rbx + 0x30]
0x10eb77:    xor     eax, eax
0x10eb79:    call    0x104f10
0x10e992:    mov     r13, qword ptr [rbp + 0x38]
0x10e996:    cmp     rax, 7
0x10e99a:    lea     rdi, [rax + 1]
0x10e99e:    mov     eax, 9
0x10e9a3:    cmovbe  rdi, rax
0x10e9a7:    add     r13, rbx
0x10e9aa:    call    0x11d670
0x10ec09:    mov     edi, dword ptr [rsp + 0x1c]
0x10ec0d:    jmp     0x10e911

```

```

0x10eb7e:    jmp     0x10e9d9
0x10e9af:    mov     r8, r12
0x10e9b2:    mov     esi, 1
0x10e9b7:    lea     rcx, [rip + 0x12762]
0x10e9be:    mov     qword ptr [r13 + 0x30], rax
0x10e9c2:    mov     rax, qword ptr [rbp + 0x38]
0x10e9c6:    mov     rdx, -1
0x10e9cd:    mov     rdi, qword ptr [rax + rbx + 0x30]
0x10e9d2:    xor     eax, eax
0x10e9d4:    call    0x104f10
0x10e9d9:    mov     rdx, qword ptr [rsp + 0x58]
0x10e9de:    mov     rax, qword ptr [rbp + 0x38]
0x10e9e2:    lea     rdi, [rsp + 0x58]
0x10e9e7:    mov     qword ptr [rax + rbx + 0x20], rdx
0x10e9ec:    call    0x104840
0x10e9f1:    mov     rcx, rax
0x10e9f4:    movsxd  rax, dword ptr [rax + 8]
0x10e9f8:    mov     r9d, dword ptr [rcx + 0xc]
0x10e9fc:    mov     rdx, rax
0x10e9ff:    imul    rax, rax, 0x2aaaaaab
0x10ea06:    mov     esi, edx
0x10ea08:    sar     esi, 0x1f
0x10ea0b:    sar     rax, 0x21
0x10ea0f:    sub     eax, esi
0x10ea11:    lea     esi, [rax + rax*2]
0x10ea14:    mov     eax, edx
0x10ea16:    shl     esi, 2
0x10ea19:    sub     eax, esi
0x10ea1b:    mov     esi, 0xc
0x10ea20:    cmovbe  eax, esi
0x10ea23:    cmp     edx, 0xc
0x10ea26:    mov     esi, 0x70
0x10ea2b:    mov     edx, 0x61
0x10ea30:    cmovge  edx, esi
0x10ea33:    mov     esi, dword ptr [rcx + 0x10]
0x10ea36:    add     rbx, qword ptr [rbp + 0x38]
0x10ea3a:    lea     rdi, [rbx + 0x38]
0x10ea3e:    push    rdx
0x10ea3f:    mov     edx, dword ptr [rcx + 4]
0x10ea42:    lea     r8d, [rsi + 1]
0x10ea46:    mov     esi, 1
0x10ea4b:    push    rdx
0x10ea4c:    push    rax
0x10ea4d:    movsxd  rax, dword ptr [rcx + 0x14]
0x10ea51:    mov     rdx, rax
0x10ea54:    imul    rax, rax, 0x51eb851f
0x10ea5b:    mov     ecx, edx
0x10ea5d:    sar     ecx, 0x1f

```

```

0x10ea60:    sar     rax, 0x25
0x10ea64:    sub     eax, ecx
0x10ea66:    lea     rcx, [rip + 0x126bd]
0x10ea6d:    imul    eax, eax, 0x64
0x10ea70:    sub     edx, eax
0x10ea72:    xor     eax, eax
0x10ea74:    push    rdx
0x10ea75:    mov     edx, 0x10
0x10ea7a:    call    0x104f10
0x10ea7f:    add     rsp, 0x20
0x10ea83:    mov     rax, qword ptr [rsp + 0x128]
0x10ea8b:    sub     rax, qword ptr fs:[0x28]
0x10ea94:    jne     0x10ec3e
0x10ea9a:    add     rsp, 0x138
0x10eaa1:    pop     rbx
0x10eaa2:    pop     rbp
0x10eaa3:    pop     r12
0x10eaa5:    pop     r13
0x10eaa7:    ret
0x10ec3e:    call    0x1049d0
0x10e903:    nop     dword ptr [rax + rax]
0x10eaa8:    nop     dword ptr [rax + rax]
0x10eb27:    nop     word ptr [rax + rax]
0x10eb3e:    nop
0x10eb83:    nop     dword ptr [rax + rax]
0x10eba8:    nop     dword ptr [rax + rax]
0x10ebbe:    nop
0x10ec43:    nop     word ptr cs:[rax + rax]
0x10ec4e:    nop
0x10ec50:    push    r15
0x10ec52:    push    r14
0x10ec54:    push    r13
0x10ec56:    push    r12
0x10ec58:    push    rbp
0x10ec59:    push    rbx
0x10ec5a:    mov     rbx, rdi
0x10ec5d:    sub     rsp, 0x6c8
0x10ec64:    mov     rax, qword ptr fs:[0x28]
0x10ec6d:    mov     qword ptr [rsp + 0x6b8], rax
0x10ec75:    xor     eax, eax
0x10ec77:    lea     rbp, [rsp + 0x1f]
0x10ec7c:    mov     rdi, rbp
0x10ec7f:    call    0x10a6a0
0x10ec84:    cmp     dword ptr [rbx + 0x60], 2
0x10ec88:    jg      0x10ef20
0x10ec8e:    mov     eax, dword ptr [rbx + 0x64]
0x10ec91:    cmp     eax, 0x27
0x10ec94:    jle     0x10eeb8

```

```

0x10ef20:    mov     esi, dword ptr [rip + 0x1b3fa]
0x10ef26:    xor     edi, edi
0x10ef28:    mov     edx, esi
0x10ef2a:    call    0x10a570
0x10eeb8:    cmp     eax, 0xd
0x10eebb:    jle     0x10ee46
0x10ec9a:    mov     esi, dword ptr [rbx + 0x70]
0x10ec9d:    test    esi, esi
0x10ec9f:    je      0x10ee38
0x10ef2f:    mov     eax, dword ptr [rbx + 0x64]
0x10ef32:    mov     rdi, qword ptr [rbx + 8]
0x10ef36:    lea     edx, [rax - 1]
0x10ef39:    mov     eax, dword ptr [rbx + 0x8c]
0x10ef3f:    lea     esi, [rax + 1]
0x10ef42:    call    0x10c680
0x10ee46:    mov     byte ptr [rsp + 0x2b0], 0
0x10ee4e:    lea     r13, [rsp + 0x2b0]
0x10ee56:    mov     edi, dword ptr [rip + 0x1b508]
0x10ee5c:    call    0x10a530
0x10eebd:    mov     esi, dword ptr [rbx + 0x28]
0x10eec0:    lea     r12, [rsp + 0x20]
0x10eec5:    mov     rdi, rbx
0x10eec8:    xor     eax, eax
0x10eeca:    mov     rdx, r12
0x10eecd:    lea     r13, [rsp + 0x2b0]
0x10eed5:    call    0x10e560
0x10ee38:    mov     eax, dword ptr [rip + 0x1c7e2]
0x10ee3e:    test    eax, eax
0x10ee40:    jne     0x10efc0
0x10eca5:    mov     edx, dword ptr [rbx + 0x68]
0x10eca8:    test    edx, edx
0x10ecaa:    jle     0x10f0e8
0x10ef47:    mov     rdi, qword ptr [rbx + 8]
0x10ef4b:    mov     esi, 0x20
0x10ef50:    call    0x10c650
0x10ee61:    mov     edi, dword ptr [rip + 0x1b4f9]
0x10ee67:    call    0x10a4f0
0x10eeda:    mov     r8, r12
0x10eedd:    mov     edx, 0x400
0x10eee2:    mov     rdi, r13
0x10eee5:    lea     rcx, [rip + 0x12289]
0x10eeec:    mov     esi, 1
0x10eef1:    xor     eax, eax
0x10eef3:    call    0x104f10
0x10efc0:    lea     r9, [rsp + 0x20]
0x10efc5:    mov     esi, dword ptr [rbx + 0x28]
0x10efc8:    xor     eax, eax
0x10efca:    mov     rdi, rbx

```

```

0x10efcd:    mov     rdx, r9
0x10efd0:    mov     qword ptr [rsp + 8], r9
0x10efd5:    lea     r13, [rsp + 0x2b0]
0x10efdd:    call    0x10e560
0x10f0e8:    xor     edi, edi
0x10f0ea:    jmp     0x10ece9
0x10ecb0:    sub     edx, 1
0x10ecb3:    mov     rcx, qword ptr [rbx + 0x38]
0x10ecb7:    xor     edi, edi
0x10ecb9:    lea     rdx, [rdx + rdx*4]
0x10ecbd:    shl     rdx, 4
0x10ecc1:    lea     rax, [rcx + 8]
0x10ecc5:    lea     rdx, [rcx + rdx + 0x58]
0x10ecca:    nop     word ptr [rax + rax]
0x10ecd0:    cmp     byte ptr [rax + 0x41], 0
0x10ecd4:    je      0x10ece0
0x10ef55:    mov     eax, dword ptr [rbx + 0x60]
0x10ef58:    mov     ecx, dword ptr [rbx + 0x68]
0x10ef5b:    lea     edx, [rax - 1]
0x10ef5e:    cmp     edx, ecx
0x10ef60:    jg      0x10ec8e
0x10ee6c:    jmp     0x10ed77
0x10eef8:    jmp     0x10ee56
0x10efe2:    mov     eax, dword ptr [rbx + 0x64]
0x10efe5:    lea     r12d, [rax - 0x1a]
0x10efe9:    movsxd  rax, dword ptr [rbx + 0x28]
0x10efed:    movsxd  r12, r12d
0x10eff0:    lea     r15, [rax + rax*4]
0x10eff4:    shl     r15, 4
0x10eff8:    add     r15, qword ptr [rbx + 0x38]
0x10effc:    mov     rsi, qword ptr [r15]
0x10efff:    mov     rdi, rsi
0x10f002:    mov     qword ptr [rsp], rsi
0x10f006:    call    0x1049b0
0x10ece9:    mov     r8d, 1
0x10ecef:    lea     rsi, [rsp + 0x20]
0x10ecf4:    mov     ecx, 1
0x10ecf9:    mov     edx, 0x19c
0x10ecfe:    call    0x11c520
0x10ece0:    add     rax, 0x50
0x10ece4:    cmp     rax, rdx
0x10ece7:    jne     0x10ecd0
0x10ecd6:    cmp     byte ptr [rax + 0x40], 1
0x10ecda:    jne     0x10ece0
0x10ef66:    sub     eax, 2
0x10ef69:    imul    eax, dword ptr [rbx + 0x28]
0x10ef6d:    mov     esi, dword ptr [rip + 0x1b3b5]
0x10ef73:    mov     edi, dword ptr [rip + 0x1b3b7]

```



```

0x10ef79:      cdq
0x10ef7a:      idiv     ecx
0x10ef7c:      mov     edx, dword ptr [rip + 0x1b3aa]
0x10ef82:      mov     dword ptr [rbx + 0x8c], eax
0x10ef88:      call    0x10a570
0x10ed77:      mov     rdi, r13
0x10ed7a:      call    0x1049b0
0x10ee56:      mov     edi, dword ptr [rip + 0x1b508]
0x10ee5c:      call    0x10a530
0x10f00b:      mov     rsi, qword ptr [rsp]
0x10f00f:      mov     ecx, 0x400
0x10f014:      mov     rdi, r13
0x10f017:      cmp     r12, rax
0x10f01a:      cmovbe   rax, r12
0x10f01e:      mov     rdx, rax
0x10f021:      mov     r14, rax
0x10f024:      call    0x104b70
0x10ed03:      cmp     byte ptr [rax], 0x20
0x10ed06:      mov     r8, rax
0x10ed09:      jne     0x10ed1d
0x10ecd0:      cmp     byte ptr [rax + 0x41], 0
0x10ecd4:      je      0x10ece0
0x10ecdc:      add     rdi, qword ptr [rax]
0x10ecdf:      nop
0x10ece0:      add     rax, 0x50
0x10ece4:      cmp     rax, rdx
0x10ece7:      jne     0x10ecd0
0x10ef8d:      mov     eax, dword ptr [rbx + 0x64]
0x10ef90:      mov     rdi, qword ptr [rbx + 8]
0x10ef94:      lea     edx, [rax - 1]
0x10ef97:      mov     eax, dword ptr [rbx + 0x8c]
0x10ef9d:      lea     esi, [rax + 1]
0x10efa0:      call    0x10c680
0x10ed7f:      mov     r8, qword ptr [rbx + 0x20]
0x10ed83:      mov     r12, rax
0x10ed86:      cmp     eax, 8
0x10ed89:      jae     0x10ee78
0x10f029:      mov     rdx, r12
0x10f02c:      lea     rdi, [r13 + r14]
0x10f031:      mov     esi, 0x20
0x10f036:      sub     rdx, r14
0x10f039:      call    0x104a50
0x10ed0b:      lea     rdx, [rax + 1]
0x10ed0f:      nop
0x10ed10:      mov     r8, rdx
0x10ed13:      add     rdx, 1
0x10ed17:      cmp     byte ptr [rdx - 1], 0x20
0x10ed1b:      je      0x10ed10

```

```

0x10ed1d:    mov     r9d, dword ptr [rbx + 0x70]
0x10ed21:    lea     rdx, [rip + 0x136bd]
0x10ed28:    lea     rax, [rip + 0x126ec]
0x10ed2f:    mov     esi, 1
0x10ed34:    lea     r13, [rsp + 0x2b0]
0x10ed3c:    lea     rcx, [rip + 0x12410]
0x10ed43:    cmp     r9d, 1
0x10ed47:    mov     rdi, r13
0x10ed4a:    cmovle  rax, rdx
0x10ed4e:    sub     rsp, 8
0x10ed52:    mov     edx, 0x400
0x10ed57:    push    rax
0x10ed58:    xor     eax, eax
0x10ed5a:    call    0x104f10
0x10efa5:    mov     rdi, qword ptr [rbx + 8]
0x10efa9:    mov     esi, 0x23
0x10efae:    call    0x10c650
0x10ed8f:    test    al, 4
0x10ed91:    jne     0x10f0b0
0x10ee78:    mov     rax, qword ptr [rsp + 0x2b0]
0x10ee80:    lea     rdi, [r8 + 8]
0x10ee84:    mov     rsi, r13
0x10ee87:    and     rdi, 0xfffffffffffff8
0x10ee8b:    mov     qword ptr [r8], rax
0x10ee8e:    mov     eax, r12d
0x10ee91:    mov     rdx, qword ptr [r13 + rax - 8]
0x10ee96:    mov     qword ptr [r8 + rax - 8], rdx
0x10ee9b:    sub     r8, rdi
0x10ee9e:    sub     rsi, r8
0x10eea1:    add     r8d, r12d
0x10eea4:    shr     r8d, 3
0x10eea8:    mov     ecx, r8d
0x10eeab:    rep movsq      qword ptr [rdi], qword ptr [rsi]
0x10f03e:    cmp     byte ptr [r15 + 0x48], 0
0x10f043:    mov     r9, qword ptr [rsp + 8]
0x10f048:    jne     0x10f088
0x10ed10:    mov     r8, rdx
0x10ed13:    add     rdx, 1
0x10ed17:    cmp     byte ptr [rdx - 1], 0x20
0x10ed1b:    je      0x10ed10
0x10ed5f:    mov     edi, dword ptr [rip + 0x1b607]
0x10ed65:    call    0x10a530
0x10efb3:    jmp     0x10ec8e
0x10ed97:    test    eax, eax
0x10ed99:    je      0x10edb4
0x10f0b0:    mov     eax, dword ptr [rsp + 0x2b0]
0x10f0b7:    mov     dword ptr [r8], eax
0x10f0ba:    mov     eax, r12d

```

```

0x10f0bd:    mov     edx, dword ptr [r13 + rax - 4]
0x10f0c2:    mov     dword ptr [r8 + rax - 4], edx
0x10f0c7:    mov     r8, qword ptr [rbx + 0x20]
0x10f0cb:    jmp     0x10edb4
0x10eeae:    jmp     0x10edb0
0x10eeab:    rep movsq      qword ptr [rdi], qword ptr [rsi]
0x10f04a:    mov     rax, qword ptr [r15]
0x10f04d:    cmp     byte ptr [rax], 0x2e
0x10f050:    jne     0x10f058
0x10f088:    mov     r8, qword ptr [r15 + 8]
0x10f08c:    lea     rdi, [r13 + r12]
0x10f091:    mov     esi, 1
0x10f096:    xor     eax, eax
0x10f098:    lea     rcx, [rip + 0x120cf]
0x10f09f:    mov     rdx, -1
0x10f0a6:    call    0x104f10
0x10ed6a:    mov     edi, dword ptr [rip + 0x1b5f8]
0x10ed70:    call    0x10a4f0
0x10edb4:    movsxd  rdx, dword ptr [rbx + 0x64]
0x10edb8:    lea     rcx, [r12 + 2]
0x10edbd:    cmp     rcx, rdx
0x10edc0:    jb      0x10ef00
0x10ed9b:    movzx   eax, byte ptr [rsp + 0x2b0]
0x10eda3:    mov     byte ptr [r8], al
0x10eda6:    test    r12b, 2
0x10edaa:    jne     0x10f0ef
0x10edb0:    mov     r8, qword ptr [rbx + 0x20]
0x10edb4:    movsxd  rdx, dword ptr [rbx + 0x64]
0x10edb8:    lea     rcx, [r12 + 2]
0x10edbd:    cmp     rcx, rdx
0x10edc0:    jb      0x10ef00
0x10f052:    cmp     byte ptr [rax + 1], 0x2e
0x10f056:    je      0x10f0d0
0x10f058:    lea     r8, [rip + 0x120ec]
0x10f05f:    lea     rdi, [r13 + r12]
0x10f064:    lea     rcx, [rip + 0x120f8]
0x10f06b:    mov     rdx, -1
0x10f072:    xor     eax, eax
0x10f074:    mov     esi, 1
0x10f079:    call    0x104f10
0x10f0ab:    jmp     0x10ee56
0x10ed75:    pop     rdx
0x10ed76:    pop     rcx
0x10ed77:    mov     rdi, r13
0x10ed7a:    call    0x1049b0
0x10ef00:    lea     edx, [rdx - 2]
0x10ef03:    lea     rdi, [r8 + r12]
0x10ef07:    mov     esi, 0x20

```

```

0x10ef0c:      movsxd  rdx, edx
0x10ef0f:      sub    rdx, r12
0x10ef12:      call   0x104a50
0x10edc6:      mov    rdi, r8
0x10edc9:      mov    rsi, r12
0x10edcc:      call   0x1184d0
0x10f0ef:      mov    eax, r12d
0x10f0f2:      movzx   edx, word ptr [r13 + rax - 2]
0x10f0f8:      mov     word ptr [r8 + rax - 2], dx
0x10f0fe:      mov     r8, qword ptr [rbx + 0x20]
0x10f102:      jmp     0x10edb4
0x10f0d0:      cmp     byte ptr [rax + 2], 0
0x10f0d4:      lea     r8, [rip + 0x12068]
0x10f0db:      je      0x10f05f
0x10f07e:      jmp     0x10ee56
0x10ef17:      mov     r8, qword ptr [rbx + 0x20]
0x10ef1b:      jmp     0x10edc6
0x10edd1:      mov     edi, dword ptr [rip + 0x1b549]
0x10edd7:      call    0x10a510
0x10f05f:      lea     rdi, [r13 + r12]
0x10f064:      lea     rcx, [rip + 0x120f8]
0x10f06b:      mov     rdx, -1
0x10f072:      xor     eax, eax
0x10f074:      mov     esi, 1
0x10f079:      call    0x104f10
0x10f0dd:      jmp     0x10f058
0x10eddc:      mov     eax, dword ptr [rbx + 0x60]
0x10eddf:      mov     rdi, qword ptr [rbx + 8]
0x10ede3:      mov     edx, 2
0x10ede8:      lea     esi, [rax - 1]
0x10edeb:      call    0x10c680
0x10edf0:      mov     eax, dword ptr [rbx + 0x64]
0x10edf3:      mov     rdi, qword ptr [rbx + 8]
0x10edf7:      mov     rsi, qword ptr [rbx + 0x20]
0x10edfb:      lea     edx, [rax - 4]
0x10edfe:      call    0x10c600
0x10ee03:      mov     rdi, rbp
0x10ee06:      call    0x10a6b0
0x10ee0b:      mov     rax, qword ptr [rsp + 0x6b8]
0x10ee13:      sub     rax, qword ptr fs:[0x28]
0x10ee1c:      jne     0x10f107
0x10ee22:      add     rsp, 0x6c8
0x10ee29:      pop     rbx
0x10ee2a:      pop     rbp
0x10ee2b:      pop     r12
0x10ee2d:      pop     r13
0x10ee2f:      pop     r14
0x10ee31:      pop     r15

```

```

0x10ee33:      ret
0x10f107:      call    0x1049d0
0x10ee34:      nop     dword ptr [rax]
0x10ee71:      nop     dword ptr [rax]
0x10eeb3:      nop     dword ptr [rax + rax]
0x10eefd:      nop     dword ptr [rax]
0x10efb8:      nop     dword ptr [rax + rax]
0x10f083:      nop     dword ptr [rax + rax]
0x10f0e2:      nop     word ptr [rax + rax]
0x10f10c:      nop     dword ptr [rax]
0x10f110:      mov     eax, dword ptr [rdi + 0x2c]
0x10f113:      mov     edx, dword ptr [rdi + 0x28]
0x10f116:      mov     ecx, dword ptr [rdi + 0x60]
0x10f119:      mov     esi, dword ptr [rdi + 0x68]
0x10f11c:      cmp     eax, edx
0x10f11e:      jg      0x10f130
0x10f120:      lea     edi, [rcx + rax - 2]
0x10f124:      cmp     edx, edi
0x10f126:      jge     0x10f130
0x10f130:      sub     ecx, 2
0x10f133:      mov     eax, ecx
0x10f135:      shr     eax, 0x1f
0x10f138:      add     eax, ecx
0x10f13a:      sar     eax, 1
0x10f13c:      sub     edx, eax
0x10f13e:      mov     eax, 0
0x10f143:      lea     edi, [rcx + rdx]
0x10f146:      cmovns  eax, edx
0x10f149:      cmp     esi, edi
0x10f14b:      jg      0x10f159
0x10f128:      cmp     edi, esi
0x10f12a:      jle     0x10f159
0x10f14d:      mov     eax, esi
0x10f14f:      mov     edx, 0
0x10f154:      sub     eax, ecx
0x10f156:      cmovs   eax, edx
0x10f159:      ret
0x10f159:      ret
0x10f12c:      nop     dword ptr [rax]
0x10f15a:      nop     word ptr [rax + rax]
0x10f160:      endbr64
0x10f164:      push    r12
0x10f166:      push    rbp
0x10f167:      mov     rbp, rdi
0x10f16a:      mov     edi, 0x98
0x10f16f:      sub     rsp, 8
0x10f173:      call    0x11d670
0x10f178:      mov     ecx, 0x100

```

```

0x10f17d:    mov     rdi, rbp
0x10f180:    mov     dword ptr [rax + 0x50], 0xffffffff
0x10f187:    mov     r12, rax
0x10f18a:    mov     qword ptr [rax + 0x20], 0
0x10f192:    mov     qword ptr [rax + 0x28], 0
0x10f19a:    mov     qword ptr [rax], 0
0x10f1a1:    mov     qword ptr [rax + 0x48], 0
0x10f1a9:    mov     qword ptr [rax + 0x58], 0
0x10f1b1:    movabs  rax, 0x100000001
0x10f1bb:    mov     qword ptr [r12 + 0x60], rax
0x10f1c0:    movabs  rax, 0x3fffffff00000000
0x10f1ca:    mov     dword ptr [r12 + 0x68], 0
0x10f1d3:    mov     word ptr [r12 + 0x6c], cx
0x10f1d9:    mov     qword ptr [r12 + 0x80], 0
0x10f1e5:    mov     qword ptr [r12 + 0x88], 1
0x10f1f1:    mov     qword ptr [r12 + 0x70], 0
0x10f1fa:    mov     qword ptr [r12 + 0x10], rax
0x10f1ff:    mov     qword ptr [r12 + 0x38], 0
0x10f208:    call    0x1049c0
0x10f20d:    cmp     eax, -1
0x10f210:    je      0x10f3b0
0x10f3b0:    lea     rdi, [rip + 0x12641]
0x10f3b7:    call    0x108b00
0x10f216:    xor     eax, eax
0x10f218:    call    0x118fe0
0x10f21d:    mov     qword ptr [r12 + 0x18], rax
0x10f222:    mov     rdi, rax
0x10f225:    test    rax, rax
0x10f228:    je      0x10f3c8
0x10f3c8:    lea     rdi, [rip + 0x11581]
0x10f3cf:    call    0x108b00
0x10f22e:    call    0x118040
0x10f233:    mov     rdi, qword ptr [r12 + 0x18]
0x10f238:    call    0x1049b0
0x10f23d:    mov     qword ptr [r12 + 0x78], rax
0x10f242:    xor     eax, eax
0x10f244:    call    0x10c590
0x10f249:    mov     edx, dword ptr [rip + 0x1c3b1]
0x10f24f:    mov     qword ptr [r12 + 8], rax
0x10f254:    test    edx, edx
0x10f256:    je      0x10f280
0x10f280:    lea     rdi, [rip + 0x11406]
0x10f287:    call    0x117c50
0x10f258:    mov     eax, dword ptr [rip + 0x1c3d6]
0x10f25e:    mov     dword ptr [r12 + 0x80], eax
0x10f266:    mov     eax, dword ptr [rip + 0x1c3c4]
0x10f26c:    mov     dword ptr [r12 + 0x84], eax
0x10f274:    add     rsp, 8

```

```

0x10f278:    mov     rax, r12
0x10f27b:    pop     rbp
0x10f27c:    pop     r12
0x10f27e:    ret
0x10f28c:    xor     esi, esi
0x10f28e:    lea     rdi, [rip + 0x11ee5]
0x10f295:    call    0x117d10
0x10f29a:    lea     rdi, [rip + 0x11411]
0x10f2a1:    mov     dword ptr [rip + 0x1c385], eax
0x10f2a7:    call    0x117c50
0x10f2ac:    xor     ecx, ecx
0x10f2ae:    mov     edx, 6
0x10f2b3:    lea     rsi, [rip + 0x1b326]
0x10f2ba:    lea     rdi, [rip + 0x11ecb]
0x10f2c1:    call    0x117d70
0x10f2c6:    xor     ecx, ecx
0x10f2c8:    mov     edx, 9
0x10f2cd:    lea     rsi, [rip + 0x1b2ac]
0x10f2d4:    mov     dword ptr [r12 + 0x80], eax
0x10f2dc:    lea     rdi, [rip + 0x11ec0]
0x10f2e3:    mov     dword ptr [rip + 0x1c34b], eax
0x10f2e9:    call    0x117d70
0x10f2ee:    mov     esi, 1
0x10f2f3:    lea     rdi, [rip + 0x11ebf]
0x10f2fa:    mov     dword ptr [r12 + 0x84], eax
0x10f302:    mov     dword ptr [rip + 0x1c328], eax
0x10f308:    call    0x117e50
0x10f30d:    mov     esi, 1
0x10f312:    lea     rdi, [rip + 0x11eac]
0x10f319:    mov     dword ptr [rip + 0x1c301], eax
0x10f31f:    call    0x117e50
0x10f324:    lea     rdi, [rip + 0x1a91d]
0x10f32b:    mov     dword ptr [rip + 0x1c2f3], eax
0x10f331:    mov     eax, dword ptr [rip + 0x1a929]
0x10f337:    test    eax, eax
0x10f339:    lea     rax, [rip + 0x1a8f0]
0x10f340:    cmov     rdi, rax
0x10f344:    call    0x117c50
0x10f349:    mov     edx, 0x14
0x10f34e:    lea     rsi, [rip + 0x1b18b]
0x10f355:    lea     rdi, [rip + 0x1afc4]
0x10f35c:    call    0x117f70
0x10f361:    mov     edi, 8
0x10f366:    call    0x119100
0x10f36b:    mov     edi, 8
0x10f370:    mov     qword ptr [rip + 0x1c2a1], rax
0x10f377:    call    0x119100
0x10f37c:    mov     edi, 8

```

```

0x10f381:    mov     qword ptr [rip + 0x1c288], rax
0x10f388:    call    0x119100
0x10f38d:    mov     dword ptr [rip + 0x1c269], 1
0x10f397:    mov     qword ptr [rip + 0x1c26a], rax
0x10f39e:    add     rsp, 8
0x10f3a2:    mov     rax, r12
0x10f3a5:    pop     rbp
0x10f3a6:    pop     r12
0x10f3a8:    ret
0x10f27f:    nop
0x10f3a9:    nop     dword ptr [rax]
0x10f3bc:    jmp     0x10f216
0x10f3c1:    nop     dword ptr [rax]
0x10f3d4:    mov     rdi, qword ptr [r12 + 0x18]
0x10f3d9:    jmp     0x10f22e
0x10f3de:    nop
0x10f3e0:    endbr64
0x10f3e4:    push    r12
0x10f3e6:    mov     r12, rdi
0x10f3e9:    push    rbp
0x10f3ea:    push    rbx
0x10f3eb:    mov     rdi, qword ptr [rdi]
0x10f3ee:    test    rdi, rdi
0x10f3f1:    je      0x10f3f8
0x10f3f8:    mov     eax, dword ptr [r12 + 0x68]
0x10f3fd:    mov     r8, qword ptr [r12 + 0x38]
0x10f402:    test    eax, eax
0x10f404:    jle     0x10f43e
0x10f3f3:    call    0x104ac0
0x10f43e:    mov     rdi, r8
0x10f441:    call    0x118d10
0x10f406:    xor     ebx, ebx
0x10f408:    nop     dword ptr [rax + rax]
0x10f410:    lea     rbp, [rbx + rbx*4]
0x10f414:    shl     rbp, 4
0x10f418:    mov     rdi, qword ptr [r8 + rbp]
0x10f41c:    test    rdi, rdi
0x10f41f:    je      0x10f433
0x10f446:    mov     rdi, qword ptr [r12 + 0x20]
0x10f44b:    call    0x118d10
0x10f433:    add     rbx, 1
0x10f437:    cmp     dword ptr [r12 + 0x68], ebx
0x10f43c:    jg      0x10f410
0x10f421:    call    0x118d10
0x10f450:    mov     rdi, qword ptr [r12 + 8]
0x10f455:    call    0x10c5c0
0x10f410:    lea     rbp, [rbx + rbx*4]
0x10f414:    shl     rbp, 4

```



```

0x10f418:    mov     rdi, qword ptr [r8 + rbp]
0x10f41c:    test    rdi, rdi
0x10f41f:    je      0x10f433
0x10f426:    mov     r8, qword ptr [r12 + 0x38]
0x10f42b:    mov     qword ptr [r8 + rbp], 0
0x10f433:    add     rbx, 1
0x10f437:    cmp     dword ptr [r12 + 0x68], ebx
0x10f43c:    jg      0x10f410
0x10f45a:    pop     rbx
0x10f45b:    mov     rdi, r12
0x10f45e:    pop     rbp
0x10f45f:    pop     r12
0x10f461:    jmp     0x118d10
0x10f466:    nop     word ptr cs:[rax + rax]
0x10f470:    endbr64
0x10f474:    mov     rax, qword ptr [rdi + 8]
0x10f478:    ret
0x10f479:    nop     dword ptr [rax]
0x10f480:    endbr64
0x10f484:    mov     dword ptr [rdi + 0x14], 0x3fffffff
0x10f48b:    ret
0x10f48c:    nop     dword ptr [rax]
0x10f490:    endbr64
0x10f494:    movsxd  rax, dword ptr [rdi + 0x28]
0x10f498:    lea     rax, [rax + rax*4]
0x10f49c:    shl     rax, 4
0x10f4a0:    add     rax, qword ptr [rdi + 0x38]
0x10f4a4:    mov     rax, qword ptr [rax]
0x10f4a7:    ret
0x10f4a8:    nop     dword ptr [rax + rax]
0x10f4b0:    endbr64
0x10f4b4:    movsxd  rax, dword ptr [rdi + 0x28]
0x10f4b8:    lea     rax, [rax + rax*4]
0x10f4bc:    shl     rax, 4
0x10f4c0:    add     rax, qword ptr [rdi + 0x38]
0x10f4c4:    mov     eax, dword ptr [rax + 0x14]
0x10f4c7:    ret
0x10f4c8:    nop     dword ptr [rax + rax]
0x10f4d0:    endbr64
0x10f4d4:    movsxd  rax, dword ptr [rdi + 0x28]
0x10f4d8:    lea     rax, [rax + rax*4]
0x10f4dc:    shl     rax, 4
0x10f4e0:    add     rax, qword ptr [rdi + 0x38]
0x10f4e4:    mov     eax, dword ptr [rax + 0x18]
0x10f4e7:    ret
0x10f4e8:    nop     dword ptr [rax + rax]
0x10f4f0:    endbr64
0x10f4f4:    movsxd  rax, dword ptr [rdi + 0x28]

```

```

0x10f4f8:    lea     rax, [rax + rax*4]
0x10f4fc:    shl     rax, 4
0x10f500:    add     rax, qword ptr [rdi + 0x38]
0x10f504:    mov     eax, dword ptr [rax + 0x10]
0x10f507:    ret
0x10f508:    nop     dword ptr [rax + rax]
0x10f510:    endbr64
0x10f514:    movsxd  rax, dword ptr [rdi + 0x28]
0x10f518:    lea     rax, [rax + rax*4]
0x10f51c:    shl     rax, 4
0x10f520:    add     rax, qword ptr [rdi + 0x38]
0x10f524:    movsx   eax, byte ptr [rax + 0x48]
0x10f528:    ret
0x10f529:    nop     dword ptr [rax]
0x10f530:    endbr64
0x10f534:    mov     eax, dword ptr [rdi + 0x60]
0x10f537:    mov     esi, dword ptr [rdi + 0x68]
0x10f53a:    mov     r8d, 0
0x10f540:    lea     edx, [rax - 2]
0x10f543:    mov     eax, dword ptr [rdi + 0x28]
0x10f546:    mov     ecx, edx
0x10f548:    shr     ecx, 0x1f
0x10f54b:    add     ecx, edx
0x10f54d:    sar     ecx, 1
0x10f54f:    sub     eax, ecx
0x10f551:    lea     ecx, [rdx + rax]
0x10f554:    cmovs   eax, r8d
0x10f558:    cmp     esi, ecx
0x10f55a:    jg      0x10f568
0x10f55c:    mov     eax, esi
0x10f55e:    sub     eax, edx
0x10f560:    mov     edx, 0
0x10f565:    cmovs   eax, edx
0x10f568:    mov     dword ptr [rdi + 0x2c], eax
0x10f56b:    ret
0x10f568:    mov     dword ptr [rdi + 0x2c], eax
0x10f56b:    ret
0x10f56c:    nop     dword ptr [rax]
0x10f570:    endbr64
0x10f574:    mov     byte ptr [rdi + 0x6d], 1
0x10f578:    ret
0x10f579:    nop     dword ptr [rax]
0x10f580:    endbr64
0x10f584:    mov     byte ptr [rdi + 0x6d], 0
0x10f588:    ret
0x10f589:    nop     dword ptr [rax]
0x10f590:    endbr64
0x10f594:    mov     eax, dword ptr [rip + 0x1c092]

```

```

0x10f59a:    push    r14
0x10f59c:    mov     r14d, ecx
0x10f59f:    push    r13
0x10f5a1:    mov     r13d, edx
0x10f5a4:    push    r12
0x10f5a6:    mov     r12d, esi
0x10f5a9:    push    rbp
0x10f5aa:    mov     rbp, r8
0x10f5ad:    push    rbx
0x10f5ae:    mov     rbx, rdi
0x10f5b1:    mov     dword ptr [rdi + 0x58], esi
0x10f5b4:    mov     dword ptr [rdi + 0x5c], edx
0x10f5b7:    mov     dword ptr [rdi + 0x60], ecx
0x10f5ba:    mov     dword ptr [rdi + 0x64], r8d
0x10f5be:    mov     dword ptr [rdi + 0x8c], 0
0x10f5c8:    test    eax, eax
0x10f5ca:    jle     0x10f648
0x10f648:    mov     eax, ecx
0x10f64a:    shr     eax, 0x1f
0x10f64d:    add     eax, ecx
0x10f64f:    sar     eax, 1
0x10f651:    jmp     0x10f5d3
0x10f5cc:    lea     edx, [rcx - 1]
0x10f5cf:    cmp     eax, edx
0x10f5d1:    jge     0x10f648
0x10f5d3:    mov     dword ptr [rbx + 0x88], eax
0x10f5d9:    mov     rdi, qword ptr [rbx + 0x20]
0x10f5dd:    movsxd  rsi, ebp
0x10f5e0:    call    0x11d690
0x10f5e5:    mov     rdi, qword ptr [rbx + 8]
0x10f5e9:    mov     ecx, r14d
0x10f5ec:    mov     edx, r13d
0x10f5ef:    mov     qword ptr [rbx + 0x20], rax
0x10f5f3:    mov     esi, r12d
0x10f5f6:    mov     r8d, ebp
0x10f5f9:    call    0x10c5e0
0x10f5fe:    mov     eax, dword ptr [rbx + 0x60]
0x10f601:    mov     esi, dword ptr [rbx + 0x68]
0x10f604:    mov     edi, 0
0x10f609:    lea     edx, [rax - 2]
0x10f60c:    mov     eax, dword ptr [rbx + 0x28]
0x10f60f:    mov     ecx, edx
0x10f611:    shr     ecx, 0x1f
0x10f614:    add     ecx, edx
0x10f616:    sar     ecx, 1
0x10f618:    sub     eax, ecx
0x10f61a:    lea     ecx, [rdx + rax]
0x10f61d:    cmovs   eax, edi

```

```

0x10f620:    cmp     esi, ecx
0x10f622:    jg      0x10f630
0x10f624:    mov     eax, esi
0x10f626:    sub     eax, edx
0x10f628:    mov     edx, 0
0x10f62d:    cmovs   eax, edx
0x10f630:    mov     dword ptr [rbx + 0x2c], eax
0x10f633:    mov     dword ptr [rbx + 0x74], 0
0x10f63a:    pop     rbx
0x10f63b:    pop     rbp
0x10f63c:    pop     r12
0x10f63e:    pop     r13
0x10f640:    pop     r14
0x10f642:    ret
0x10f630:    mov     dword ptr [rbx + 0x2c], eax
0x10f633:    mov     dword ptr [rbx + 0x74], 0
0x10f63a:    pop     rbx
0x10f63b:    pop     rbp
0x10f63c:    pop     r12
0x10f63e:    pop     r13
0x10f640:    pop     r14
0x10f642:    ret
0x10f643:    nop     dword ptr [rax + rax]
0x10f653:    nop     word ptr cs:[rax + rax]
0x10f65e:    nop
0x10f660:    endbr64
0x10f664:    mov     byte ptr [rdi + 0x6e], sil
0x10f668:    ret
0x10f669:    nop     dword ptr [rax]
0x10f670:    endbr64
0x10f674:    push    r13
0x10f676:    push    r12
0x10f678:    push    rbp
0x10f679:    push    rbx
0x10f67a:    sub     rsp, 8
0x10f67e:    test    ecx, ecx
0x10f680:    js      0x10f6ce
0x10f6ce:    mov     r12d, 0xffffffff
0x10f6d4:    jmp     0x10f6c0
0x10f682:    movsxd  rax, ecx
0x10f685:    mov     rbp, rsi
0x10f688:    mov     r13, rdx
0x10f68b:    mov     r12d, ecx
0x10f68e:    lea     rbx, [rax + rax*4]
0x10f692:    shl     rbx, 4
0x10f696:    add     rbx, qword ptr [rdi + 0x38]
0x10f69a:    jmp     0x10f6ae
0x10f6c0:    add     rsp, 8

```

```

0x10f6c4:    mov     eax, r12d
0x10f6c7:    pop     rbx
0x10f6c8:    pop     rbp
0x10f6c9:    pop     r12
0x10f6cb:    pop     r13
0x10f6cd:    ret
0x10f6ae:    mov     rsi, qword ptr [rbx]
0x10f6b1:    mov     rdx, r13
0x10f6b4:    mov     rdi, rbp
0x10f6b7:    call    0x104c30
0x10f6bc:    test    eax, eax
0x10f6be:    jne     0x10f6a0
0x10f6a0:    sub     r12d, 1
0x10f6a4:    sub     rbx, 0x50
0x10f6a8:    cmp     r12d, -1
0x10f6ac:    je      0x10f6c0
0x10f69c:    nop     dword ptr [rax]
0x10f6d6:    nop     word ptr cs:[rax + rax]
0x10f6e0:    endbr64
0x10f6e4:    push    r14
0x10f6e6:    push    r13
0x10f6e8:    push    r12
0x10f6ea:    push    rbp
0x10f6eb:    push    rbx
0x10f6ec:    mov     r14d, dword ptr [rdi + 0x68]
0x10f6f0:    cmp     ecx, r14d
0x10f6f3:    jge     0x10f740
0x10f6f5:    movsxd  rax, ecx
0x10f6f8:    mov     rbp, rsi
0x10f6fb:    mov     r13, rdx
0x10f6fe:    mov     r12d, ecx
0x10f701:    lea     rbx, [rax + rax*4]
0x10f705:    shl     rbx, 4
0x10f709:    add     rbx, qword ptr [rdi + 0x38]
0x10f70d:    jmp     0x10f71d
0x10f740:    mov     r12d, 0xffffffff
0x10f746:    pop     rbx
0x10f747:    pop     rbp
0x10f748:    mov     eax, r12d
0x10f74b:    pop     r12
0x10f74d:    pop     r13
0x10f74f:    pop     r14
0x10f751:    ret
0x10f71d:    mov     rsi, qword ptr [rbx]
0x10f720:    mov     rdx, r13
0x10f723:    mov     rdi, rbp
0x10f726:    call    0x104c30
0x10f72b:    test    eax, eax

```

```

0x10f72d:    jne     0x10f710
0x10f72f:    pop     rbx
0x10f730:    mov     eax, r12d
0x10f733:    pop     rbp
0x10f734:    pop     r12
0x10f736:    pop     r13
0x10f738:    pop     r14
0x10f73a:    ret
0x10f710:    add     r12d, 1
0x10f714:    add     rbx, 0x50
0x10f718:    cmp     r12d, r14d
0x10f71b:    je      0x10f740
0x10f70f:    nop
0x10f73b:    nop     dword ptr [rax + rax]
0x10f752:    nop     word ptr cs:[rax + rax]
0x10f75d:    nop     dword ptr [rax]
0x10f760:    endbr64
0x10f764:    push    r14
0x10f766:    mov     r14d, ecx
0x10f769:    push    r13
0x10f76b:    mov     r13, rdx
0x10f76e:    push    r12
0x10f770:    mov     r12, rdi
0x10f773:    push    rbp
0x10f774:    push    rbx
0x10f775:    mov     rbx, rsi
0x10f778:    call    0x1049b0
0x10f77d:    mov     rdi, rbx
0x10f780:    mov     rbp, rax
0x10f783:    call    0x1049b0
0x10f788:    lea     rdi, [rbp + rax + 1]
0x10f78d:    call    0x11d670
0x10f792:    mov     r8, rbx
0x10f795:    mov     rcx, r12
0x10f798:    mov     esi, 1
0x10f79d:    mov     rbp, rax
0x10f7a0:    mov     rdi, rax
0x10f7a3:    mov     rdx, -1
0x10f7aa:    xor     eax, eax
0x10f7ac:    call    0x104f10
0x10f7b1:    mov     edx, r14d
0x10f7b4:    mov     rsi, r13
0x10f7b7:    mov     rdi, rbp
0x10f7ba:    call    0x108070
0x10f7bf:    mov     rdi, rbp
0x10f7c2:    mov     r12d, eax
0x10f7c5:    call    0x118d10
0x10f7ca:    pop     rbx

```

```

0x10f7cb:    mov     eax, r12d
0x10f7ce:    pop     rbp
0x10f7cf:    pop     r12
0x10f7d1:    pop     r13
0x10f7d3:    pop     r14
0x10f7d5:    ret
0x10f7d6:    nop     word ptr cs:[rax + rax]
0x10f7e0:    endbr64
0x10f7e4:    push    r15
0x10f7e6:    mov     r15d, r8d
0x10f7e9:    push    r14
0x10f7eb:    mov     r14, rcx
0x10f7ee:    push    r13
0x10f7f0:    mov     r13, rdx
0x10f7f3:    push    r12
0x10f7f5:    mov     r12, rdi
0x10f7f8:    push    rbp
0x10f7f9:    push    rbx
0x10f7fa:    mov     rbx, rsi
0x10f7fd:    sub     rsp, 0x18
0x10f801:    call    0x1049b0
0x10f806:    mov     rdi, rbx
0x10f809:    mov     qword ptr [rsp + 8], rax
0x10f80e:    call    0x1049b0
0x10f813:    mov     rdi, r13
0x10f816:    mov     rbp, rax
0x10f819:    call    0x1049b0
0x10f81e:    mov     rdx, qword ptr [rsp + 8]
0x10f823:    lea     rdi, [rdx + rbp + 1]
0x10f828:    add     rdi, rax
0x10f82b:    call    0x11d670
0x10f830:    mov     r9, r13
0x10f833:    mov     r8, rbx
0x10f836:    mov     rcx, r12
0x10f839:    mov     rbp, rax
0x10f83c:    mov     rdi, rax
0x10f83f:    mov     esi, 1
0x10f844:    xor     eax, eax
0x10f846:    mov     rdx, -1
0x10f84d:    call    0x104f10
0x10f852:    mov     edx, r15d
0x10f855:    mov     rsi, r14
0x10f858:    mov     rdi, rbp
0x10f85b:    call    0x108070
0x10f860:    mov     rdi, rbp
0x10f863:    mov     r12d, eax
0x10f866:    call    0x118d10
0x10f86b:    add     rsp, 0x18

```

```

0x10f86f:    mov     eax, r12d
0x10f872:    pop     rbx
0x10f873:    pop     rbp
0x10f874:    pop     r12
0x10f876:    pop     r13
0x10f878:    pop     r14
0x10f87a:    pop     r15
0x10f87c:    ret
0x10f87d:    nop     dword ptr [rax]
0x10f880:    endbr64
0x10f884:    push    r15
0x10f886:    mov     r15, rdi
0x10f889:    mov     rdi, rsi
0x10f88c:    push    r14
0x10f88e:    push    r13
0x10f890:    push    r12
0x10f892:    push    rbp
0x10f893:    mov     rbp, rsi
0x10f896:    push    rbx
0x10f897:    mov     ebx, edx
0x10f899:    sub     rsp, 0x198
0x10f8a0:    mov     rax, qword ptr fs:[0x28]
0x10f8a9:    mov     qword ptr [rsp + 0x188], rax
0x10f8b1:    xor     eax, eax
0x10f8b3:    call    0x104950
0x10f8b8:    mov     r12, rax
0x10f8bb:    test    rax, rax
0x10f8be:    je      0x10fda8
0x10fda8:    cmp     byte ptr [rbp], 0x2e
0x10fdac:    jne     0x10fdf0
0x10f8c4:    mov     rdi, rbp
0x10f8c7:    call    0x1049c0
0x10fdae:    cmp     byte ptr [rbp + 1], 0x2e
0x10fdb2:    jne     0x10fdf0
0x10fdf0:    xor     r13d, r13d
0x10fdf3:    jmp     0x10fd78
0x10f8cc:    cmp     eax, -1
0x10f8cf:    je      0x110050
0x10fdb4:    movzx   r13d, byte ptr [rbp + 2]
0x10fdb9:    test    r13d, r13d
0x10fdbc:    jne     0x10fdf0
0x10fd78:    mov     rax, qword ptr [rsp + 0x188]
0x10fd80:    sub     rax, qword ptr fs:[0x28]
0x10fd89:    jne     0x1100b2
0x110050:    mov     rdi, r12
0x110053:    xor     r13d, r13d
0x110056:    call    0x104ac0
0x10f8d5:    mov     rdi, qword ptr [r15]

```



```

0x10f8d8:    test    rdi, rdi
0x10f8db:    je      0x10f8e2
0x10fdb0:    xor     eax, eax
0x10fdc0:    call    0x118fe0
0x10fd8f:    add     rsp, 0x198
0x10fd96:    mov     eax, r13d
0x10fd99:    pop     rbx
0x10fd9a:    pop     rbp
0x10fd9b:    pop     r12
0x10fd9d:    pop     r13
0x10fd9f:    pop     r14
0x10fda1:    pop     r15
0x10fda3:    ret
0x1100b2:    call    0x1049d0
0x11005b:    jmp     0x10fd78
0x10f8e2:    mov     rax, qword ptr [r15 + 0x78]
0x10f8e6:    mov     qword ptr [r15], r12
0x10f8e9:    lea     rdi, [rax + 1]
0x10f8ed:    call    0x11d670
0x10f8dd:    call    0x104ac0
0x10fdc5:    mov     rbp, rax
0x10fdc8:    test    rax, rax
0x10fdcb:    je      0x110060
0x10f8f2:    mov     rsi, qword ptr [r15 + 0x18]
0x10f8f6:    mov     rdi, rax
0x10f8f9:    mov     qword ptr [rsp + 0x18], rax
0x10f8fe:    call    0x1048b0
0x110060:    mov     rdi, qword ptr [r15 + 0x18]
0x110064:    call    0x11d7a0
0x10fdd1:    mov     rdi, rbp
0x10fdd4:    call    0x104950
0x10f903:    cmp     byte ptr [rbp], 0x2f
0x10f907:    je      0x10fb20
0x110069:    mov     esi, 0x2f
0x11006e:    mov     rdi, rax
0x110071:    mov     rbp, rax
0x110074:    call    0x104a20
0x10fdd9:    mov     r12, rax
0x10fddc:    test    rax, rax
0x10fddf:    jne     0x10f8c4
0x10fb20:    mov     rdi, rbp
0x10fb23:    call    0x11d7a0
0x10f90d:    xor     eax, eax
0x10f90f:    call    0x118fe0
0x110079:    mov     r12, rax
0x11007c:    test    rax, rax
0x11007f:    je      0x1100a4
0x10fde5:    mov     rdi, rbp

```

```

0x10fde8:      call    0x118d10
0x10fb28:      mov     qword ptr [r15 + 0x18], rax
0x10fb2c:      mov     r12, rax
0x10fb2f:      jmp     0x10f92d
0x10f914:      mov     r12, rax
0x10f917:      test    rax, rax
0x10f91a:      je      0x10ff20
0x1100a4:      lea     rdi, [rip + 0x1112a]
0x1100ab:      call    0x108b00
0x110081:      mov     byte ptr [r12], 0
0x110086:      jmp     0x10fdd1
0x10fded:      jmp     0x10fd78
0x10f92d:      mov     rdi, r12
0x10f930:      call    0x118040
0x10ff20:      mov     r13, qword ptr [r15 + 0x18]
0x10ff24:      mov     rdi, r13
0x10ff27:      call    0x1049b0
0x10f920:      mov     rdi, qword ptr [r15 + 0x18]
0x10f924:      call    0x118d10
0x10f935:      mov     rbp, qword ptr [r15 + 0x18]
0x10f939:      mov     rdi, rbp
0x10f93c:      call    0x1049b0
0x10ff2c:      mov     qword ptr [r15 + 0x78], rax
0x10ff30:      cmp     byte ptr [rbp], 0x2e
0x10ff34:      mov     r12, rax
0x10ff37:      jne     0x10ffd0
0x10f929:      mov     qword ptr [r15 + 0x18], r12
0x10f92d:      mov     rdi, r12
0x10f930:      call    0x118040
0x10f941:      lea     rsi, [rsp + 0xf0]
0x10f949:      mov     rdi, rbp
0x10f94c:      mov     qword ptr [r15 + 0x78], rax
0x10f950:      call    0x118f40
0x10ff3d:      cmp     byte ptr [rbp + 1], 0x2e
0x10ff41:      jne     0x10ffd0
0x10ffd0:      mov     rdi, rbp
0x10ffd3:      mov     r14d, 0x2f
0x10ffd9:      call    0x1049b0
0x10f955:      cmp     qword ptr [rsp + 0x120], 0x7ff
0x10f961:      jg      0x10ff08
0x10ff47:      cmp     byte ptr [rbp + 2], 0
0x10ff4b:      jne     0x10ffd0
0x10ffde:      mov     rdi, r13
0x10ffe1:      lea     rsi, [r12 + rax + 2]
0x10ffe6:      call    0x11d690
0x10f967:      lea     rsi, [rsp + 0x70]
0x10f96c:      lea     rdi, [rip + 0x11d6b]
0x10f973:      call    0x104c00

```

```

0x10ff08:    mov     rdi, qword ptr [rip + 0x1a571]
0x10ff0f:    call    0x10db60
0x10ff51:    mov     esi, 0x2f
0x10ff56:    mov     rdi, r13
0x10ff59:    call    0x104a20
0x10ffeb:    mov     qword ptr [r15 + 0x18], rax
0x10ffef:    mov     rdi, rax
0x10fff2:    mov     r12, rax
0x10fff5:    call    0x1049b0
0x10f978:    xor     eax, eax
0x10f97a:    cmp     qword ptr [rsp + 0x70], 0x4d44
0x10f983:    sete    al
0x10f986:    mov     dword ptr [r15 + 0x10], eax
0x10f98a:    test    ebx, ebx
0x10f98c:    je      0x10f99b
0x10ff14:    xor     eax, eax
0x10ff16:    call    0x10a820
0x10ff5e:    test    rax, rax
0x10ff61:    je      0x11008b
0x10fffa:    mov     rsi, rbp
0x10fffd:    mov     word ptr [r12 + rax], r14w
0x110002:    mov     rdi, qword ptr [r15 + 0x18]
0x110006:    call    0x104df0
0x10f99b:    mov     rbp, qword ptr [r15 + 0x38]
0x10f99f:    test    rbp, rbp
0x10f9a2:    je      0x110018
0x10f98e:    mov     r13d, dword ptr [r15 + 0x70]
0x10f992:    test    r13d, r13d
0x10f995:    jne     0x10ff78
0x10ff1b:    jmp     0x10f967
0x11008b:    lea     rdi, [rip + 0x11151]
0x110092:    mov     qword ptr [rsp], rax
0x110096:    call    0x108b00
0x10ff67:    mov     byte ptr [rax], 0
0x10ff6a:    mov     r12, qword ptr [r15 + 0x18]
0x10ff6e:    jmp     0x10f92d
0x11000b:    mov     r12, qword ptr [r15 + 0x18]
0x11000f:    jmp     0x10f92d
0x110018:    mov     dword ptr [rsp + 0xc], 0
0x110020:    mov     dword ptr [rsp], 0
0x110027:    jmp     0x10fa07
0x10f9a8:    mov     r12d, dword ptr [r15 + 0x68]
0x10f9ac:    test    r12d, r12d
0x10f9af:    jle     0x10f9e6
0x10ff78:    mov     rsi, qword ptr [r15 + 0x18]
0x10ff7c:    mov     rdi, qword ptr [rsp + 0x18]
0x10ff81:    call    0x104b30
0x10fa07:    mov     edi, 0x50

```

```

0x10fa0c:    call    0x11d670
0x10f9e6:    mov     rdi, rbp
0x10f9e9:    xor     ebp, ebp
0x10f9eb:    call    0x118d10
0x10f9b1:    xor     ebx, ebx
0x10f9b3:    nop     dword ptr [rax + rax]
0x10f9b8:    lea     r12, [rbx + rbx*4]
0x10f9bc:    shl     r12, 4
0x10f9c0:    mov     rdi, qword ptr [rbp + r12]
0x10f9c5:    test    rdi, rdi
0x10f9c8:    je      0x10f9dc
0x10ff86:    test    eax, eax
0x10ff88:    jne     0x10f99b
0x10fa11:    mov     dword ptr [r15 + 0x70], 0
0x10fa19:    mov     dword ptr [r15 + 0x30], 0
0x10fa21:    mov     dword ptr [rsp + 0x14], 0xffffffff
0x10fa29:    mov     dword ptr [rsp + 0x10], 0
0x10fa31:    mov     qword ptr [r15 + 0x38], rax
0x10fa35:    xor     eax, eax
0x10fa37:    nop     word ptr [rax + rax]
0x10fa40:    mov     dword ptr [r15 + 0x68], eax
0x10fa44:    mov     rdi, qword ptr [r15]
0x10fa47:    call    0x104c10
0x10f9f0:    mov     qword ptr [r15 + 0x38], 0
0x10f9f8:    mov     dword ptr [rsp + 0xc], 0
0x10fa00:    mov     dword ptr [rsp], 0
0x10fa07:    mov     edi, 0x50
0x10fa0c:    call    0x11d670
0x10f9dc:    add     rbx, 1
0x10f9e0:    cmp     dword ptr [r15 + 0x68], ebx
0x10f9e4:    jg      0x10f9b8
0x10f9ca:    call    0x118d10
0x10ff8e:    mov     eax, dword ptr [r15 + 0x68]
0x10ff92:    mov     rbp, qword ptr [r15 + 0x38]
0x10ff96:    mov     dword ptr [rsp + 0xc], 1
0x10ff9e:    mov     qword ptr [r15 + 0x38], 0
0x10ffa6:    mov     dword ptr [rsp], eax
0x10ffa9:    jmp     0x10fa07
0x10fa4c:    test    rax, rax
0x10fa4f:    je      0x10fbf0
0x10f9b8:    lea     r12, [rbx + rbx*4]
0x10f9bc:    shl     r12, 4
0x10f9c0:    mov     rdi, qword ptr [rbp + r12]
0x10f9c5:    test    rdi, rdi
0x10f9c8:    je      0x10f9dc
0x10f9cf:    mov     rbp, qword ptr [r15 + 0x38]
0x10f9d3:    mov     qword ptr [rbp + r12], 0
0x10f9dc:    add     rbx, 1

```

```

0x10f9e0:    cmp     dword ptr [r15 + 0x68], ebx
0x10f9e4:    jg      0x10f9b8
0x10fbf0:    mov     r9d, dword ptr [rsp + 0x10]
0x10fbf5:    test    r9d, r9d
0x10fbf8:    je      0x10fe48
0x10fa55:    movabs  rdx, 0xffff000000
0x10fa5f:    and     rdx, qword ptr [rax + 0x10]
0x10fa63:    cmp     rdx, 0x2e000000
0x10fa6a:    je      0x10fb70
0x10fe48:    mov     eax, dword ptr [r15 + 0x68]
0x10fe4c:    mov     rdi, qword ptr [r15 + 0x38]
0x10fe50:    add     eax, 1
0x10fe53:    cdqe
0x10fe55:    lea     rsi, [rax + rax*4]
0x10fe59:    shl     rsi, 4
0x10fe5d:    call    0x11d690
0x10fbfe:    movsxd  rax, dword ptr [rsp + 0x14]
0x10fc03:    cmp     eax, -1
0x10fc06:    je      0x10fe00
0x10fb70:    mov     eax, dword ptr [r15 + 0x68]
0x10fb74:    jmp     0x10fa40
0x10fa70:    cmp     byte ptr [rax + 0x13], 0x2e
0x10fa74:    lea     r13, [rax + 0x13]
0x10fa78:    je      0x10fb38
0x10fe62:    movsxd  rdx, dword ptr [r15 + 0x68]
0x10fe66:    mov     edi, 3
0x10fe6b:    mov     qword ptr [r15 + 0x38], rax
0x10fe6f:    lea     rdx, [rdx + rdx*4]
0x10fe73:    shl     rdx, 4
0x10fe77:    lea     rbx, [rax + rdx]
0x10fe7b:    mov     byte ptr [rbx + 0x49], 0
0x10fe7f:    mov     byte ptr [rbx + 0x4b], 1
0x10fe83:    call    0x11d670
0x10fe00:    mov     eax, dword ptr [rsp + 0xc]
0x10fe04:    test    eax, eax
0x10fe06:    jne     0x10fc3f
0x10fc0c:    mov     rdi, qword ptr [r15 + 0x38]
0x10fc10:    lea     rax, [rax + rax*4]
0x10fc14:    shl     rax, 4
0x10fc18:    add     rax, rdi
0x10fc1b:    mov     edx, dword ptr [rax + 0x10]
0x10fc1e:    and     edx, 0xf000
0x10fc24:    cmp     edx, 0x4000
0x10fc2a:    je      0x10fc33
0x10fa40:    mov     dword ptr [r15 + 0x68], eax
0x10fa44:    mov     rdi, qword ptr [r15]
0x10fa47:    call    0x104c10
0x10fb38:    cmp     byte ptr [r13 + 1], 0x2e

```

```

0x10fb3d:    jne     0x10fa7e
0x10fa7e:    mov     eax, dword ptr [r15 + 0x68]
0x10fa82:    add     eax, 1
0x10fa85:    mov     rdi, qword ptr [r15 + 0x38]
0x10fa89:    cdqe
0x10fa8b:    lea     rsi, [rax + rax*4]
0x10fa8f:    shl     rsi, 4
0x10fa93:    call    0x11d690
0x10fe88:    mov     ecx, 0x2e2e
0x10fe8d:    mov     edx, 2
0x10fe92:    mov     rdi, r15
0x10fe95:    mov     qword ptr [rbx], rax
0x10fe98:    movsxd  rax, dword ptr [r15 + 0x68]
0x10fe9c:    lea     rax, [rax + rax*4]
0x10fea0:    shl     rax, 4
0x10fea4:    add     rax, qword ptr [r15 + 0x38]
0x10fea8:    mov     rax, qword ptr [rax]
0x10feab:    mov     word ptr [rax], cx
0x10feae:    mov     byte ptr [rax + 2], 0
0x10feb2:    mov     eax, dword ptr [r15 + 0x30]
0x10feb6:    mov     esi, dword ptr [r15 + 0x68]
0x10feba:    cmp     eax, edx
0x10febc:    cmovl   eax, edx
0x10febf:    mov     dword ptr [r15 + 0x30], eax
0x10fec3:    xor     eax, eax
0x10fec5:    call    0x10e840
0x10fe0c:    mov     esi, dword ptr [r15 + 0x68]
0x10fe10:    mov     rdi, qword ptr [r15 + 0x38]
0x10fe14:    mov     dword ptr [r15 + 0x74], 0
0x10fe1c:    mov     eax, dword ptr [r15 + 0x84]
0x10fe23:    mov     dword ptr [rip + 0x1b7ff], eax
0x10fe29:    movsxd  rsi, esi
0x10fe2c:    lea     rcx, [rip - 0x17b3]
0x10fe33:    mov     edx, 0x50
0x10fe38:    call    0x1048e0
0x10fc3f:    mov     eax, dword ptr [rsp]
0x10fc42:    test    eax, eax
0x10fc44:    jle     0x10fc7d
0x10fc33:    mov     edx, dword ptr [rsp + 0xc]
0x10fc37:    test    edx, edx
0x10fc39:    je      0x10ffb0
0x10fc2c:    mov     dword ptr [rax + 0x10], 0x4000
0x10fc33:    mov     edx, dword ptr [rsp + 0xc]
0x10fc37:    test    edx, edx
0x10fc39:    je      0x10ffb0
0x10fb43:    cmp     byte ptr [r13 + 2], 0
0x10fb48:    jne     0x10fa7e
0x10fa98:    mov     ebx, dword ptr [rsp + 0xc]

```

```

0x10fa9c:    mov     qword ptr [r15 + 0x38], rax
0x10faa0:    mov     r14, rax
0x10faa3:    test    ebx, ebx
0x10faa5:    jne     0x10fb90
0x10feca:    movsxd  rax, dword ptr [r15 + 0x68]
0x10fece:    mov     rdi, qword ptr [r15 + 0x38]
0x10fed2:    mov     r8d, dword ptr [rsp + 0xc]
0x10fed7:    mov     rsi, rax
0x10feda:    lea     rax, [rax + rax*4]
0x10fede:    shl     rax, 4
0x10fee2:    add     esi, 1
0x10fee5:    add     rax, rdi
0x10fee8:    mov     dword ptr [rax + 0x10], 0x4000
0x10feef:    mov     byte ptr [rax + 0x48], 0
0x10fef3:    mov     dword ptr [r15 + 0x68], esi
0x10fef7:    test    r8d, r8d
0x10fefa:    je      0x10fc8d
0x10fe3d:    jmp     0x10fd46
0x10fc7d:    mov     rdi, rbp
0x10fc80:    call    0x118d10
0x10fc46:    sub     eax, 1
0x10fc49:    mov     rbx, rbp
0x10fc4c:    lea     rax, [rax + rax*4]
0x10fc50:    shl     rax, 4
0x10fc54:    lea     r12, [rbp + rax + 0x50]
0x10fc59:    nop     dword ptr [rax]
0x10fc60:    mov     rdi, qword ptr [rbx]
0x10fc63:    test    rdi, rdi
0x10fc66:    je      0x10fc74
0x10ffb0:    mov     edx, dword ptr [r15 + 0x84]
0x10ffb7:    mov     esi, dword ptr [r15 + 0x68]
0x10ffbb:    mov     dword ptr [r15 + 0x74], 0
0x10ffc3:    mov     dword ptr [rip + 0x1b65f], edx
0x10ffc9:    jmp     0x10fcbb
0x10fb4e:    mov     rdx, qword ptr [r15 + 0x18]
0x10fb52:    mov     eax, dword ptr [r15 + 0x68]
0x10fb56:    cmp     byte ptr [rdx + 1], 0
0x10fb5a:    je      0x10fb80
0x10faab:    movsxd  rax, dword ptr [r15 + 0x68]
0x10faaf:    lea     rdx, [rax + rax*4]
0x10fab3:    shl     rdx, 4
0x10fab7:    add     r14, rdx
0x10faba:    mov     byte ptr [r14 + 0x49], 0
0x10fabf:    mov     byte ptr [r14 + 0x4b], 0
0x10fac4:    mov     rdi, r13
0x10fac7:    call    0x1049b0
0x10fb90:    mov     r11d, dword ptr [rsp]
0x10fb94:    test    r11d, r11d

```

```

0x10fb97:    jle     0x11002c
0x10fc8d:    mov     eax, dword ptr [r15 + 0x84]
0x10fc94:    mov     dword ptr [r15 + 0x74], 0
0x10fc9c:    mov     dword ptr [rip + 0x1b986], eax
0x10fca2:    movsxd  rax, dword ptr [rsp + 0x14]
0x10fca7:    cmp     eax, -1
0x10fcaa:    je      0x10fe29
0x10ff00:    jmp     0x10fc3f
0x10fd46:    mov     edx, dword ptr [r15 + 0x68]
0x10fd4a:    mov     eax, dword ptr [r15 + 0x28]
0x10fd4e:    mov     rdi, r15
0x10fd51:    mov     r13d, 1
0x10fd57:    cmp     edx, eax
0x10fd59:    lea     ecx, [rdx - 1]
0x10fd5c:    cmovle  eax, ecx
0x10fd5f:    mov     dword ptr [r15 + 0x28], eax
0x10fd63:    xor     eax, eax
0x10fd65:    call    0x10f110
0x10fc85:    mov     esi, dword ptr [r15 + 0x68]
0x10fc89:    mov     rdi, qword ptr [r15 + 0x38]
0x10fc8d:    mov     eax, dword ptr [r15 + 0x84]
0x10fc94:    mov     dword ptr [r15 + 0x74], 0
0x10fc9c:    mov     dword ptr [rip + 0x1b986], eax
0x10fca2:    movsxd  rax, dword ptr [rsp + 0x14]
0x10fca7:    cmp     eax, -1
0x10fcaa:    je      0x10fe29
0x10fc74:    add     rbx, 0x50
0x10fc78:    cmp     r12, rbx
0x10fc7b:    jne     0x10fc60
0x10fc68:    call    0x118d10
0x10fcbb:    movdqu  xmm5, xmmword ptr [rax]
0x10fcbf:    movdqu  xmm4, xmmword ptr [rdi]
0x10fcc3:    sub     esi, 1
0x10fcc6:    add     rdi, 0x50
0x10fcca:    movdqu  xmm3, xmmword ptr [rdi - 0x40]
0x10fccf:    movdqu  xmm2, xmmword ptr [rdi - 0x30]
0x10fcd4:    movsxd  rsi, esi
0x10fcd7:    mov     edx, 0x50
0x10fcdc:    movups  xmmword ptr [rdi - 0x50], xmm5
0x10fce0:    movdqu  xmm6, xmmword ptr [rax + 0x10]
0x10fce5:    movdqu  xmm1, xmmword ptr [rdi - 0x20]
0x10fcea:    lea     rcx, [rip - 0x1671]
0x10fcf1:    movdqu  xmm0, xmmword ptr [rdi - 0x10]
0x10fcf6:    movaps  xmmword ptr [rsp + 0x20], xmm4
0x10fcfb:    movups  xmmword ptr [rdi - 0x40], xmm6
0x10fcff:    movdqu  xmm7, xmmword ptr [rax + 0x20]
0x10fd04:    movaps  xmmword ptr [rsp + 0x30], xmm3
0x10fd09:    movups  xmmword ptr [rdi - 0x30], xmm7

```



```

0x10fd0d:    movdqu  xmm5, xmmword ptr [rax + 0x30]
0x10fd12:    movaps  xmmword ptr [rsp + 0x40], xmm2
0x10fd17:    movups  xmmword ptr [rdi - 0x20], xmm5
0x10fd1b:    movdqu  xmm6, xmmword ptr [rax + 0x40]
0x10fd20:    movaps  xmmword ptr [rsp + 0x50], xmm1
0x10fd25:    movups  xmmword ptr [rdi - 0x10], xmm6
0x10fd29:    movups  xmmword ptr [rax], xmm4
0x10fd2c:    movups  xmmword ptr [rax + 0x10], xmm3
0x10fd30:    movups  xmmword ptr [rax + 0x20], xmm2
0x10fd34:    movups  xmmword ptr [rax + 0x30], xmm1
0x10fd38:    movups  xmmword ptr [rax + 0x40], xmm0
0x10fd3c:    movaps  xmmword ptr [rsp + 0x60], xmm0
0x10fd41:    call    0x1048e0
0x10fb80:    mov     dword ptr [rsp + 0x10], 1
0x10fb88:    jmp     0x10fa40
0x10fb5c:    mov     dword ptr [rsp + 0x14], eax
0x10fb60:    mov     dword ptr [rsp + 0x10], 1
0x10fb68:    jmp     0x10fa82
0x10facc:    lea     edi, [rax + 1]
0x10facf:    mov     rbx, rax
0x10fad2:    movsxd  rdi, edi
0x10fad5:    call    0x11d670
0x11002c:    movsxd  rax, dword ptr [r15 + 0x68]
0x110030:    mov     r10d, dword ptr [rsp]
0x110034:    lea     rdx, [rax + rax*4]
0x110038:    shl     rdx, 4
0x11003c:    add     r14, rdx
0x11003f:    test    r10d, r10d
0x110042:    jne     0x10fabf
0x10fb9d:    mov     rbx, rbp
0x10fba0:    xor     r12d, r12d
0x10fba3:    jmp     0x10fbba
0x10fe29:    movsxd  rsi, esi
0x10fe2c:    lea     rcx, [rip - 0x17b3]
0x10fe33:    mov     edx, 0x50
0x10fe38:    call    0x1048e0
0x10fcb0:    lea     rax, [rax + rax*4]
0x10fcb4:    shl     rax, 4
0x10fcb8:    add     rax, rdi
0x10fcbb:    movdqu  xmm5, xmmword ptr [rax]
0x10fcbf:    movdqu  xmm4, xmmword ptr [rdi]
0x10fcc3:    sub     esi, 1
0x10fcc6:    add     rdi, 0x50
0x10fcc9:    movdqu  xmm3, xmmword ptr [rdi - 0x40]
0x10fccf:    movdqu  xmm2, xmmword ptr [rdi - 0x30]
0x10fcd4:    movsxd  rsi, esi
0x10fcd7:    mov     edx, 0x50
0x10fcdc:    movups  xmmword ptr [rdi - 0x50], xmm5

```

```

0x10fce0:    movdqu  xmm6, xmmword ptr [rax + 0x10]
0x10fce5:    movdqu  xmm1, xmmword ptr [rdi - 0x20]
0x10fcea:    lea      rcx, [rip - 0x1671]
0x10fcf1:    movdqu  xmm0, xmmword ptr [rdi - 0x10]
0x10fcf6:    movaps  xmmword ptr [rsp + 0x20], xmm4
0x10fcfb:    movups  xmmword ptr [rdi - 0x40], xmm6
0x10fcff:    movdqu  xmm7, xmmword ptr [rax + 0x20]
0x10fd04:    movaps  xmmword ptr [rsp + 0x30], xmm3
0x10fd09:    movups  xmmword ptr [rdi - 0x30], xmm7
0x10fd0d:    movdqu  xmm5, xmmword ptr [rax + 0x30]
0x10fd12:    movaps  xmmword ptr [rsp + 0x40], xmm2
0x10fd17:    movups  xmmword ptr [rdi - 0x20], xmm5
0x10fd1b:    movdqu  xmm6, xmmword ptr [rax + 0x40]
0x10fd20:    movaps  xmmword ptr [rsp + 0x50], xmm1
0x10fd25:    movups  xmmword ptr [rdi - 0x10], xmm6
0x10fd29:    movups  xmmword ptr [rax], xmm4
0x10fd2c:    movups  xmmword ptr [rax + 0x10], xmm3
0x10fd30:    movups  xmmword ptr [rax + 0x20], xmm2
0x10fd34:    movups  xmmword ptr [rax + 0x30], xmm1
0x10fd38:    movups  xmmword ptr [rax + 0x40], xmm0
0x10fd3c:    movaps  xmmword ptr [rsp + 0x60], xmm0
0x10fd41:    call    0x1048e0
0x10fd6a:    mov     rdi, qword ptr [rsp + 0x18]
0x10fd6f:    mov     dword ptr [r15 + 0x2c], eax
0x10fd73:    call    0x118d10
0x10fc60:    mov     rdi, qword ptr [rbx]
0x10fc63:    test    rdi, rdi
0x10fc66:    je      0x10fc74
0x10fc6d:    mov     qword ptr [rbx], 0
0x10fc74:    add     rbx, 0x50
0x10fc78:    cmp     r12, rbx
0x10fc7b:    jne     0x10fc60
0x10fa82:    add     eax, 1
0x10fa85:    mov     rdi, qword ptr [r15 + 0x38]
0x10fa89:    cdqe
0x10fa8b:    lea     rsi, [rax + rax*4]
0x10fa8f:    shl     rsi, 4
0x10fa93:    call    0x11d690
0x10fada:    mov     rsi, r13
0x10fadd:    mov     qword ptr [r14], rax
0x10fae0:    movsxd  rax, dword ptr [r15 + 0x68]
0x10fae4:    lea     rax, [rax + rax*4]
0x10fae8:    shl     rax, 4
0x10faec:    add     rax, qword ptr [r15 + 0x38]
0x10faf0:    mov     rdi, qword ptr [rax]
0x10faf3:    call    0x1048b0
0x110048:    jmp     0x10faba
0x10fabf:    mov     byte ptr [r14 + 0x4b], 0

```

```

0x10fac4:    mov     rdi, r13
0x10fac7:    call    0x1049b0
0x10fbba:    mov     rsi, qword ptr [rbx]
0x10fbbd:    mov     rdi, r13
0x10fbc0:    call    0x104b30
0x10faf8:    mov     eax, dword ptr [r15 + 0x30]
0x10fafc:    mov     esi, dword ptr [r15 + 0x68]
0x10fb00:    mov     rdi, r15
0x10fb03:    cmp     eax, ebx
0x10fb05:    cmovl   eax, ebx
0x10fb08:    mov     dword ptr [r15 + 0x30], eax
0x10fb0c:    xor     eax, eax
0x10fb0e:    call    0x10e840
0x10faba:    mov     byte ptr [r14 + 0x49], 0
0x10fabf:    mov     byte ptr [r14 + 0x4b], 0
0x10fac4:    mov     rdi, r13
0x10fac7:    call    0x1049b0
0x10fbc5:    test    eax, eax
0x10fbc7:    jne     0x10fba8
0x10fb13:    mov     eax, dword ptr [r15 + 0x68]
0x10fb17:    add     eax, 1
0x10fb1a:    jmp     0x10fa40
0x10fbc9:    movsxd  rax, dword ptr [r15 + 0x68]
0x10fbcd:    lea     rdx, [rax + rax*4]
0x10fbd1:    movsx   eax, byte ptr [rbx + 0x49]
0x10fbd5:    shl     rdx, 4
0x10fbd9:    add     r14, rdx
0x10fbdc:    mov     byte ptr [r14 + 0x49], al
0x10fbe0:    add     dword ptr [r15 + 0x70], eax
0x10fbe4:    jmp     0x10fabf
0x10fba8:    add     r12d, 1
0x10fbac:    add     rbx, 0x50
0x10fbb0:    cmp     dword ptr [rsp], r12d
0x10fbb4:    je      0x10faab
0x10fb1f:    nop
0x10fb34:    nop     dword ptr [rax]
0x10fb6d:    nop     dword ptr [rax]
0x10fb79:    nop     dword ptr [rax]
0x10fb8d:    nop     dword ptr [rax]
0x10fba5:    nop     dword ptr [rax]
0x10fbe9:    nop     dword ptr [rax]
0x10fda4:    nop     dword ptr [rax]
0x10fdef:    nop
0x10fdf8:    nop     dword ptr [rax + rax]
0x10fe42:    nop     word ptr [rax + rax]
0x10ff05:    nop     dword ptr [rax]
0x10ff73:    nop     dword ptr [rax + rax]
0x10ffae:    nop

```

```

0x10ffce:    nop
0x110014:    nop        dword ptr [rax]
0x11004d:    nop        dword ptr [rax]
0x11009b:    mov        rax, qword ptr [rsp]
0x11009f:    jmp        0x10ff67
0x1100b0:    jmp        0x110081
0x1100b7:    nop        word ptr [rax + rax]
0x1100c0:    endbr64
0x1100c4:    mov        eax, dword ptr [rdi + 0x70]
0x1100c7:    mov        dword ptr [rdi + 0x50], 0xffffffff
0x1100ce:    mov        dword ptr [rdi + 0x54], eax
0x1100d1:    ret
0x1100d2:    nop        word ptr cs:[rax + rax]
0x1100dd:    nop        dword ptr [rax]
0x1100e0:    endbr64
0x1100e4:    mov        eax, dword ptr [rdi + 0x54]
0x1100e7:    movsxd    rdx, dword ptr [rdi + 0x50]
0x1100eb:    test       eax, eax
0x1100ed:    je         0x110128
0x110128:    test       edx, edx
0x11012a:    je         0x110154
0x1100ef:    mov        ecx, dword ptr [rdi + 0x68]
0x1100f2:    lea        eax, [rdx + 1]
0x1100f5:    cmp        ecx, eax
0x1100f7:    jle        0x110154
0x110154:    mov        eax, 0xffffffff
0x110159:    ret
0x11012c:    mov        rdx, qword ptr [rdi + 0x18]
0x110130:    mov        eax, dword ptr [rdi + 0x28]
0x110133:    mov        dword ptr [rdi + 0x50], 0
0x11013a:    cmp        byte ptr [rdx], 0x2f
0x11013d:    jne        0x110150
0x1100f9:    mov        rsi, qword ptr [rdi + 0x38]
0x1100fd:    lea        rdx, [rdx + rdx*4]
0x110101:    shl        rdx, 4
0x110105:    lea        rdx, [rsi + rdx + 0x99]
0x11010d:    jmp        0x11011b
0x11013f:    cmp        byte ptr [rdx + 1], 0
0x110143:    jne        0x110150
0x110150:    test       eax, eax
0x110152:    jne        0x110145
0x11011b:    cmp        byte ptr [rdx], 0
0x11011e:    je         0x110110
0x110145:    ret
0x110110:    add        eax, 1
0x110113:    add        rdx, 0x50
0x110117:    cmp        ecx, eax
0x110119:    je         0x110154

```

```

0x110120:    mov     dword ptr [rdi + 0x50], eax
0x110123:    ret
0x11010f:    nop
0x110124:    nop     dword ptr [rax]
0x110146:    nop     word ptr cs:[rax + rax]
0x11015a:    nop     word ptr [rax + rax]
0x110160:    endbr64
0x110164:    push    r13
0x110166:    push    r12
0x110168:    push    rbp
0x110169:    push    rbx
0x11016a:    sub     rsp, 0x18
0x11016e:    mov     rax, qword ptr fs:[0x28]
0x110177:    mov     qword ptr [rsp + 8], rax
0x11017c:    xor     eax, eax
0x11017e:    cmp     dword ptr [rdi + 0x64], 0x13
0x110182:    jle     0x11018d
0x11018d:    mov     rax, qword ptr [rsp + 8]
0x110192:    sub     rax, qword ptr fs:[0x28]
0x11019b:    jne     0x110236
0x110184:    cmp     byte ptr [rdi + 0x6d], 0
0x110188:    mov     rbx, rdi
0x11018b:    jne     0x1101b0
0x1101a1:    add     rsp, 0x18
0x1101a5:    pop     rbx
0x1101a6:    pop     rbp
0x1101a7:    pop     r12
0x1101a9:    pop     r13
0x1101ab:    ret
0x110236:    call    0x1049d0
0x1101b0:    lea     r13, [rsp + 7]
0x1101b5:    mov     rdi, r13
0x1101b8:    call    0x10a6a0
0x1101bd:    movsxd  rdx, dword ptr [rbx + 0x64]
0x1101c1:    mov     rdi, qword ptr [rbx + 0x20]
0x1101c5:    mov     esi, 0x20
0x1101ca:    mov     rbp, rdx
0x1101cd:    lea     r12d, [rdx - 0x13]
0x1101d1:    call    0x104a50
0x1101d6:    mov     rsi, qword ptr [rbx + 0x20]
0x1101da:    mov     rdi, qword ptr [rbx + 0x18]
0x1101de:    mov     edx, r12d
0x1101e1:    call    0x118200
0x1101e6:    mov     rdi, qword ptr [rbx + 0x20]
0x1101ea:    movsxd  rsi, r12d
0x1101ed:    call    0x1184d0
0x1101f2:    mov     edx, dword ptr [rip + 0x1a128]
0x1101f8:    mov     esi, dword ptr [rip + 0x1a152]

```

```

0x1101fe:    mov     edi, dword ptr [rip + 0x1a150]
0x110204:    call   0x10a570
0x110209:    mov     rdi, qword ptr [rbx + 8]
0x11020d:    xor     esi, esi
0x11020f:    mov     edx, 2
0x110214:    call   0x10c680
0x110219:    mov     rdi, qword ptr [rbx + 8]
0x11021d:    mov     rsi, qword ptr [rbx + 0x20]
0x110221:    lea     edx, [rbp - 0x12]
0x110224:    call   0x10c600
0x110229:    mov     rdi, r13
0x11022c:    call   0x10a6b0
0x110231:    jmp     0x11018d
0x1101ac:    nop     dword ptr [rax]
0x11023b:    nop     dword ptr [rax + rax]
0x110240:    endbr64
0x110244:    push    r15
0x110246:    push    r14
0x110248:    push    r13
0x11024a:    push    r12
0x11024c:    push    rbp
0x11024d:    push    rbx
0x11024e:    sub     rsp, 0x2f8
0x110255:    mov     rax, qword ptr fs:[0x28]
0x11025e:    mov     qword ptr [rsp + 0x2e8], rax
0x110266:    xor     eax, eax
0x110268:    cmp     byte ptr [rdi + 0x6d], 0
0x11026c:    jne     0x1102a0
0x11026e:    mov     rax, qword ptr [rsp + 0x2e8]
0x110276:    sub     rax, qword ptr fs:[0x28]
0x11027f:    jne     0x1103f2
0x1102a0:    mov     eax, dword ptr [rdi + 0x64]
0x1102a3:    lea     r14, [rsp + 0xf]
0x1102a8:    mov     rbx, rdi
0x1102ab:    mov     rdi, r14
0x1102ae:    lea     r12d, [rax - 4]
0x1102b2:    call   0x10a6a0
0x110285:    add     rsp, 0x2f8
0x11028c:    pop     rbx
0x11028d:    pop     rbp
0x11028e:    pop     r12
0x110290:    pop     r13
0x110292:    pop     r14
0x110294:    pop     r15
0x110296:    ret
0x1103f2:    call   0x1049d0
0x1102b7:    mov     qword ptr [rsp + 0x18], -1
0x1102c0:    cmp     r12d, 5

```

```

0x1102c4:    jle     0x110350
0x110350:    lea     rbp, [rsp + 0x50]
0x110355:    mov     ecx, 0x51
0x11035a:    movabs  rax, 0x2020202020202020
0x110364:    mov     rdi, rbp
0x110367:    rep stosq      qword ptr [rdi], rax
0x1102ca:    mov     rdi, qword ptr [rbx + 0x18]
0x1102ce:    xor     esi, esi
0x1102d0:    lea     rdx, [rsp + 0x10]
0x1102d5:    call    0x11c490
0x11036a:    mov     eax, 0x2020
0x11036f:    mov     word ptr [rdi], ax
0x110372:    mov     byte ptr [rdi + 2], 0x20
0x110376:    xor     edi, edi
0x110378:    call    0x10a530
0x110367:    rep stosq      qword ptr [rdi], rax
0x1102da:    test    eax, eax
0x1102dc:    js      0x110350
0x11037d:    mov     edi, dword ptr [rip + 0x19f9d]
0x110383:    call    0x10a4f0
0x1102de:    cmp     qword ptr [rsp + 0x18], -1
0x1102e4:    je      0x110350
0x110388:    mov     edi, dword ptr [rip + 0x19f92]
0x11038e:    movsxd  r15, r12d
0x110391:    call    0x10a510
0x1102e6:    call    0x104a60
0x110396:    mov     rdi, rbp
0x110399:    mov     r13d, dword ptr [rbx + 0x64]
0x11039d:    call    0x1049b0
0x1102eb:    test    eax, eax
0x1102ed:    je      0x1103e8
0x1103a2:    mov     rdi, qword ptr [rbx + 8]
0x1103a6:    cmp     rax, r15
0x1103a9:    lea     edx, [r13 - 2]
0x1103ad:    cmova   eax, r12d
0x1103b1:    xor     esi, esi
0x1103b3:    sub     edx, eax
0x1103b5:    call    0x10c680
0x1103e8:    mov     rdi, qword ptr [rsp + 0x20]
0x1103ed:    jmp     0x1102f8
0x1102f3:    mov     rdi, qword ptr [rsp + 0x28]
0x1102f8:    xor     eax, eax
0x1102fa:    test    rdi, rdi
0x1102fd:    mov     edx, 0xc
0x110302:    mov     r8d, 1
0x110308:    cmovs   rdi, rax
0x11030c:    imul    rdi, qword ptr [rsp + 0x10]
0x110312:    mov     eax, 0x9c

```

```

0x110317:    mov     ecx, 1
0x11031c:    lea     rsi, [rsp + 0x50]
0x110321:    cmp     rdi, 0x2710
0x110328:    cmovge  edx, eax
0x11032b:    call    0x11c520
0x1103ba:    mov     rdi, rbp
0x1103bd:    call    0x1049b0
0x1102f8:    xor     eax, eax
0x1102fa:    test    rdi, rdi
0x1102fd:    mov     edx, 0xc
0x110302:    mov     r8d, 1
0x110308:    cmovs   rdi, rax
0x11030c:    imul    rdi, qword ptr [rsp + 0x10]
0x110312:    mov     eax, 0x9c
0x110317:    mov     ecx, 1
0x11031c:    lea     rsi, [rsp + 0x50]
0x110321:    cmp     rdi, 0x2710
0x110328:    cmovge  edx, eax
0x11032b:    call    0x11c520
0x110330:    mov     edi, dword ptr [rip + 0x1a026]
0x110336:    mov     rbp, rax
0x110339:    call    0x10a530
0x1103c2:    mov     rdi, qword ptr [rbx + 8]
0x1103c6:    mov     rsi, rbp
0x1103c9:    cmp     r15, rax
0x1103cc:    cmovae  r12d, eax
0x1103d0:    mov     edx, r12d
0x1103d3:    call    0x10c600
0x11033e:    mov     edi, dword ptr [rip + 0x1a014]
0x110344:    call    0x10a4f0
0x1103d8:    mov     rdi, r14
0x1103db:    call    0x10a6b0
0x110349:    jmp     0x110388
0x1103e0:    jmp     0x11026e
0x110297:    nop     word ptr [rax + rax]
0x11034b:    nop     dword ptr [rax + rax]
0x1103e5:    nop     dword ptr [rax]
0x1103f7:    nop     word ptr [rax + rax]
0x110400:    endbr64
0x110404:    cmp     byte ptr [rdi + 0x6d], 0
0x110408:    jne     0x110410
0x11040a:    ret
0x110410:    xor     eax, eax
0x110412:    jmp     0x10ec50
0x11040b:    nop     dword ptr [rax + rax]
0x110417:    nop     word ptr [rax + rax]
0x110420:    endbr64
0x110424:    push    r12

```



```

0x110426:    push    rbp
0x110427:    push    rbx
0x110428:    sub     rsp, 0x2a0
0x11042f:    mov     rax, qword ptr fs:[0x28]
0x110438:    mov     qword ptr [rsp + 0x298], rax
0x110440:    xor     eax, eax
0x110442:    cmp     edx, 5
0x110445:    ja      0x110660
0x11044b:    mov     ebp, ecx
0x11044d:    mov     edx, edx
0x11044f:    lea     rcx, [rip + 0x1146e]
0x110456:    mov     rbx, rdi
0x110459:    movsxd  rax, dword ptr [rcx + rdx*4]
0x11045d:    add     rax, rcx
0x110460:    jmp     rax
0x110660:    mov     rax, qword ptr [rsp + 0x298]
0x110668:    sub     rax, qword ptr fs:[0x28]
0x110671:    jne     0x11075a
0x110580:    movsxd  rsi, esi
0x110583:    mov     edx, 0xc
0x110588:    mov     ecx, 1
0x11058d:    mov     r8d, 1
0x110593:    lea     rax, [rsi + rsi*4]
0x110597:    mov     rsi, rsp
0x11059a:    shl     rax, 4
0x11059e:    add     rax, qword ptr [rdi + 0x38]
0x1105a2:    mov     rdi, qword ptr [rax + 8]
0x1105a6:    mov     eax, 0x9c
0x1105ab:    cmp     rdi, 0x2710
0x1105b2:    cmovge  edx, eax
0x1105b5:    call    0x11c520
0x110468:    movsxd  rdx, dword ptr [rdi + 0x64]
0x11046c:    movsxd  rsi, esi
0x11046f:    movsxd  rcx, ebp
0x110472:    lea     rax, [rsi + rsi*4]
0x110476:    mov     rsi, qword ptr [rdi + 0x20]
0x11047a:    sub     rdx, rcx
0x11047d:    shl     rax, 4
0x110481:    lea     rdx, [rsi + rdx - 2]
0x110486:    mov     rsi, qword ptr [rdi + 0x38]
0x11048a:    mov     rsi, qword ptr [rsi + rax + 0x28]
0x11048f:    mov     edi, dword ptr [rsi]
0x110491:    mov     dword ptr [rdx], edi
0x110493:    movzx   edi, word ptr [rsi + 4]
0x110497:    mov     word ptr [rdx + 4], di
0x11049b:    movzx   esi, byte ptr [rsi + 6]
0x11049f:    mov     byte ptr [rdx + 6], sil
0x1104a3:    movsxd  rdx, dword ptr [rbx + 0x64]

```

```

0x1104a7:    sub    rdx, rcx
0x1104aa:    mov    rcx, qword ptr [rbx + 0x20]
0x1104ae:    lea    rdx, [rcx + rdx + 6]
0x1104b3:    mov    rcx, qword ptr [rbx + 0x38]
0x1104b7:    mov    rax, qword ptr [rcx + rax + 0x30]
0x1104bc:    mov    ecx, dword ptr [rax]
0x1104be:    mov    dword ptr [rdx], ecx
0x1104c0:    movzx  ecx, word ptr [rax + 4]
0x1104c4:    mov    word ptr [rdx + 4], cx
0x1104c8:    movzx  eax, byte ptr [rax + 6]
0x1104cc:    mov    byte ptr [rdx + 6], al
0x1104cf:    nop
0x1104d0:    mov    rax, qword ptr [rsp + 0x298]
0x1104d8:    sub    rax, qword ptr fs:[0x28]
0x1104e1:    jne    0x11075a
0x110530:    movsxd rax, dword ptr [rdi + 0x64]
0x110534:    mov    rdx, qword ptr [rdi + 0x20]
0x110538:    movsxd rcx, ebp
0x11053b:    movsxd rsi, esi
0x11053e:    sub    rax, rcx
0x110541:    lea    rdx, [rdx + rax - 2]
0x110546:    lea    rax, [rsi + rsi*4]
0x11054a:    shl    rax, 4
0x11054e:    add    rax, qword ptr [rdi + 0x38]
0x110552:    lea    rcx, [rax + 0x38]
0x110556:    mov    rax, qword ptr [rax + 0x38]
0x11055a:    mov    qword ptr [rdx], rax
0x11055d:    mov    eax, dword ptr [rcx + 8]
0x110560:    mov    dword ptr [rdx + 8], eax
0x110563:    movzx  eax, word ptr [rcx + 0xc]
0x110567:    mov    word ptr [rdx + 0xc], ax
0x11056b:    movzx  eax, byte ptr [rcx + 0xe]
0x11056f:    mov    byte ptr [rdx + 0xe], al
0x110572:    jmp    0x1104d0
0x1104d0:    mov    rax, qword ptr [rsp + 0x298]
0x1104d8:    sub    rax, qword ptr fs:[0x28]
0x1104e1:    jne    0x11075a
0x1104f8:    mov    rdx, rsp
0x1104fb:    xor    eax, eax
0x1104fd:    call   0x10e560
0x110677:    add    rsp, 0x2a0
0x11067e:    lea    rdi, [rip + 0x10b67]
0x110685:    pop    rbx
0x110686:    pop    rbp
0x110687:    pop    r12
0x110689:    jmp    0x108b00
0x11075a:    call   0x1049d0
0x1105ba:    mov    rdi, rax

```

```

0x1105bd:      mov     r12, rax
0x1105c0:      call   0x1049b0
0x1104e7:      add     rsp, 0x2a0
0x1104ee:      pop     rbx
0x1104ef:      pop     rbp
0x1104f0:      pop     r12
0x1104f2:      ret
0x110502:      movsxd  rax, dword ptr [rbx + 0x64]
0x110506:      mov     rdx, qword ptr [rbx + 0x20]
0x11050a:      movsxd  rcx, ebp
0x11050d:      sub     rax, rcx
0x110510:      lea     rax, [rdx + rax - 2]
0x110515:      mov     rdx, qword ptr [rsp]
0x110519:      mov     qword ptr [rax], rdx
0x11051c:      movzx   edx, word ptr [rsp + 8]
0x110521:      mov     word ptr [rax + 8], dx
0x110525:      jmp     0x1104d0
0x1105c5:      mov     edx, 0xa
0x1105ca:      movsxd  rsi, dword ptr [rbx + 0x64]
0x1105ce:      movsxd  rcx, ebp
0x1105d1:      cmp     rax, rdx
0x1105d4:      cmovbe  rdx, rax
0x1105d8:      sub     rsi, rcx
0x1105db:      mov     rcx, qword ptr [rbx + 0x20]
0x1105df:      lea     rcx, [rcx + rsi - 2]
0x1105e4:      cmp     rax, 9
0x1105e8:      ja      0x110621
0x1105ea:      movabs  rdi, 0x2020202020202020
0x1105f4:      mov     eax, 0xa
0x1105f9:      sub     eax, edx
0x1105fb:      movsxd  r10, eax
0x1105fe:      cmp     eax, 8
0x110601:      jae     0x1106e0
0x110621:      cmp     edx, 8
0x110624:      jae     0x110690
0x110607:      test    al, 4
0x110609:      jne     0x110734
0x1106e0:      mov     esi, eax
0x1106e2:      lea     r8, [rcx + 8]
0x1106e6:      mov     qword ptr [rcx], rdi
0x1106e9:      and     r8, 0xfffffffffffffff8
0x1106ed:      mov     qword ptr [rcx + rsi - 8], rdi
0x1106f2:      mov     rsi, rcx
0x1106f5:      sub     rsi, r8
0x1106f8:      add     eax, esi
0x1106fa:      and     eax, 0xffffffff8
0x1106fd:      cmp     eax, 8
0x110700:      jb      0x11061e

```

```

0x110626:    test    dl, 4
0x110629:    jne     0x11071e
0x110690:    mov     rax, qword ptr [r12]
0x110694:    mov     qword ptr [rcx], rax
0x110697:    mov     eax, edx
0x110699:    mov     rsi, qword ptr [r12 + rax - 8]
0x11069e:    mov     qword ptr [rcx + rax - 8], rsi
0x1106a3:    lea     rsi, [rcx + 8]
0x1106a7:    and     rsi, 0xffffffffffff8
0x1106ab:    sub     rcx, rsi
0x1106ae:    add     edx, ecx
0x1106b0:    sub     r12, rcx
0x1106b3:    and     edx, 0xffffffff8
0x1106b6:    cmp     edx, 8
0x1106b9:    jb      0x1104d0
0x11060f:    test    eax, eax
0x110611:    je      0x11061e
0x110734:    mov     eax, eax
0x110736:    mov     dword ptr [rcx], 0x20202020
0x11073c:    mov     dword ptr [rcx + rax - 4], 0x20202020
0x110744:    jmp     0x11061e
0x11061e:    add     rcx, r10
0x110621:    cmp     edx, 8
0x110624:    jae     0x110690
0x110706:    and     eax, 0xffffffff8
0x110709:    xor     esi, esi
0x11070b:    mov     r9d, esi
0x11070e:    add     esi, 8
0x110711:    mov     qword ptr [r8 + r9], rdi
0x110715:    cmp     esi, eax
0x110717:    jb      0x11070b
0x11062f:    test    edx, edx
0x110631:    je      0x1104d0
0x11071e:    mov     eax, dword ptr [r12]
0x110722:    mov     edx, edx
0x110724:    mov     dword ptr [rcx], eax
0x110726:    mov     eax, dword ptr [r12 + rdx - 4]
0x11072b:    mov     dword ptr [rcx + rdx - 4], eax
0x11072f:    jmp     0x1104d0
0x1106bf:    and     edx, 0xffffffff8
0x1106c2:    xor     eax, eax
0x1106c4:    mov     ecx, eax
0x1106c6:    add     eax, 8
0x1106c9:    mov     rdi, qword ptr [r12 + rcx]
0x1106cd:    mov     qword ptr [rsi + rcx], rdi
0x1106d1:    cmp     eax, edx
0x1106d3:    jb      0x1106c4
0x110613:    mov     byte ptr [rcx], 0x20

```

```

0x110616:    test    al, 2
0x110618:    jne     0x110749
0x11070b:    mov     r9d, esi
0x11070e:    add     esi, 8
0x110711:    mov     qword ptr [r8 + r9], rdi
0x110715:    cmp     esi, eax
0x110717:    jb     0x11070b
0x110719:    jmp     0x11061e
0x110637:    movzx   eax, byte ptr [r12]
0x11063c:    mov     byte ptr [rcx], al
0x11063e:    test    dl, 2
0x110641:    je     0x1104d0
0x1106c4:    mov     ecx, eax
0x1106c6:    add     eax, 8
0x1106c9:    mov     rdi, qword ptr [r12 + rcx]
0x1106cd:    mov     qword ptr [rsi + rcx], rdi
0x1106d1:    cmp     eax, edx
0x1106d3:    jb     0x1106c4
0x1106d5:    jmp     0x1104d0
0x110749:    mov     eax, eax
0x11074b:    mov     esi, 0x2020
0x110750:    mov     word ptr [rcx + rax - 2], si
0x110755:    jmp     0x11061e
0x110647:    mov     edx, edx
0x110649:    movzx   eax, word ptr [r12 + rdx - 2]
0x11064f:    mov     word ptr [rcx + rdx - 2], ax
0x110654:    jmp     0x1104d0
0x110463:    nop     dword ptr [rax + rax]
0x1104f3:    nop     dword ptr [rax + rax]
0x110527:    nop     word ptr [rax + rax]
0x110577:    nop     word ptr [rax + rax]
0x110659:    nop     dword ptr [rax]
0x11068e:    nop
0x1106da:    nop     word ptr [rax + rax]
0x11075f:    nop
0x110760:    push    r15
0x110762:    push    r14
0x110764:    mov     r14d, 4
0x11076a:    push    r13
0x11076c:    mov     r13, rdi
0x11076f:    push    r12
0x110771:    push    rbp
0x110772:    push    rbx
0x110773:    mov     ebx, esi
0x110775:    mov     esi, 0x20
0x11077a:    sub     rsp, 0x28
0x11077e:    movsxd  rdx, dword ptr [rdi + 0x64]
0x110782:    mov     rdi, qword ptr [rdi + 0x20]

```

```

0x110786:      call    0x104a50
0x11078b:      mov     ebp, dword ptr [r13 + 0x64]
0x11078f:      cmp     ebp, 0x27
0x110792:      jle     0x1107b0
0x1107b0:      movsxd   rax, ebx
0x1107b3:      lea     r12, [rax + rax*4]
0x1107b7:      mov     rax, qword ptr [r13 + 0x38]
0x1107bb:      shl     r12, 4
0x1107bf:      add     rax, r12
0x1107c2:      test    ebx, ebx
0x1107c4:      jle     0x1107e8
0x110794:      movsxd   rax, dword ptr [r13 + 0x80]
0x11079b:      cmp     ebp, 0x49
0x11079e:      jg      0x1107a5
0x1107e8:      mov     rdx, qword ptr [r13 + 0x18]
0x1107ec:      mov     byte ptr [rsp + 8], 0
0x1107f1:      cmp     byte ptr [rdx], 0x2f
0x1107f4:      je      0x110c18
0x1107c6:      cmp     byte ptr [rax + 0x48], 4
0x1107ca:      ja      0x104f20
0x1107a0:      cmp     eax, 5
0x1107a3:      je      0x1107b0
0x1107a5:      lea     rdx, [rip + 0x11234]
0x1107ac:      movsxd   r14, dword ptr [rdx + rax*4]
0x1107b0:      movsxd   rax, ebx
0x1107b3:      lea     r12, [rax + rax*4]
0x1107b7:      mov     rax, qword ptr [r13 + 0x38]
0x1107bb:      shl     r12, 4
0x1107bf:      add     rax, r12
0x1107c2:      test    ebx, ebx
0x1107c4:      jle     0x1107e8
0x110c18:      cmp     byte ptr [rdx + 1], 0
0x110c1c:      jne     0x1107fa
0x1107d0:      movzx    edx, byte ptr [rax + 0x48]
0x1107d4:      lea     rcx, [rip + 0x11101]
0x1107db:      movsxd   rdx, dword ptr [rcx + rdx*4]
0x1107df:      add     rdx, rcx
0x1107e2:      jmp     rdx
0x110c22:      jmp     0x1107c6
0x110b00:      mov     byte ptr [rsp + 8], 0x2f
0x110b05:      jmp     0x110ab5
0x110ac0:      mov     byte ptr [rsp + 8], 0x7c
0x110ac5:      jmp     0x110ab5
0x110ab0:      mov     byte ptr [rsp + 8], 0x3d
0x110ab5:      add     r14, 1
0x110ab9:      jmp     0x1107fa
0x110af0:      mov     byte ptr [rsp + 8], 0x40
0x110af5:      jmp     0x110ab5

```

```

0x110ad0:    movzx    ecx, byte ptr [rax + 0x4a]
0x110ad4:    mov      byte ptr [rsp + 8], cl
0x110ad8:    test     cl, cl
0x110ada:    je       0x1107fa
0x110ab5:    add      r14, 1
0x110ab9:    jmp      0x1107fa
0x110ae0:    mov      byte ptr [rsp + 8], 0x2a
0x110ae5:    jmp      0x110ab5
0x1107e5:    nop      dword ptr [rax]
0x11089e:    nop
0x110942:    nop      word ptr [rax + rax]
0x1109b8:    nop      dword ptr [rax + rax]
0x1109d6:    nop      word ptr cs:[rax + rax]
0x110a4c:    nop      dword ptr [rax]
0x110a8b:    nop      dword ptr [rax + rax]
0x110aa7:    nop      word ptr [rax + rax]
0x110abe:    nop
0x110ac7:    nop      word ptr [rax + rax]
0x110ae7:    nop      word ptr [rax + rax]
0x110af7:    nop      word ptr [rax + rax]
0x110b07:    nop      word ptr [rax + rax]
0x110c03:    nop      dword ptr [rax + rax]
0x110c13:    nop      dword ptr [rax + rax]
0x110c27:    nop      word ptr [rax + rax]
0x110c46:    nop      word ptr cs:[rax + rax]
0x110c66:    nop      word ptr cs:[rax + rax]
0x110c8a:    nop      word ptr [rax + rax]
0x110d63:    nop      word ptr cs:[rax + rax]
0x110d6e:    nop
0x110d70:    push     r13
0x110d72:    push     r12
0x110d74:    push     rbp
0x110d75:    push     rbx
0x110d76:    mov      rbx, rdi
0x110d79:    sub      rsp, 0x18
0x110d7d:    mov      rax, qword ptr fs:[0x28]
0x110d86:    mov      qword ptr [rsp + 8], rax
0x110d8b:    xor      eax, eax
0x110d8d:    lea      r12, [rsp + 7]
0x110d92:    mov      rdi, r12
0x110d95:    call     0x10a6a0
0x110d9a:    mov      ebp, dword ptr [rbx + 0x2c]
0x110d9d:    mov      eax, ebp
0x110d9f:    cmp      ebp, dword ptr [rbx + 0x68]
0x110da2:    jl       0x110dcb
0x110dcb:    mov      edx, ebp
0x110dcd:    sub      edx, eax
0x110dcf:    mov      eax, dword ptr [rbx + 0x60]

```

```

0x110dd2:    sub     eax, 2
0x110dd5:    cmp     edx, eax
0x110dd7:    jl      0x110db0
0x110da4:    jmp     0x110dd9
0x110db0:    cmp     byte ptr [rbx + 0x6d], 0
0x110db4:    je      0x110dc0
0x110dd9:    mov     edx, dword ptr [rip + 0x19545]
0x110ddf:    xor     edi, edi
0x110de1:    mov     esi, 7
0x110de6:    call    0x10a570
0x110dc0:    add     ebp, 1
0x110dc3:    cmp     ebp, dword ptr [rbx + 0x68]
0x110dc6:    jge     0x110dd9
0x110db6:    mov     esi, ebp
0x110db8:    mov     rdi, rbx
0x110dbb:    call    0x110760
0x110deb:    movsxd  rdx, dword ptr [rbx + 0x64]
0x110def:    mov     rdi, qword ptr [rbx + 0x20]
0x110df3:    mov     esi, 0x20
0x110df8:    call    0x104a50
0x110dc8:    mov     eax, dword ptr [rbx + 0x2c]
0x110dcb:    mov     edx, ebp
0x110dcd:    sub     edx, eax
0x110dcf:    mov     eax, dword ptr [rbx + 0x60]
0x110dd2:    sub     eax, 2
0x110dd5:    cmp     edx, eax
0x110dd7:    jl      0x110db0
0x110dfd:    mov     eax, dword ptr [rbx + 0x60]
0x110e00:    lea     r13d, [rax - 2]
0x110e04:    mov     eax, dword ptr [rbx + 0x14]
0x110e07:    cmp     r13d, eax
0x110e0a:    cmovg   r13d, eax
0x110e0e:    cmp     r13d, ebp
0x110e11:    jle     0x110e49
0x110e49:    mov     eax, dword ptr [rbx + 0x68]
0x110e4c:    mov     rdi, r12
0x110e4f:    mov     dword ptr [rbx + 0x14], eax
0x110e52:    call    0x10a6b0
0x110e13:    nop     dword ptr [rax + rax]
0x110e57:    mov     rax, qword ptr [rsp + 8]
0x110e5c:    sub     rax, qword ptr fs:[0x28]
0x110e65:    jne     0x110e72
0x110e18:    mov     rdi, qword ptr [rbx + 8]
0x110e1c:    mov     esi, ebp
0x110e1e:    sub     esi, dword ptr [rbx + 0x2c]
0x110e21:    mov     edx, 1
0x110e26:    add     esi, 1
0x110e29:    add     ebp, 1

```



```

0x110e2c:    call    0x10c680
0x110e67:    add     rsp, 0x18
0x110e6b:    pop     rbx
0x110e6c:    pop     rbp
0x110e6d:    pop     r12
0x110e6f:    pop     r13
0x110e71:    ret
0x110e72:    call    0x1049d0
0x110e31:    mov     eax, dword ptr [rbx + 0x64]
0x110e34:    mov     rsi, qword ptr [rbx + 0x20]
0x110e38:    mov     rdi, qword ptr [rbx + 8]
0x110e3c:    lea     edx, [rax - 2]
0x110e3f:    call    0x10c600
0x110e44:    cmp     r13d, ebp
0x110e47:    jne     0x110e18
0x110da6:    nop     word ptr cs:[rax + rax]
0x110e77:    nop     word ptr [rax + rax]
0x110e80:    endbr64
0x110e84:    cmp     byte ptr [rdi + 0x6d], 0
0x110e88:    jne     0x110e90
0x110e8a:    ret
0x110e90:    xor     eax, eax
0x110e92:    jmp     0x110d70
0x110e8b:    nop     dword ptr [rax + rax]
0x110e97:    nop     word ptr [rax + rax]
0x110ea0:    endbr64
0x110ea4:    cmp     byte ptr [rdi + 0x6d], 0
0x110ea8:    jne     0x110eb0
0x110eaa:    ret
0x110eb0:    jmp     0x110760
0x110eab:    nop     dword ptr [rax + rax]
0x110eb5:    nop     word ptr cs:[rax + rax]
0x110ec0:    endbr64
0x110ec4:    push    r13
0x110ec6:    push    r12
0x110ec8:    push    rbp
0x110ec9:    push    rbx
0x110eca:    mov     rbx, rdi
0x110ecd:    sub     rsp, 0x18
0x110ed1:    movsxd  rdi, dword ptr [rdi + 0x64]
0x110ed5:    mov     rax, qword ptr fs:[0x28]
0x110ede:    mov     qword ptr [rsp + 8], rax
0x110ee3:    xor     eax, eax
0x110ee5:    call    0x11d670
0x110eea:    cmp     byte ptr [rbx + 0x6d], 0
0x110eee:    jne     0x110f10
0x110ef0:    mov     rax, qword ptr [rsp + 8]
0x110ef5:    sub     rax, qword ptr fs:[0x28]

```

```

0x110efe:    jne     0x111006
0x110f10:    lea     r13, [rsp + 7]
0x110f15:    mov     r12, rax
0x110f18:    mov     ebp, 1
0x110f1d:    mov     rdi, r13
0x110f20:    call    0x10a6a0
0x110f04:    add     rsp, 0x18
0x110f08:    pop     rbx
0x110f09:    pop     rbp
0x110f0a:    pop     r12
0x110f0c:    pop     r13
0x110f0e:    ret
0x111006:    call    0x1049d0
0x110f25:    mov     esi, dword ptr [rip + 0x193f5]
0x110f2b:    xor     edi, edi
0x110f2d:    mov     edx, esi
0x110f2f:    call    0x10a570
0x110f34:    cmp     dword ptr [rbx + 0x60], 2
0x110f38:    jle     0x110fa4
0x110fa4:    movsxd  rdx, dword ptr [rbx + 0x64]
0x110fa8:    mov     rdi, r12
0x110fab:    mov     esi, 0x20
0x110fb0:    call    0x104a50
0x110f3a:    nop     word ptr [rax + rax]
0x110fb5:    mov     rdi, qword ptr [rbx + 8]
0x110fb9:    xor     edx, edx
0x110fbb:    xor     esi, esi
0x110fbd:    call    0x10c680
0x110f40:    mov     rdi, qword ptr [rbx + 8]
0x110f44:    xor     edx, edx
0x110f46:    mov     esi, ebp
0x110f48:    add     ebp, 1
0x110f4b:    call    0x10c680
0x110fc2:    mov     edx, dword ptr [rbx + 0x64]
0x110fc5:    mov     rdi, qword ptr [rbx + 8]
0x110fc9:    mov     rsi, r12
0x110fcc:    call    0x10c600
0x110f50:    mov     rdi, qword ptr [rbx + 8]
0x110f54:    mov     esi, 0x20
0x110f59:    call    0x10c650
0x110fd1:    mov     eax, dword ptr [rbx + 0x60]
0x110fd4:    mov     rdi, qword ptr [rbx + 8]
0x110fd8:    xor     edx, edx
0x110fda:    lea     esi, [rax - 1]
0x110fdd:    call    0x10c680
0x110f5e:    mov     eax, dword ptr [rbx + 0x60]
0x110f61:    lea     edx, [rax - 1]
0x110f64:    cmp     edx, ebp

```

```

0x110f66:      jg      0x110f40
0x110fe2:      mov     edx, dword ptr [rbx + 0x64]
0x110fe5:      mov     rdi, qword ptr [rbx + 8]
0x110fe9:      mov     rsi, r12
0x110fec:      call    0x10c600
0x110f68:      cmp     eax, 2
0x110f6b:      jle     0x110fa4
0x110ff1:      mov     rdi, r12
0x110ff4:      call    0x118d10
0x110f6d:      mov     ebp, 1
0x110f72:      nop     word ptr [rax + rax]
0x110f78:      mov     eax, dword ptr [rbx + 0x64]
0x110f7b:      mov     rdi, qword ptr [rbx + 8]
0x110f7f:      mov     esi, ebp
0x110f81:      add     ebp, 1
0x110f84:      lea     edx, [rax - 1]
0x110f87:      call    0x10c680
0x110ff9:      mov     rdi, r13
0x110ffc:      call    0x10a6b0
0x110f8c:      mov     rdi, qword ptr [rbx + 8]
0x110f90:      mov     esi, 0x20
0x110f95:      call    0x10c650
0x111001:      jmp     0x110ef0
0x110f9a:      mov     eax, dword ptr [rbx + 0x60]
0x110f9d:      sub     eax, 1
0x110fa0:      cmp     eax, ebp
0x110fa2:      jg      0x110f78
0x110f78:      mov     eax, dword ptr [rbx + 0x64]
0x110f7b:      mov     rdi, qword ptr [rbx + 8]
0x110f7f:      mov     esi, ebp
0x110f81:      add     ebp, 1
0x110f84:      lea     edx, [rax - 1]
0x110f87:      call    0x10c680
0x110f0f:      nop
0x11100b:      nop     dword ptr [rax + rax]
0x111010:      endbr64
0x111014:      push    rbp
0x111015:      xor     eax, eax
0x111017:      mov     rbp, rdi
0x11101a:      call    0x110ec0
0x11101f:      xor     eax, eax
0x111021:      mov     rdi, rbp
0x111024:      call    0x110160
0x111029:      cmp     byte ptr [rbp + 0x6d], 0
0x11102d:      je      0x111039
0x111039:      xor     eax, eax
0x11103b:      mov     rdi, rbp
0x11103e:      call    0x110240

```

```

0x11102f:    mov     rdi, rbp
0x111032:    xor     eax, eax
0x111034:    call    0x10ec50
0x111043:    cmp     byte ptr [rbp + 0x6d], 0
0x111047:    jne     0x111050
0x111049:    pop     rbp
0x11104a:    ret
0x111050:    mov     rdi, rbp
0x111053:    xor     eax, eax
0x111055:    pop     rbp
0x111056:    jmp     0x110d70
0x11104b:    nop     dword ptr [rax + rax]
0x11105b:    nop     dword ptr [rax + rax]
0x111060:    endbr64
0x111064:    mov     edx, dword ptr [rdi + 0x68]
0x111067:    test    edx, edx
0x111069:    jle     0x1110ab
0x1110ab:    ret
0x11106b:    sub     edx, 1
0x11106e:    mov     rcx, qword ptr [rdi + 0x38]
0x111072:    lea     rdx, [rdx + rdx*4]
0x111076:    shl     rdx, 4
0x11107a:    lea     rax, [rcx + 0x49]
0x11107e:    lea     rdx, [rcx + rdx + 0x99]
0x111086:    nop     word ptr cs:[rax + rax]
0x111090:    cmp     byte ptr [rax], 0
0x111093:    jne     0x1110a2
0x111095:    cmp     byte ptr [rax - 1], 0
0x111099:    je      0x1110a2
0x1110a2:    add     rax, 0x50
0x1110a6:    cmp     rax, rdx
0x1110a9:    jne     0x111090
0x11109b:    mov     byte ptr [rax], 1
0x11109e:    add     dword ptr [rdi + 0x70], 1
0x1110a2:    add     rax, 0x50
0x1110a6:    cmp     rax, rdx
0x1110a9:    jne     0x111090
0x111090:    cmp     byte ptr [rax], 0
0x111093:    jne     0x1110a2
0x1110ac:    nop     dword ptr [rax]
0x1110b0:    endbr64
0x1110b4:    mov     edx, dword ptr [rdi + 0x68]
0x1110b7:    test    edx, edx
0x1110b9:    jle     0x1110ec
0x1110ec:    mov     dword ptr [rdi + 0x70], 0
0x1110f3:    ret
0x1110bb:    sub     edx, 1
0x1110be:    mov     rcx, qword ptr [rdi + 0x38]

```

```

0x1110c2:    lea     rdx, [rdx + rdx*4]
0x1110c6:    shl     rdx, 4
0x1110ca:    lea     rax, [rcx + 0x49]
0x1110ce:    lea     rdx, [rcx + rdx + 0x99]
0x1110d6:    nop     word ptr cs:[rax + rax]
0x1110e0:    mov     byte ptr [rax], 0
0x1110e3:    add     rax, 0x50
0x1110e7:    cmp     rax, rdx
0x1110ea:    jne     0x1110e0
0x1110e0:    mov     byte ptr [rax], 0
0x1110e3:    add     rax, 0x50
0x1110e7:    cmp     rax, rdx
0x1110ea:    jne     0x1110e0
0x1110f4:    nop     word ptr cs:[rax + rax]
0x1110ff:    nop
0x111100:    endbr64
0x111104:    mov     rax, qword ptr [rdi + 0x18]
0x111108:    ret
0x111109:    nop     dword ptr [rax]
0x111110:    endbr64
0x111114:    mov     eax, dword ptr [rip + 0x1a996]
0x11111a:    test    eax, eax
0x11111c:    jne     0x111120
0x11111e:    ret
0x111120:    push    rbp
0x111121:    xor     eax, eax
0x111123:    push    rbx
0x111124:    sub     rsp, 8
0x111128:    mov     dword ptr [rip + 0x1a97e], 0
0x111132:    call    0x10c8c0
0x111137:    mov     edx, 4
0x11113c:    lea     rsi, [rip + 0xf3ef]
0x111143:    lea     rdi, [rip + 0x100af]
0x11114a:    mov     rbp, rax
0x11114d:    call    0x108070
0x111152:    mov     rdi, rbp
0x111155:    mov     ebx, eax
0x111157:    call    0x10c970
0x11115c:    xor     eax, eax
0x11115e:    and     ebx, 0xffffffff
0x111161:    call    0x10d870
0x111166:    xor     eax, eax
0x111168:    call    0x10d740
0x11116d:    xor     eax, eax
0x11116f:    call    0x10a820
0x111174:    xor     eax, eax
0x111176:    cmp     bl, 0x4e
0x111179:    setne   al

```

```

0x11117c:      add     rsp, 8
0x111180:      pop     rbx
0x111181:      pop     rbp
0x111182:      ret
0x11111f:      nop
0x111183:      nop     word ptr cs:[rax + rax]
0x11118e:      nop
0x111190:      endbr64
0x111194:      push    rbp
0x111195:      mov     rbp, rsi
0x111198:      sub     rsp, 0x130
0x11119f:      mov     rax, qword ptr fs:[0x28]
0x1111a8:      mov     qword ptr [rsp + 0x128], rax
0x1111b0:      xor     eax, eax
0x1111b2:      mov     rsi, rsp
0x1111b5:      call    0x118f40
0x1111ba:      test    eax, eax
0x1111bc:      je      0x1111e0
0x1111e0:      lea     rsi, [rsp + 0x90]
0x1111e8:      mov     rdi, rbp
0x1111eb:      call    0x118f40
0x1111be:      xor     eax, eax
0x1111c0:      mov     rdx, qword ptr [rsp + 0x128]
0x1111c8:      sub     rdx, qword ptr fs:[0x28]
0x1111d1:      jne     0x111217
0x1111f0:      test    eax, eax
0x1111f2:      jne     0x1111be
0x1111d3:      add     rsp, 0x130
0x1111da:      pop     rbp
0x1111db:      ret
0x111217:      call    0x1049d0
0x1111f4:      mov     rax, qword ptr [rsp + 0x90]
0x1111fc:      cmp     qword ptr [rsp], rax
0x111200:      jne     0x1111be
0x111202:      mov     rax, qword ptr [rsp + 0x98]
0x11120a:      cmp     qword ptr [rsp + 8], rax
0x11120f:      sete    al
0x111212:      movzx   eax, al
0x111215:      jmp     0x1111c0
0x1111c0:      mov     rdx, qword ptr [rsp + 0x128]
0x1111c8:      sub     rdx, qword ptr fs:[0x28]
0x1111d1:      jne     0x111217
0x1111dc:      nop     dword ptr [rax]
0x11121c:      nop     dword ptr [rax]
0x111220:      endbr64
0x111224:      push    rbx
0x111225:      mov     edx, dword ptr [rdi + 0x70]
0x111228:      mov     rbx, rdi

```

```

0x11122b:    mov     ecx, 0x1e
0x111230:    test    edx, edx
0x111232:    je      0x111268
0x111268:    lea     rdx, [rip + 0xffa9]
0x11126f:    lea     rdi, [rip + 0x107da]
0x111276:    xor     eax, eax
0x111278:    call    0x10f760
0x111234:    lea     rdx, [rip + 0xffd8]
0x11123b:    lea     rdi, [rip + 0x107de]
0x111242:    xor     eax, eax
0x111244:    call    0x10f760
0x11127d:    jmp     0x111249
0x111249:    xor     r8d, r8d
0x11124c:    cmp     al, 0x6f
0x11124e:    je      0x11125e
0x11125e:    mov     eax, r8d
0x111261:    pop     rbx
0x111262:    ret
0x111250:    cmp     al, 0x73
0x111252:    je      0x111280
0x111280:    cmp     dword ptr [rbx + 0x70], 1
0x111284:    mov     r8d, 2
0x11128a:    pop     rbx
0x11128b:    sbb     r8d, -1
0x11128f:    mov     eax, r8d
0x111292:    ret
0x111254:    mov     r8d, 2
0x11125a:    cmp     al, 0x61
0x11125c:    je      0x111298
0x111298:    mov     eax, dword ptr [rbx + 0x70]
0x11129b:    test    eax, eax
0x11129d:    je      0x11125e
0x11129f:    mov     dword ptr [rbx + 0x90], 0
0x1112a9:    xor     r8d, r8d
0x1112ac:    pop     rbx
0x1112ad:    mov     eax, r8d
0x1112b0:    ret
0x111263:    nop     dword ptr [rax + rax]
0x11127f:    nop
0x111293:    nop     dword ptr [rax + rax]
0x1112b1:    nop     word ptr cs:[rax + rax]
0x1112bc:    nop     dword ptr [rax]
0x1112c0:    endbr64
0x1112c4:    push    rbp
0x1112c5:    mov     rbp, rdi
0x1112c8:    sub     rsp, 0xa0
0x1112cf:    mov     rax, qword ptr fs:[0x28]
0x1112d8:    mov     qword ptr [rsp + 0x98], rax

```

```

0x1112e0:    xor     eax, eax
0x1112e2:    mov     rsi, rsp
0x1112e5:    call    0x118f40
0x1112ea:    test    eax, eax
0x1112ec:    jne     0x1112fe
0x1112ee:    mov     eax, dword ptr [rsp + 0x18]
0x1112f2:    and     eax, 0xf000
0x1112f7:    cmp     eax, 0x8000
0x1112fc:    je      0x111320
0x1112fe:    mov     rax, qword ptr [rsp + 0x98]
0x111306:    sub     rax, qword ptr fs:[0x28]
0x11130f:    jne     0x11132a
0x111320:    mov     rdi, rbp
0x111323:    call    0x104870
0x111311:    add     rsp, 0xa0
0x111318:    pop     rbp
0x111319:    ret
0x11132a:    call    0x1049d0
0x111328:    jmp     0x1112fe
0x11131a:    nop     word ptr [rax + rax]
0x11132f:    nop
0x111330:    endbr64
0x111334:    push    r15
0x111336:    push    r14
0x111338:    mov     r14, rdx
0x11133b:    push    r13
0x11133d:    push    r12
0x11133f:    mov     r12d, ecx
0x111342:    push    rbp
0x111343:    and     r12d, 0xf000
0x11134a:    mov     rbp, rdi
0x11134d:    push    rbx
0x11134e:    mov     ebx, ecx
0x111350:    sub     rsp, 0x178
0x111357:    mov     qword ptr [rsp + 8], rsi
0x11135c:    mov     dword ptr [rsp + 0x30], r8d
0x111361:    mov     dword ptr [rsp + 0x34], r9d
0x111366:    mov     rax, qword ptr fs:[0x28]
0x11136f:    mov     qword ptr [rsp + 0x168], rax
0x111377:    xor     eax, eax
0x111379:    cmp     r12d, 0xa000
0x111380:    je      0x111a10
0x111a10:    mov     rdi, qword ptr [rsp + 8]
0x111a15:    lea     rsi, [rsp + 0xd0]
0x111a1d:    call    0x118f40
0x111386:    cmp     r12d, 0x4000
0x11138d:    je      0x111828
0x111a22:    cmp     eax, -1

```



```

0x111a25:      je      0x111393
0x111828:      lea     rsi, [rsp + 0x40]
0x11182d:      mov     rdi, r14
0x111830:      call    0x118f40
0x111393:      mov     rdi, r14
0x111396:      call    0x1049b0
0x111a2b:      mov     ebx, dword ptr [rsp + 0xe8]
0x111a32:      mov     r12d, ebx
0x111a35:      and     r12d, 0xf000
0x111a3c:      jmp     0x111386
0x111835:      test    eax, eax
0x111837:      jne     0x1118c3
0x11139b:      mov     rdi, r14
0x11139e:      mov     r13, rax
0x1113a1:      call    0x11d7a0
0x11183d:      mov     eax, dword ptr [rsp + 0x58]
0x111841:      mov     r13d, 0xb
0x111847:      and     eax, 0xf000
0x11184c:      cmp     eax, 0x4000
0x111851:      jne     0x111748
0x1118c3:      mov     r15, qword ptr [rsp + 8]
0x1118c8:      mov     r13d, 6
0x1118ce:      mov     rdi, r15
0x1118d1:      call    0x1049b0
0x1113a6:      lea     rsi, [rsp + 0x40]
0x1113ab:      mov     rdi, rax
0x1113ae:      mov     qword ptr [rsp + 0x18], rax
0x1113b3:      call    0x118f40
0x111857:      mov     rdi, r14
0x11185a:      call    0x1049b0
0x111748:      mov     rax, qword ptr [rsp + 0x168]
0x111750:      sub     rax, qword ptr fs:[0x28]
0x111759:      jne     0x111cfc
0x1118d6:      mov     rdi, r14
0x1118d9:      mov     rbx, rax
0x1118dc:      call    0x1049b0
0x1113b8:      test    eax, eax
0x1113ba:      je      0x111630
0x11185f:      mov     r15, qword ptr [rsp + 8]
0x111864:      mov     rbx, rax
0x111867:      mov     rdi, r15
0x11186a:      call    0x1049b0
0x11175f:      add     rsp, 0x178
0x111766:      mov     eax, r13d
0x111769:      pop     rbx
0x11176a:      pop     rbp
0x11176b:      pop     r12
0x11176d:      pop     r13

```

```

0x11176f:    pop     r14
0x111771:    pop     r15
0x111773:    ret
0x111cfc:    call    0x1049d0
0x1118e1:    lea     rdi, [rbx + rax + 0x21]
0x1118e6:    call    0x11d670
0x111630:    mov     eax, dword ptr [rsp + 0x58]
0x111634:    and     eax, 0xf000
0x111639:    cmp     eax, 0x4000
0x11163e:    jne     0x1113c0
0x1113c0:    mov     rdi, qword ptr [rsp + 0x18]
0x1113c5:    call    0x1190c0
0x11186f:    lea     rdi, [rbx + rax + 2]
0x111874:    call    0x11d670
0x1118eb:    mov     rdi, r14
0x1118ee:    mov     r12, rax
0x1118f1:    call    0x1190c0
0x111644:    mov     r14, qword ptr [rsp + 8]
0x111649:    mov     rdi, r14
0x11164c:    call    0x1049b0
0x1113ca:    cmp     byte ptr [rax], 0
0x1113cd:    mov     r13, rax
0x1113d0:    je      0x111ae5
0x111879:    mov     r9, r15
0x11187c:    mov     r8, r14
0x11187f:    mov     esi, 1
0x111884:    mov     rdi, rax
0x111887:    mov     r12, rax
0x11188a:    xor     eax, eax
0x11188c:    mov     rdx, -1
0x111893:    lea     rcx, [rip + 0xee67]
0x11189a:    call    0x104f10
0x1118f6:    cmp     byte ptr [rax], 0
0x1118f9:    je      0x111748
0x111651:    mov     rdi, qword ptr [rsp + 0x18]
0x111656:    lea     rsi, [r13 + rax + 2]
0x11165b:    call    0x11d690
0x111ae5:    mov     rdi, qword ptr [rsp + 0x18]
0x111aea:    mov     r13d, 6
0x111af0:    call    0x118d10
0x1113d6:    mov     esi, dword ptr [rbp + 0x90]
0x1113dc:    test    esi, esi
0x1113de:    je      0x1113f4
0x11189f:    mov     edi, dword ptr [rbp + 0x90]
0x1118a5:    test    edi, edi
0x1118a7:    je      0x1118bb
0x1118ff:    mov     rdi, r15
0x111902:    call    0x1049b0

```

```

0x111660:    mov     rdi, rax
0x111663:    mov     r15, rax
0x111666:    mov     qword ptr [rsp + 0x18], rax
0x11166b:    call    0x1049b0
0x111af5:    jmp     0x111748
0x1113f4:    mov     rdi, qword ptr [rsp + 8]
0x1113f9:    call    0x1049b0
0x1113e0:    mov     rdi, qword ptr [rsp + 0x18]
0x1113e5:    xor     esi, esi
0x1113e7:    call    0x104d90
0x1118bb:    mov     rdi, r12
0x1118be:    call    0x118d10
0x1118a9:    xor     esi, esi
0x1118ab:    mov     rdi, r12
0x1118ae:    call    0x104d90
0x111907:    mov     rdi, r14
0x11190a:    mov     rbx, rax
0x11190d:    call    0x1049b0
0x111670:    mov     rsi, r14
0x111673:    mov     byte ptr [r15 + rax], 0x2f
0x111678:    lea     rdi, [r15 + rax + 1]
0x11167d:    call    0x1048b0
0x1113fe:    lea     rdi, [rax + 0x20]
0x111402:    call    0x11d670
0x1113ec:    test    eax, eax
0x1113ee:    je      0x111a60
0x1118b3:    test    eax, eax
0x1118b5:    je      0x111cbb
0x111912:    lea     rdi, [rbx + rax + 0x20]
0x111917:    call    0x11d670
0x111682:    jmp     0x1113c0
0x111407:    mov     r8, qword ptr [rsp + 8]
0x11140c:    lea     rcx, [rip + 0xfe34]
0x111413:    mov     r13, rax
0x111416:    cmp     r12d, 0x8000
0x11141d:    je      0x111a48
0x111a60:    call    0x10e4e0
0x111cbb:    call    0x10e4e0
0x11191c:    mov     r9, r14
0x11191f:    mov     r8, r15
0x111922:    mov     esi, 1
0x111927:    lea     rcx, [rip + 0xf8ed]
0x11192e:    mov     r13, rax
0x111931:    mov     rdi, rax
0x111934:    xor     eax, eax
0x111936:    mov     rdx, -1
0x11193d:    call    0x104f10
0x111a48:    mov     r8, qword ptr [rsp + 8]

```

```

0x111a4d:    lea     rcx, [rip + 0xf7e1]
0x111a54:    jmp     0x111423
0x111423:    mov     rdx, -1
0x11142a:    mov     esi, 1
0x11142f:    mov     rdi, r13
0x111432:    xor     eax, eax
0x111434:    call    0x104f10
0x111a65:    mov     rsi, r13
0x111a68:    xor     eax, eax
0x111a6a:    mov     rdi, rbp
0x111a6d:    call    0x111220
0x111cc0:    mov     rdi, rbp
0x111cc3:    mov     rsi, r12
0x111cc6:    xor     eax, eax
0x111cc8:    call    0x111220
0x111942:    mov     edx, 1
0x111947:    mov     esi, 1
0x11194c:    mov     rdi, r13
0x11194f:    call    0x10e480
0x111439:    mov     edx, 1
0x11143e:    mov     esi, 1
0x111443:    mov     rdi, r13
0x111446:    call    0x10e480
0x111a72:    mov     r13d, eax
0x111a75:    xor     eax, eax
0x111a77:    call    0x10a820
0x111ccd:    mov     r13d, eax
0x111cd0:    xor     eax, eax
0x111cd2:    call    0x10a820
0x111954:    xor     eax, eax
0x111956:    call    0x10a820
0x11144b:    xor     eax, eax
0x11144d:    call    0x10a820
0x111a7c:    test    r13d, r13d
0x111a7f:    je      0x1113f4
0x111cd7:    mov     rdi, r12
0x111cda:    call    0x118d10
0x11195b:    mov     rdi, r13
0x11195e:    mov     r13d, 0xc
0x111964:    call    0x118d10
0x111452:    mov     rdi, r13
0x111455:    call    0x118d10
0x111a85:    jmp     0x111748
0x111cdf:    test    r13d, r13d
0x111ce2:    je      0x1118c3
0x111969:    mov     r9, r14
0x11196c:    mov     r8, r15
0x11196f:    mov     esi, 1

```

```

0x111974:    lea     rcx, [rip + 0xf8a7]
0x11197b:    mov     rdx, -1
0x111982:    mov     rdi, r12
0x111985:    xor     eax, eax
0x111987:    call    0x104f10
0x11145a:    mov     rdi, qword ptr [rsp + 8]
0x11145f:    xor     esi, esi
0x111461:    xor     eax, eax
0x111463:    call    0x104d30
0x111ce8:    jmp     0x111748
0x11198c:    mov     esi, 1
0x111991:    mov     rdi, r12
0x111994:    call    0x1193f0
0x111468:    mov     r15d, eax
0x11146b:    cmp     eax, -1
0x11146e:    je      0x111a90
0x111999:    mov     rdi, r12
0x11199c:    mov     ebx, eax
0x11199e:    call    0x118d10
0x111a90:    mov     r13d, 4
0x111a96:    jmp     0x111748
0x111474:    mov     ecx, dword ptr [rbp + 0x10]
0x111477:    test    ecx, ecx
0x111479:    je      0x11147e
0x1119a3:    mov     rdi, qword ptr [rbp + 0x18]
0x1119a7:    call    0x10c4b0
0x11147e:    mov     rdi, qword ptr [rsp + 0x18]
0x111483:    mov     esi, ebx
0x111485:    call    0x104e00
0x11147b:    and     ebx, 0xffffffffb6
0x11147e:    mov     rdi, qword ptr [rsp + 0x18]
0x111483:    mov     esi, ebx
0x111485:    call    0x104e00
0x1119ac:    mov     eax, ebx
0x1119ae:    and     eax, 0x7f
0x1119b1:    add     eax, 1
0x1119b4:    cmp     al, 1
0x1119b6:    jg      0x111748
0x11148a:    mov     edi, r15d
0x11148d:    mov     dword ptr [rsp + 0x28], eax
0x111491:    cmp     eax, -1
0x111494:    je      0x111ae0
0x1119bc:    mov     r13d, 1
0x1119c2:    test    bl, 0x7f
0x1119c5:    jne     0x111748
0x111ae0:    call    0x104ab0
0x11149a:    call    0x1182f0
0x1119cb:    and     bh, 0xff

```

```

0x1119ce:      je      0x111748
0x11149f:      mov     qword ptr [rsp + 0x10], 0x8000
0x1114a8:      mov     r13, rax
0x1114ab:      cmp     rax, 0x8000
0x1114b1:      jle     0x111b38
0x1119d4:      lea     rdi, [rip + 0xf857]
0x1119db:      mov     r13d, 9
0x1119e1:      call    0x1195e0
0x111b38:      mov     edi, r15d
0x111b3b:      call    0x1182f0
0x1114b7:      mov     eax, ebx
0x1114b9:      and     ebx, 0xb000
0x1114bf:      and     eax, 0xf000
0x1114c4:      mov     dword ptr [rsp + 0x2c], eax
0x1114c8:      cmp     ebx, 0x2000
0x1114ce:      je      0x111b00
0x1119e6:      jmp     0x111748
0x111b40:      mov     qword ptr [rsp + 0x10], rax
0x111b45:      mov     r13, rax
0x111b48:      jmp     0x1114b7
0x111b00:      movabs  r13, 0x7fffffffffffffff
0x111b0a:      mov     edi, 0x8000
0x111b0f:      call    0x11d670
0x1114d4:      test    r13, r13
0x1114d7:      je      0x111c58
0x111b14:      mov     qword ptr [rsp + 0x10], 0x8000
0x111b1d:      mov     r14, rax
0x111b20:      movabs  rax, 0x147ae147ae147ae
0x111b2a:      mov     qword ptr [rsp + 0x38], rax
0x111b2f:      jmp     0x111519
0x111c58:      call    0x1049e0
0x1114dd:      mov     rdi, qword ptr [rsp + 0x10]
0x1114e2:      call    0x11d670
0x111519:      xor     r12d, r12d
0x11151c:      jmp     0x111530
0x111c5d:      test    eax, eax
0x111c5f:      je      0x111ca7
0x1114e7:      mov     r14, rax
0x1114ea:      test    r13, r13
0x1114ed:      jle     0x111aa0
0x111530:      mov     rax, qword ptr [rsp + 0x10]
0x111535:      mov     rbp, r13
0x111538:      mov     edx, dword ptr [rip + 0x1a572]
0x11153e:      sub     rbp, r12
0x111541:      cmp     rbp, rax
0x111544:      cmovg   rbp, rax
0x111548:      test    edx, edx
0x11154a:      jne     0x111690

```

```

0x111ca7:    mov     edx, dword ptr [rsp + 0x34]
0x111cab:    mov     esi, dword ptr [rsp + 0x30]
0x111caf:    mov     rdi, qword ptr [rsp + 0x18]
0x111cb4:    call    0x104cc0
0x111c61:    mov     edi, r15d
0x111c64:    mov     r13d, 1
0x111c6a:    call    0x104ab0
0x111aa0:    call    0x1049e0
0x1114f3:    movabs  rdx, 0xa3d70a3d70a3d70b
0x1114fd:    mov     rax, r13
0x111500:    imul    rdx
0x111503:    mov     rax, r13
0x111506:    sar     rax, 0x3f
0x11150a:    add     rdx, r13
0x11150d:    sar     rdx, 6
0x111511:    sub     rdx, rax
0x111514:    mov     qword ptr [rsp + 0x38], rdx
0x111519:    xor     r12d, r12d
0x11151c:    jmp     0x111530
0x111550:    mov     edi, 1
0x111555:    call    0x118b70
0x111690:    mov     dword ptr [rip + 0x1a416], 0
0x11169a:    xor     eax, eax
0x11169c:    call    0x10c8c0
0x111cb9:    jmp     0x111c61
0x111c6f:    mov     edi, dword ptr [rsp + 0x28]
0x111c73:    call    0x104ab0
0x111aa5:    test    eax, eax
0x111aa7:    je      0x111c90
0x11155a:    mov     edi, r15d
0x11155d:    movsxd  rdx, ebp
0x111560:    mov     rsi, r14
0x111563:    call    0x118d30
0x1116a1:    mov     edx, 4
0x1116a6:    lea     rsi, [rip + 0xee85]
0x1116ad:    lea     rdi, [rip + 0xfb45]
0x1116b4:    mov     qword ptr [rsp + 0x20], rax
0x1116b9:    call    0x108070
0x111c78:    mov     rdi, qword ptr [rsp + 0x18]
0x111c7d:    call    0x118d10
0x111c90:    mov     edx, dword ptr [rsp + 0x34]
0x111c94:    mov     esi, dword ptr [rsp + 0x30]
0x111c98:    mov     rdi, qword ptr [rsp + 0x18]
0x111c9d:    call    0x104cc0
0x111aad:    mov     edi, r15d
0x111ab0:    mov     r13d, 1
0x111ab6:    call    0x104ab0
0x111568:    xor     edi, edi

```

```

0x11156a:    mov     ebx, eax
0x11156c:    call   0x118b70
0x1116be:    mov     rdi, qword ptr [rsp + 0x20]
0x1116c3:    mov     ebx, eax
0x1116c5:    call   0x10c970
0x111c82:    jmp     0x111748
0x111ca2:    jmp     0x111aad
0x111abb:    mov     edi, dword ptr [rsp + 0x28]
0x111abf:    call   0x104ab0
0x111571:    cmp     ebp, ebx
0x111573:    je      0x111583
0x1116ca:    xor     eax, eax
0x1116cc:    and     ebx, 0xffffffff
0x1116cf:    call   0x10d870
0x111ac4:    mov     rdi, r14
0x111ac7:    call   0x118d10
0x111583:    mov     eax, dword ptr [rip + 0x1a527]
0x111589:    test    eax, eax
0x11158b:    jne     0x111778
0x111575:    test    ebx, ebx
0x111577:    js      0x111bf0
0x1116d4:    xor     eax, eax
0x1116d6:    call   0x10d740
0x111acc:    mov     rdi, qword ptr [rsp + 0x18]
0x111ad1:    call   0x118d10
0x111591:    movsxd  rbp, ebx
0x111594:    mov     edi, dword ptr [rsp + 0x28]
0x111598:    mov     rsi, r14
0x11159b:    mov     rdx, rbp
0x11159e:    call   0x118da0
0x111778:    mov     dword ptr [rip + 0x1a32e], 0
0x111782:    xor     eax, eax
0x111784:    call   0x10c8c0
0x111bf0:    mov     edi, r15d
0x111bf3:    call   0x104ab0
0x11157d:    je      0x111aa0
0x1116db:    xor     eax, eax
0x1116dd:    call   0x10a820
0x111ad6:    jmp     0x111748
0x1115a3:    cmp     ebx, eax
0x1115a5:    jne     0x111b50
0x111789:    mov     edx, 4
0x11178e:    lea     rsi, [rip + 0xed9d]
0x111795:    lea     rdi, [rip + 0xfa5d]
0x11179c:    mov     qword ptr [rsp + 0x20], rax
0x1117a1:    call   0x108070
0x111bf8:    mov     edi, dword ptr [rsp + 0x28]
0x111bfc:    call   0x104ab0

```



```

0x1116e2:    cmp     bl, 0x4e
0x1116e5:    je      0x111550
0x1115ab:    add     rbp, r12
0x1115ae:    cmp     rbp, r13
0x1115b1:    jg      0x111520
0x111b50:    mov     dword ptr [rsp + 8], eax
0x111b54:    call    0x104860
0x1117a6:    mov     rdi, qword ptr [rsp + 0x20]
0x1117ab:    mov     ebp, eax
0x1117ad:    call    0x10c970
0x111c01:    mov     rdi, qword ptr [rsp + 0x18]
0x111c06:    lea     rsi, [rsp + 0xd0]
0x111c0e:    call    0x118f40
0x1116eb:    mov     edi, r15d
0x1116ee:    call    0x104ab0
0x1115b7:    mov     rdi, qword ptr [rsp + 8]
0x1115bc:    call    0x1049b0
0x111520:    add     r12, 0x8000
0x111527:    cmp     r12, r13
0x11152a:    jge     0x111aa0
0x111b59:    mov     edi, r15d
0x111b5c:    mov     ebx, dword ptr [rax]
0x111b5e:    call    0x104ab0
0x1117b2:    xor     eax, eax
0x1117b4:    and     ebp, 0xfffffffff
0x1117b7:    call    0x10d870
0x111c13:    test    eax, eax
0x111c15:    jne     0x111c38
0x1116f3:    mov     edi, dword ptr [rsp + 0x28]
0x1116f7:    call    0x104ab0
0x1115c1:    lea     rdi, [rax + 0x20]
0x1115c5:    call    0x11d670
0x111b63:    mov     edi, dword ptr [rsp + 0x28]
0x111b67:    call    0x104ab0
0x1117bc:    xor     eax, eax
0x1117be:    call    0x10d740
0x111c17:    mov     eax, dword ptr [rsp + 0xe8]
0x111c1e:    and     eax, 0xf000
0x111c23:    cmp     eax, 0x8000
0x111c28:    jne     0x111c38
0x111c38:    mov     rdi, r14
0x111c3b:    mov     r13d, 5
0x111c41:    call    0x118d10
0x1116fc:    mov     rdi, qword ptr [rsp + 0x18]
0x111701:    lea     rsi, [rsp + 0xd0]
0x111709:    call    0x118f40
0x1115ca:    cmp     dword ptr [rsp + 0x2c], 0x8000
0x1115d2:    mov     rbx, rax

```

```

0x1115d5:    je      0x1117e0
0x111b6c:    mov     rdi, qword ptr [rsp + 0x18]
0x111b71:    lea     rsi, [rsp + 0xd0]
0x111b79:    call    0x118f40
0x1117c3:    xor     eax, eax
0x1117c5:    call    0x10a820
0x111c2a:    mov     rdi, qword ptr [rsp + 0x18]
0x111c2f:    call    0x104870
0x111c46:    mov     rdi, qword ptr [rsp + 0x18]
0x111c4b:    call    0x118d10
0x11170e:    test    eax, eax
0x111710:    jne     0x111729
0x1117e0:    cmp     r13, 0x1869f
0x1117e7:    jle     0x1119f0
0x1115db:    mov     r9, qword ptr [rsp + 8]
0x1115e0:    mov     rdi, rax
0x1115e3:    mov     r8, rbp
0x1115e6:    mov     esi, 1
0x1115eb:    lea     rcx, [rip + 0xfc7b]
0x1115f2:    mov     rdx, -1
0x1115f9:    xor     eax, eax
0x1115fb:    call    0x104f10
0x111b7e:    mov     edx, dword ptr [rsp + 8]
0x111b82:    test    eax, eax
0x111b84:    jne     0x111bb0
0x1117ca:    cmp     bpl, 0x4e
0x1117ce:    je      0x111591
0x111c34:    nop     dword ptr [rax]
0x111c50:    jmp     0x111748
0x111712:    mov     eax, dword ptr [rsp + 0xe8]
0x111719:    and     eax, 0xf000
0x11171e:    cmp     eax, 0x8000
0x111723:    je      0x111ced
0x111729:    mov     rdi, r14
0x11172c:    mov     r13d, 2
0x111732:    call    0x118d10
0x1119f0:    lea     rax, [rbp + rbp*4]
0x1119f5:    lea     r8, [rax + rax*4]
0x1119f9:    shl     r8, 2
0x1119fd:    mov     rax, r8
0x111a00:    cqo
0x111a02:    idiv    r13
0x111a05:    mov     r8, rax
0x111a08:    jmp     0x1117fa
0x1117ed:    mov     rax, rbp
0x1117f0:    cqo
0x1117f2:    idiv    qword ptr [rsp + 0x38]
0x1117f7:    mov     r8, rax

```

```

0x1117fa:    mov     r9, qword ptr [rsp + 8]
0x1117ff:    mov     esi, 1
0x111804:    mov     rdi, rbx
0x111807:    xor     eax, eax
0x111809:    lea     rcx, [rip + 0xfa4b]
0x111810:    mov     rdx, -1
0x111817:    call    0x104f10
0x111600:    mov     rdi, rbx
0x111603:    mov     edx, 1
0x111608:    mov     esi, 1
0x11160d:    call    0x10e480
0x111b86:    mov     eax, dword ptr [rsp + 0xe8]
0x111b8d:    and     eax, 0xf000
0x111b92:    cmp     eax, 0x8000
0x111b97:    jne     0x111bb0
0x111bb0:    mov     rdi, r14
0x111bb3:    mov     dword ptr [rsp + 8], edx
0x111bb7:    call    0x118d10
0x1117d4:    jmp     0x1116eb
0x111ced:    mov     rdi, qword ptr [rsp + 0x18]
0x111cf2:    call    0x104870
0x111737:    mov     rdi, qword ptr [rsp + 0x18]
0x11173c:    call    0x118d10
0x1117fa:    mov     r9, qword ptr [rsp + 8]
0x1117ff:    mov     esi, 1
0x111804:    mov     rdi, rbx
0x111807:    xor     eax, eax
0x111809:    lea     rcx, [rip + 0xfa4b]
0x111810:    mov     rdx, -1
0x111817:    call    0x104f10
0x11181c:    jmp     0x111600
0x111612:    xor     eax, eax
0x111614:    call    0x10a820
0x111b99:    mov     rdi, qword ptr [rsp + 0x18]
0x111b9e:    call    0x104870
0x111bbc:    mov     rdi, qword ptr [rsp + 0x18]
0x111bc1:    call    0x118d10
0x111cf7:    jmp     0x111729
0x111741:    nop     dword ptr [rax]
0x111619:    mov     rdi, rbx
0x11161c:    call    0x118d10
0x111ba3:    mov     edx, dword ptr [rsp + 8]
0x111ba7:    nop     word ptr [rax + rax]
0x111bb0:    mov     rdi, r14
0x111bb3:    mov     dword ptr [rsp + 8], edx
0x111bb7:    call    0x118d10
0x111bc6:    mov     edx, dword ptr [rsp + 8]
0x111bca:    cmp     ebx, 0x1c

```

```

0x111bcd:    sete    r13b
0x111bd1:    test    edx, edx
0x111bd3:    setg    al
0x111bd6:    or      r13d, eax
0x111bd9:    movzx   r13d, r13b
0x111bdd:    add     r13d, 7
0x111be1:    jmp     0x111748
0x111621:    jmp     0x111520
0x11151e:    nop
0x111626:    nop     word ptr cs:[rax + rax]
0x111687:    nop     word ptr [rax + rax]
0x111774:    nop     dword ptr [rax]
0x1117d9:    nop     dword ptr [rax]
0x111821:    nop     dword ptr [rax]
0x1119eb:    nop     dword ptr [rax + rax]
0x111a0d:    nop     dword ptr [rax]
0x111a41:    nop     dword ptr [rax]
0x111a59:    nop     dword ptr [rax]
0x111a8a:    nop     word ptr [rax + rax]
0x111a9b:    nop     dword ptr [rax + rax]
0x111adb:    nop     dword ptr [rax + rax]
0x111afa:    nop     word ptr [rax + rax]
0x111b34:    nop     dword ptr [rax]
0x111b4d:    nop     dword ptr [rax]
0x111be6:    nop     word ptr cs:[rax + rax]
0x111c55:    nop     dword ptr [rax]
0x111c87:    nop     word ptr [rax + rax]
0x111d01:    nop     word ptr cs:[rax + rax]
0x111d0c:    nop     dword ptr [rax]
0x111d10:    endbr64
0x111d14:    push    r15
0x111d16:    and     ecx, 0xf000
0x111d1c:    push    r14
0x111d1e:    mov     r14, rdi
0x111d21:    push    r13
0x111d23:    push    r12
0x111d25:    mov     r12, rdx
0x111d28:    push    rbp
0x111d29:    mov     rbp, rsi
0x111d2c:    push    rbx
0x111d2d:    sub     rsp, 0x148
0x111d34:    mov     rax, qword ptr fs:[0x28]
0x111d3d:    mov     qword ptr [rsp + 0x138], rax
0x111d45:    xor     eax, eax
0x111d47:    cmp     ecx, 0x4000
0x111d4d:    je      0x111f20
0x111f20:    lea     rsi, [rsp + 0x10]
0x111f25:    mov     rdi, rdx

```

```

0x111f28:    call    0x118f40
0x111d53:    mov     rdi, rdx
0x111d56:    lea     rbx, [rsp + 0x10]
0x111d5b:    call    0x1049b0
0x111f2d:    test    eax, eax
0x111f2f:    jne     0x111fb7
0x111d60:    mov     rdi, r12
0x111d63:    mov     r13, rax
0x111d66:    call    0x11d7a0
0x111f35:    mov     eax, dword ptr [rsp + 0x28]
0x111f39:    mov     r13d, 9
0x111f3f:    and     eax, 0xf000
0x111f44:    cmp     eax, 0x4000
0x111f49:    jne     0x111ea1
0x111fb7:    mov     rdi, rbp
0x111fba:    call    0x1049b0
0x111d6b:    mov     rsi, rbx
0x111d6e:    mov     rdi, rax
0x111d71:    mov     r12, rax
0x111d74:    call    0x118f40
0x111f4f:    mov     rdi, r12
0x111f52:    call    0x1049b0
0x111ea1:    mov     rax, qword ptr [rsp + 0x138]
0x111ea9:    sub     rax, qword ptr fs:[0x28]
0x111eb2:    jne     0x1122c1
0x111fbf:    mov     rdi, r12
0x111fc2:    mov     rbx, rax
0x111fc5:    call    0x1049b0
0x111d79:    test    eax, eax
0x111d7b:    je      0x111ed0
0x111f57:    mov     rdi, rbp
0x111f5a:    mov     rbx, rax
0x111f5d:    call    0x1049b0
0x111eb8:    add     rsp, 0x148
0x111ebf:    mov     eax, r13d
0x111ec2:    pop     rbx
0x111ec3:    pop     rbp
0x111ec4:    pop     r12
0x111ec6:    pop     r13
0x111ec8:    pop     r14
0x111eca:    pop     r15
0x111ecc:    ret
0x1122c1:    call    0x1049d0
0x111fca:    lea     rdi, [rbx + rax + 0x21]
0x111fcf:    call    0x11d670
0x111ed0:    mov     eax, dword ptr [rsp + 0x28]
0x111ed4:    and     eax, 0xf000
0x111ed9:    cmp     eax, 0x4000

```

```

0x111ede:    jne     0x111d81
0x111d81:    mov     rdi, r12
0x111d84:    call    0x1190c0
0x111f62:    lea     rdi, [rbx + rax + 2]
0x111f67:    call    0x11d670
0x111fd4:    mov     rdi, r12
0x111fd7:    mov     r13, rax
0x111fda:    call    0x1190c0
0x111ee4:    mov     rdi, rbp
0x111ee7:    call    0x1049b0
0x111d89:    cmp     byte ptr [rax], 0
0x111d8c:    mov     r15, rax
0x111d8f:    je      0x112170
0x111f6c:    mov     esi, 1
0x111f71:    mov     r9, rbp
0x111f74:    mov     r8, r12
0x111f77:    mov     rdi, rax
0x111f7a:    mov     r15, rax
0x111f7d:    xor     eax, eax
0x111f7f:    mov     rdx, -1
0x111f86:    lea     rcx, [rip + 0xe774]
0x111f8d:    call    0x104f10
0x111fdf:    cmp     byte ptr [rax], 0
0x111fe2:    je      0x112178
0x111eec:    mov     rdi, r12
0x111eef:    lea     rsi, [r13 + rax + 2]
0x111ef4:    call    0x11d690
0x112170:    mov     rdi, r12
0x112173:    call    0x118d10
0x111d95:    cmp     r12, rax
0x111d98:    je      0x1120e8
0x111f92:    mov     esi, dword ptr [r14 + 0x90]
0x111f99:    test    esi, esi
0x111f9b:    je      0x111faf
0x112178:    mov     r13d, 4
0x11217e:    jmp     0x111ea1
0x111fe8:    mov     rdi, rbp
0x111feb:    call    0x1049b0
0x111ef9:    mov     rdi, rax
0x111efc:    mov     r12, rax
0x111eff:    call    0x1049b0
0x1120e8:    mov     rdi, r12
0x1120eb:    call    0x1049b0
0x111d9e:    mov     edx, dword ptr [r14 + 0x90]
0x111da5:    test    edx, edx
0x111da7:    je      0x111dbb
0x111faf:    mov     rdi, r15
0x111fb2:    call    0x118d10

```

```

0x111f9d:    xor     esi, esi
0x111f9f:    mov     rdi, r15
0x111fa2:    call    0x104d90
0x111ff0:    mov     rdi, r12
0x111ff3:    mov     rbx, rax
0x111ff6:    call    0x1049b0
0x111f04:    mov     rsi, rbp
0x111f07:    mov     byte ptr [r12 + rax], 0x2f
0x111f0c:    lea     rdi, [r12 + rax + 1]
0x111f11:    call    0x1048b0
0x1120f0:    lea     rdi, [rax + 3]
0x1120f4:    mov     r13, rax
0x1120f7:    call    0x11d670
0x111dbb:    mov     rdi, rbp
0x111dbe:    call    0x1049b0
0x111da9:    xor     esi, esi
0x111dab:    mov     rdi, r12
0x111dae:    call    0x104d90
0x111fa7:    test    eax, eax
0x111fa9:    je      0x112278
0x111ffb:    lea     rdi, [rbx + rax + 0x20]
0x112000:    call    0x11d670
0x111f16:    jmp     0x111d81
0x1120fc:    mov     ecx, 0x2f2e
0x112101:    lea     rdx, [r13 + 1]
0x112105:    mov     rsi, r15
0x112108:    mov     word ptr [rax], cx
0x11210b:    lea     rcx, [rax + 2]
0x11210f:    mov     r12, rax
0x112112:    mov     rdi, rcx
0x112115:    mov     qword ptr [rsp + 8], rcx
0x11211a:    call    0x104bb0
0x111dc3:    lea     rdi, [rax + 0x20]
0x111dc7:    call    0x11d670
0x111db3:    test    eax, eax
0x111db5:    je      0x112140
0x112278:    call    0x10e4e0
0x112005:    mov     r9, r12
0x112008:    mov     r8, rbp
0x11200b:    mov     esi, 1
0x112010:    lea     rcx, [rip + 0xf26c]
0x112017:    mov     r15, rax
0x11201a:    mov     rdi, rax
0x11201d:    xor     eax, eax
0x11201f:    mov     rdx, -1
0x112026:    call    0x104f10
0x11211f:    mov     rdi, r15
0x112122:    call    0x118d10

```

```

0x111dcc:    mov     r8, rbp
0x111dcf:    mov     esi, 1
0x111dd4:    lea     rcx, [rip + 0xf4c2]
0x111ddb:    mov     r13, rax
0x111dde:    mov     rdi, rax
0x111de1:    mov     rdx, -1
0x111de8:    xor     eax, eax
0x111dea:    call    0x104f10
0x112140:    call    0x10e4e0
0x11227d:    mov     rdi, r14
0x112280:    mov     rsi, r15
0x112283:    xor     eax, eax
0x112285:    call    0x111220
0x11202b:    mov     edx, 1
0x112030:    mov     esi, 1
0x112035:    mov     rdi, r15
0x112038:    call    0x10e480
0x112127:    mov     edx, dword ptr [r14 + 0x90]
0x11212e:    mov     r15, qword ptr [rsp + 8]
0x112133:    test    edx, edx
0x112135:    jne     0x111da9
0x111def:    mov     edx, 1
0x111df4:    mov     esi, 1
0x111df9:    mov     rdi, r13
0x111dfc:    call    0x10e480
0x112145:    xor     eax, eax
0x112147:    mov     rsi, r15
0x11214a:    mov     rdi, r14
0x11214d:    call    0x111220
0x11228a:    mov     r13d, eax
0x11228d:    xor     eax, eax
0x11228f:    call    0x10a820
0x11203d:    xor     eax, eax
0x11203f:    call    0x10a820
0x11213b:    jmp     0x111dbb
0x111e01:    xor     eax, eax
0x111e03:    call    0x10a820
0x112152:    mov     r13d, eax
0x112155:    xor     eax, eax
0x112157:    call    0x10a820
0x112294:    mov     rdi, r15
0x112297:    call    0x118d10
0x112044:    mov     rdi, r15
0x112047:    call    0x118d10
0x111e08:    mov     rdi, r13
0x111e0b:    mov     r13d, 6
0x111e11:    call    0x118d10
0x11215c:    test    r13d, r13d

```



```

0x11215f:    je     0x111dbb
0x11229c:    test   r13d, r13d
0x11229f:    je     0x111fb7
0x11204c:    mov     r9, r12
0x11204f:    mov     r8, rbp
0x112052:    mov     rdi, r13
0x112055:    lea     rcx, [rip + 0xf22e]
0x11205c:    mov     rdx, -1
0x112063:    mov     esi, 1
0x112068:    xor     eax, eax
0x11206a:    call    0x104f10
0x111e16:    lea     rsi, [rsp + 0xa0]
0x111e1e:    mov     rdi, rbp
0x111e21:    call    0x118f40
0x112165:    jmp     0x111ea1
0x1122a5:    jmp     0x111ea1
0x11206f:    mov     esi, 1
0x112074:    mov     rdi, r13
0x112077:    call    0x1193f0
0x111e26:    cmp     eax, -1
0x111e29:    je     0x111ea1
0x11207c:    mov     rdi, r13
0x11207f:    mov     r13d, 0xc
0x112085:    mov     ebx, eax
0x112087:    call    0x118d10
0x111e2b:    mov     rsi, rbx
0x111e2e:    mov     rdi, r12
0x111e31:    call    0x118f40
0x11208c:    mov     rdi, qword ptr [r14 + 0x18]
0x112090:    call    0x10c4b0
0x111e36:    cmp     eax, -1
0x111e39:    je     0x1121c8
0x112095:    mov     eax, ebx
0x112097:    and     eax, 0x7f
0x11209a:    add     eax, 1
0x11209d:    cmp     al, 1
0x11209f:    jg     0x111ea1
0x1121c8:    movzx   r13d, byte ptr [r15 - 1]
0x1121cd:    mov     byte ptr [r15 - 1], 0
0x1121d2:    mov     rsi, rbx
0x1121d5:    cmp     byte ptr [r12], 0
0x1121da:    je     0x112200
0x111e3f:    mov     rax, qword ptr [rsp + 0xa0]
0x111e47:    cmp     qword ptr [rsp + 0x10], rax
0x111e4c:    jne     0x111e5a
0x1120a5:    mov     r13d, 8
0x1120ab:    test    bl, 0x7f
0x1120ae:    jne     0x111ea1

```

```

0x112200:    lea     rdi, [rip + 0xf0a0]
0x112207:    call   0x118f40
0x1121dc:    mov     rdi, r12
0x1121df:    call   0x118f40
0x111e4e:    mov     eax, dword ptr [r14 + 0x10]
0x111e52:    test    eax, eax
0x111e54:    je      0x112188
0x111e5a:    mov     r9d, dword ptr [rsp + 0xc0]
0x111e62:    mov     rdx, r12
0x111e65:    mov     rsi, rbp
0x111e68:    xor     eax, eax
0x111e6a:    mov     r8d, dword ptr [rsp + 0xbc]
0x111e72:    mov     ecx, dword ptr [rsp + 0xb8]
0x111e79:    mov     rdi, r14
0x111e7c:    call   0x111330
0x1120b4:    and     bh, 0xff
0x1120b7:    jne     0x1122b0
0x11220c:    jmp     0x1121e4
0x1121e4:    mov     byte ptr [r15 - 1], r13b
0x1121e8:    cmp     eax, -1
0x1121eb:    jne     0x111e3f
0x112188:    xor     eax, eax
0x11218a:    call   0x118ab0
0x111e81:    mov     r13d, eax
0x111e84:    cmp     eax, 3
0x111e87:    je      0x111ea1
0x1120bd:    mov     r13d, 1
0x1120c3:    jmp     0x111ea1
0x1122b0:    lea     rdi, [rip + 0xfef3]
0x1122b7:    call   0x1195e0
0x1121f1:    mov     r13d, 7
0x1121f7:    jmp     0x111ea1
0x11218f:    mov     eax, dword ptr [rsp + 0x28]
0x112193:    and     eax, 0xf000
0x112198:    sub     eax, 0x4000
0x11219d:    test    eax, 0xfffffb000
0x1121a2:    je      0x112250
0x111e89:    jg      0x1120d0
0x1122bc:    jmp     0x111ea1
0x112250:    mov     rdi, r12
0x112253:    call   0x104870
0x1121a8:    mov     rdi, rbp
0x1121ab:    mov     r13d, 5
0x1121b1:    call   0x104870
0x111e8f:    cmp     eax, 1
0x111e92:    je      0x1121a8
0x1120d0:    cmp     eax, 8
0x1120d3:    jne     0x112210

```

```

0x112258:      mov     rsi, r12
0x11225b:      mov     rdi, rbp
0x11225e:      call    0x104e10
0x1121b6:      cmp     eax, -1
0x1121b9:      jne     0x1120bd
0x111e98:      cmp     eax, 2
0x111e9b:      jne     0x112210
0x1120d9:      mov     r13d, 0xa
0x1120df:      jmp     0x111ea1
0x112210:      sub     r13d, 1
0x112214:      lea     rax, [rip + 0x18205]
0x11221b:      xor     ecx, ecx
0x11221d:      mov     rsi, rbp
0x112220:      movsxd  r13, r13d
0x112223:      mov     r8d, 0xe
0x112229:      lea     rdi, [rip + 0xf079]
0x112230:      mov     rdx, qword ptr [rax + r13*8]
0x112234:      xor     eax, eax
0x112236:      mov     r13d, 0xb
0x11223c:      call    0x10f7e0
0x112263:      cmp     eax, -1
0x112266:      jne     0x1121a8
0x1121bf:      jmp     0x111ea1
0x112241:      jmp     0x111ea1
0x11226c:      jmp     0x112178
0x111ecd:      nop     dword ptr [rax]
0x111f1b:      nop     dword ptr [rax + rax]
0x1120c8:      nop     dword ptr [rax + rax]
0x1120e4:      nop     dword ptr [rax]
0x11216a:      nop     word ptr [rax + rax]
0x112183:      nop     dword ptr [rax + rax]
0x1121c4:      nop     dword ptr [rax]
0x1121fc:      nop     dword ptr [rax]
0x11220e:      nop
0x112246:      nop     word ptr cs:[rax + rax]
0x112271:      nop     dword ptr [rax]
0x1122aa:      nop     word ptr [rax + rax]
0x1122c6:      nop     word ptr cs:[rax + rax]
0x1122d0:      endbr64
0x1122d4:      push    r15
0x1122d6:      push    r14
0x1122d8:      push    r13
0x1122da:      mov     r13, rsi
0x1122dd:      push    r12
0x1122df:      mov     r12, rdi
0x1122e2:      mov     rdi, rsi
0x1122e5:      push    rbp
0x1122e6:      push    rbx

```

```

0x1122e7:      sub     rsp, 0x18
0x1122eb:      call   0x1049b0
0x1122f0:      lea     rdi, [rax + 1]
0x1122f4:      mov     rbx, rax
0x1122f7:      call   0x11d670
0x1122fc:      movsxd  r14, dword ptr [r12 + 0x30]
0x112301:      mov     rsi, r13
0x112304:      mov     rbp, rax
0x112307:      mov     rdi, rax
0x11230a:      cmp     r14, rbx
0x11230d:      cmova   r14, rbx
0x112311:      mov     rdx, r14
0x112314:      call   0x104880
0x112319:      mov     byte ptr [rbp + r14], 0
0x11231f:      mov     ebx, dword ptr [r12 + 0x68]
0x112324:      test    ebx, ebx
0x112326:      jle     0x112450
0x112450:      xor     r12d, r12d
0x112453:      jmp     0x11235c
0x11232c:      mov     r15, qword ptr [r12 + 0x38]
0x112331:      xor     r12d, r12d
0x112334:      mov     qword ptr [rsp + 8], r15
0x112339:      mov     r13, r15
0x11233c:      jmp     0x11234d
0x11235c:      mov     rdi, rbp
0x11235f:      call   0x118d10
0x11234d:      mov     rsi, qword ptr [r15]
0x112350:      mov     rdi, rbp
0x112353:      call   0x104b30
0x112364:      add     rsp, 0x18
0x112368:      mov     eax, r12d
0x11236b:      pop     rbx
0x11236c:      pop     rbp
0x11236d:      pop     r12
0x11236f:      pop     r13
0x112371:      pop     r14
0x112373:      pop     r15
0x112375:      ret
0x112358:      test    eax, eax
0x11235a:      jne     0x112340
0x112340:      add     r12d, 1
0x112344:      add     r15, 0x50
0x112348:      cmp     ebx, r12d
0x11234b:      je      0x112380
0x112380:      xor     r12d, r12d
0x112383:      jmp     0x112395
0x112395:      mov     rsi, qword ptr [r13]
0x112399:      mov     rdi, rbp

```

```

0x11239c:      call    0x104830
0x1123a1:      test   eax, eax
0x1123a3:      jne     0x112388
0x1123a5:      cmp     ebx, r12d
0x1123a8:      jne     0x11235c
0x112388:      add     r12d, 1
0x11238c:      add     r13, 0x50
0x112390:      cmp     ebx, r12d
0x112393:      je      0x1123b0
0x1123aa:      nop     word ptr [rax + rax]
0x1123b0:      mov     r15, qword ptr [rsp + 8]
0x1123b5:      xor     r12d, r12d
0x1123b8:      jmp     0x1123cd
0x1123cd:      mov     r13, qword ptr [r15]
0x1123d0:      mov     rdi, r13
0x1123d3:      call    0x1049b0
0x1123d8:      mov     rsi, r13
0x1123db:      mov     rdi, rbp
0x1123de:      cmp     rax, r14
0x1123e1:      cmova   rax, r14
0x1123e5:      mov     rdx, rax
0x1123e8:      call    0x104890
0x1123ed:      test   eax, eax
0x1123ef:      jne     0x1123c0
0x1123f1:      cmp     r12d, ebx
0x1123f4:      jne     0x11235c
0x1123c0:      add     r12d, 1
0x1123c4:      add     r15, 0x50
0x1123c8:      cmp     r12d, ebx
0x1123cb:      je      0x112400
0x1123fa:      nop     word ptr [rax + rax]
0x112400:      xor     r12d, r12d
0x112403:      jmp     0x11241a
0x11241a:      mov     r15, qword ptr [rsp + 8]
0x11241f:      mov     r13, qword ptr [r15]
0x112422:      mov     rdi, r13
0x112425:      call    0x1049b0
0x11242a:      mov     rsi, r13
0x11242d:      mov     rdi, rbp
0x112430:      cmp     rax, r14
0x112433:      cmova   rax, r14
0x112437:      mov     rdx, rax
0x11243a:      call    0x104c30
0x11243f:      test   eax, eax
0x112441:      jne     0x112408
0x112443:      cmp     r12d, ebx
0x112446:      jne     0x11235c
0x112408:      add     r15, 0x50

```

```

0x11240c:    add     r12d, 1
0x112410:    mov     qword ptr [rsp + 8], r15
0x112415:    cmp     r12d, ebx
0x112418:    je      0x112450
0x11244c:    nop     dword ptr [rax]
0x11233e:    nop
0x112376:    nop     word ptr cs:[rax + rax]
0x112385:    nop     dword ptr [rax]
0x1123ba:    nop     word ptr [rax + rax]
0x112405:    nop     dword ptr [rax]
0x112458:    nop     dword ptr [rax + rax]
0x112460:    endbr64
0x112464:    sub     rsp, 0xa8
0x11246b:    mov     rax, qword ptr fs:[0x28]
0x112474:    mov     qword ptr [rsp + 0x98], rax
0x11247c:    xor     eax, eax
0x11247e:    mov     rsi, rsp
0x112481:    call    0x104b90
0x112486:    mov     r8d, eax
0x112489:    xor     eax, eax
0x11248b:    test    r8d, r8d
0x11248e:    js      0x1124a4
0x1124a4:    mov     rdx, qword ptr [rsp + 0x98]
0x1124ac:    sub     rdx, qword ptr fs:[0x28]
0x1124b5:    jne     0x1124bf
0x112490:    mov     eax, dword ptr [rsp + 0x18]
0x112494:    and     eax, 0xf000
0x112499:    cmp     eax, 0x4000
0x11249e:    sete    al
0x1124a1:    movzx   eax, al
0x1124a4:    mov     rdx, qword ptr [rsp + 0x98]
0x1124ac:    sub     rdx, qword ptr fs:[0x28]
0x1124b5:    jne     0x1124bf
0x1124b7:    add     rsp, 0xa8
0x1124be:    ret
0x1124bf:    call    0x1049d0
0x1124c4:    nop     word ptr cs:[rax + rax]
0x1124cf:    nop
0x1124d0:    endbr64
0x1124d4:    push    r13
0x1124d6:    push    r12
0x1124d8:    mov     r12d, esi
0x1124db:    push    rbp
0x1124dc:    push    rbx
0x1124dd:    sub     rsp, 0xa8
0x1124e4:    mov     rax, qword ptr fs:[0x28]
0x1124ed:    mov     qword ptr [rsp + 0x98], rax
0x1124f5:    xor     eax, eax

```

```

0x1124f7:    mov     r13, rsp
0x1124fa:    call    0x11d7a0
0x1124ff:    cmp     byte ptr [rax], 0
0x112502:    mov     rbp, rax
0x112505:    mov     rbx, rax
0x112508:    je      0x112560
0x112560:    mov     rdi, rbp
0x112563:    call    0x118d10
0x11250a:    nop     word ptr [rax + rax]
0x112568:    xor     eax, eax
0x11256a:    mov     rdx, qword ptr [rsp + 0x98]
0x112572:    sub     rdx, qword ptr fs:[0x28]
0x11257b:    jne     0x1125ae
0x112510:    movzx   eax, byte ptr [rbx + 1]
0x112514:    add     rbx, 1
0x112518:    test    al, al
0x11251a:    je      0x112560
0x11257d:    add     rsp, 0xa8
0x112584:    pop     rbx
0x112585:    pop     rbp
0x112586:    pop     r12
0x112588:    pop     r13
0x11258a:    ret
0x1125ae:    call    0x1049d0
0x11251c:    cmp     rbp, rbx
0x11251f:    je      0x112510
0x112521:    cmp     al, 0x2f
0x112523:    jne     0x112510
0x112525:    mov     byte ptr [rbx], 0
0x112528:    mov     rsi, r13
0x11252b:    mov     rdi, rbp
0x11252e:    call    0x104b90
0x112533:    test    eax, eax
0x112535:    js      0x112590
0x112590:    mov     esi, r12d
0x112593:    mov     rdi, rbp
0x112596:    call    0x1048c0
0x112537:    mov     eax, dword ptr [rsp + 0x18]
0x11253b:    and     eax, 0xf000
0x112540:    cmp     eax, 0x4000
0x112545:    jne     0x112590
0x11259b:    test    eax, eax
0x11259d:    je      0x112547
0x112547:    mov     byte ptr [rbx], 0x2f
0x11254a:    movzx   eax, byte ptr [rbx + 1]
0x11254e:    add     rbx, 1
0x112552:    test    al, al
0x112554:    jne     0x11251c

```

```

0x11259f:    mov     rdi, rbp
0x1125a2:    call   0x118d10
0x112556:    nop     word ptr cs:[rax + rax]
0x1125a7:    mov     eax, 0xffffffff
0x1125ac:    jmp     0x11256a
0x11256a:    mov     rdx, qword ptr [rsp + 0x98]
0x112572:    sub     rdx, qword ptr fs:[0x28]
0x11257b:    jne     0x1125ae
0x11258b:    nop     dword ptr [rax + rax]
0x1125b3:    nop     word ptr cs:[rax + rax]
0x1125be:    nop
0x1125c0:    endbr64
0x1125c4:    push    r15
0x1125c6:    movsxd  rsi, esi
0x1125c9:    mov     r10, rdx
0x1125cc:    movsxd  r8, r8d
0x1125cf:    push    r14
0x1125d1:    lea     rax, [rsi + rsi*4]
0x1125d5:    mov     rsi, rdi
0x1125d8:    push    r13
0x1125da:    shl     rax, 4
0x1125de:    push    r12
0x1125e0:    push    rbp
0x1125e1:    push    rbx
0x1125e2:    sub     rsp, 0x68
0x1125e6:    mov     qword ptr [rsp + 0x48], rdi
0x1125eb:    mov     rdi, rax
0x1125ee:    mov     qword ptr [rsp + 0x50], rax
0x1125f3:    add     rax, qword ptr [rsi + 0x38]
0x1125f7:    mov     qword ptr [rsp + 0x20], rdx
0x1125fc:    mov     rdx, rax
0x1125ff:    mov     rax, qword ptr [rax]
0x112602:    mov     qword ptr [rsp + 0x38], rcx
0x112607:    mov     rcx, qword ptr [rcx + 0x38]
0x11260b:    mov     qword ptr [rsp + 0x10], rax
0x112610:    lea     rax, [r8 + r8*4]
0x112614:    shl     rax, 4
0x112618:    mov     qword ptr [rsp + 0x28], r9
0x11261d:    lea     rsi, [rcx + rax]
0x112621:    mov     qword ptr [rsp + 0x58], rax
0x112626:    mov     eax, dword ptr [rdx + 0x10]
0x112629:    mov     ecx, dword ptr [rcx + rdi + 0x10]
0x11262d:    mov     rbx, qword ptr [rsi]
0x112630:    mov     qword ptr [r9], 0
0x112637:    mov     r8d, eax
0x11263a:    mov     qword ptr [r10], 0
0x112641:    mov     edi, ecx
0x112643:    and     r8d, 0xf000

```



```

0x11264a:    and     edi, 0xf000
0x112650:    cmp     r8d, 0x4000
0x112657:    je      0x112669
0x112669:    mov     rax, qword ptr [rdx + 8]
0x11266d:    mov     qword ptr [rsp + 8], rax
0x112672:    cmp     edi, 0x4000
0x112678:    je      0x112ac0
0x112659:    and     eax, 0xd000
0x11265e:    cmp     eax, 0x8000
0x112663:    jne     0x112928
0x112ac0:    mov     rcx, qword ptr [rsp + 8]
0x112ac5:    mov     rax, qword ptr [rsi + 8]
0x112ac9:    mov     dword ptr [rsp + 0x40], 0
0x112ad1:    mov     dword ptr [rsp + 0x44], 0
0x112ad9:    cmp     rcx, rax
0x112adc:    cmovge  rax, rcx
0x112ae0:    mov     qword ptr [rsp + 8], rax
0x112ae5:    jmp     0x1126a0
0x11267e:    and     ecx, 0xd000
0x112684:    mov     dword ptr [rsp + 0x40], 0
0x11268c:    mov     dword ptr [rsp + 0x44], 1
0x112694:    cmp     ecx, 0x8000
0x11269a:    je      0x112ac0
0x112928:    cmp     edi, 0x4000
0x11292e:    je      0x1129f0
0x1126a0:    cmp     qword ptr [rsp + 8], 0
0x1126a6:    je      0x112c90
0x1129f0:    mov     rax, qword ptr [rsi + 8]
0x1129f4:    mov     dword ptr [rsp + 0x40], 1
0x1129fc:    mov     dword ptr [rsp + 0x44], 0
0x112a04:    mov     qword ptr [rsp + 8], rax
0x112a09:    jmp     0x1126a0
0x112934:    and     ecx, 0xd000
0x11293a:    cmp     ecx, 0x8000
0x112940:    je      0x1129f0
0x112c90:    xor     r12d, r12d
0x112c93:    jmp     0x112b09
0x1126ac:    mov     r14, qword ptr [rsp + 0x38]
0x1126b1:    mov     rdi, qword ptr [r14 + 0x18]
0x1126b5:    call    0x1049b0
0x112946:    mov     r14, qword ptr [rsp + 0x38]
0x11294b:    mov     rdi, qword ptr [r14 + 0x18]
0x11294f:    call    0x1049b0
0x112b09:    add     rsp, 0x68
0x112b0d:    mov     rax, r12
0x112b10:    pop     rbx
0x112b11:    pop     rbp
0x112b12:    pop     r12

```

```

0x112b14:    pop     r13
0x112b16:    pop     r14
0x112b18:    pop     r15
0x112b1a:    ret
0x1126ba:    mov     rdi, rbx
0x1126bd:    mov     rbp, rax
0x1126c0:    call    0x1049b0
0x112954:    mov     rdi, rbx
0x112957:    mov     rbp, rax
0x11295a:    call    0x1049b0
0x1126c5:    lea     rdi, [rbp + rax + 2]
0x1126ca:    call    0x11d670
0x11295f:    lea     rdi, [rbp + rax + 2]
0x112964:    call    0x11d670
0x1126cf:    mov     r8, qword ptr [r14 + 0x18]
0x1126d3:    mov     r9, rbx
0x1126d6:    lea     rcx, [rip + 0xe024]
0x1126dd:    mov     rdx, -1
0x1126e4:    mov     esi, 1
0x1126e9:    mov     rdi, rax
0x1126ec:    mov     r12, rax
0x1126ef:    xor     eax, eax
0x1126f1:    call    0x104f10
0x112969:    mov     r8, qword ptr [r14 + 0x18]
0x11296d:    mov     r9, rbx
0x112970:    lea     rcx, [rip + 0xdd8a]
0x112977:    mov     rdx, -1
0x11297e:    mov     esi, 1
0x112983:    mov     rdi, rax
0x112986:    mov     r12, rax
0x112989:    xor     eax, eax
0x11298b:    call    0x104f10
0x1126f6:    mov     rbx, qword ptr [rsp + 0x10]
0x1126fb:    mov     rdi, rbx
0x1126fe:    call    0x1049b0
0x112990:    mov     rbx, qword ptr [rsp + 0x10]
0x112995:    mov     rdi, rbx
0x112998:    call    0x1049b0
0x112703:    lea     rdi, [rax + 0x21]
0x112707:    call    0x11d670
0x11299d:    lea     rdi, [rax + 0x21]
0x1129a1:    call    0x11d670
0x11270c:    mov     r8, rbx
0x11270f:    mov     esi, 1
0x112714:    lea     rcx, [rip + 0xebbd]
0x11271b:    mov     rbp, rax
0x11271e:    mov     rdi, rax
0x112721:    mov     rdx, -1

```

```

0x112728:    xor     eax, eax
0x11272a:    call    0x104f10
0x1129a6:    mov     r8, rbx
0x1129a9:    mov     esi, 1
0x1129ae:    lea     rcx, [rip + 0xe909]
0x1129b5:    mov     rdi, rax
0x1129b8:    mov     rbp, rax
0x1129bb:    mov     rdx, -1
0x1129c2:    xor     eax, eax
0x1129c4:    call    0x104f10
0x11272f:    mov     edx, 1
0x112734:    mov     esi, 1
0x112739:    mov     rdi, rbp
0x11273c:    call    0x10e480
0x1129c9:    mov     dword ptr [rsp + 0x44], 1
0x1129d1:    mov     qword ptr [rsp + 8], 0
0x1129da:    mov     dword ptr [rsp + 0x40], 1
0x1129e2:    jmp     0x11272f
0x112741:    xor     eax, eax
0x112743:    call    0x10a820
0x112748:    mov     rdi, rbp
0x11274b:    call    0x118d10
0x112750:    mov     rdi, qword ptr [rsp + 0x10]
0x112755:    xor     esi, esi
0x112757:    xor     eax, eax
0x112759:    call    0x104d30
0x11275e:    mov     r15d, eax
0x112761:    cmp     eax, -1
0x112764:    je      0x112c6b
0x112c6b:    mov     r12, -4
0x112c72:    jmp     0x112b09
0x11276a:    mov     rdi, r12
0x11276d:    xor     esi, esi
0x11276f:    xor     eax, eax
0x112771:    call    0x104d30
0x112776:    mov     rdi, r12
0x112779:    mov     dword ptr [rsp + 0x1c], eax
0x11277d:    cmp     eax, -1
0x112780:    je      0x112c77
0x112c77:    call    0x118d10
0x112786:    call    0x118d10
0x112c7c:    mov     edi, r15d
0x112c7f:    mov     r12, -5
0x112c86:    call    0x104ab0
0x11278b:    mov     edi, 0x8000
0x112790:    xor     r12d, r12d
0x112793:    call    0x11d670
0x112c8b:    jmp     0x112b09

```

```

0x112798:    mov     edi, 0x8000
0x11279d:    mov     rbp, rax
0x1127a0:    call    0x11d670
0x1127a5:    mov     rdi, qword ptr [rsp + 0x10]
0x1127aa:    mov     r13, rax
0x1127ad:    call    0x1049b0
0x1127b2:    lea     rdi, [rax + 0x21]
0x1127b6:    call    0x11d670
0x1127bb:    mov     dword ptr [rsp + 0x18], r15d
0x1127c0:    movabs  rdx, 0xa3d70a3d70a3d70b
0x1127ca:    mov     r14, rax
0x1127cd:    mov     rax, qword ptr [rsp + 8]
0x1127d2:    imul    rdx
0x1127d5:    mov     rax, qword ptr [rsp + 8]
0x1127da:    add     rdx, rax
0x1127dd:    sar     rax, 0x3f
0x1127e1:    sar     rdx, 6
0x1127e5:    sub     rdx, rax
0x1127e8:    mov     qword ptr [rsp + 0x30], rdx
0x1127ed:    jmp     0x11282b
0x11282b:    mov     ecx, dword ptr [rip + 0x1927f]
0x112831:    test    ecx, ecx
0x112833:    jne     0x112a30
0x112839:    mov     edi, 1
0x11283e:    call    0x118b70
0x112a30:    mov     dword ptr [rip + 0x19076], 0
0x112a3a:    xor     eax, eax
0x112a3c:    call    0x10c8c0
0x112843:    mov     edi, dword ptr [rsp + 0x18]
0x112847:    mov     edx, 0x8000
0x11284c:    mov     rsi, rbp
0x11284f:    call    0x1047a0
0x112a41:    mov     edx, 4
0x112a46:    lea     rsi, [rip + 0xdae5]
0x112a4d:    lea     rdi, [rip + 0xe7a5]
0x112a54:    mov     r15, rax
0x112a57:    call    0x108070
0x112854:    xor     edi, edi
0x112856:    mov     rbx, rax
0x112859:    call    0x118b70
0x112a5c:    mov     rdi, r15
0x112a5f:    mov     ebx, eax
0x112a61:    call    0x10c970
0x11285e:    test    ebx, ebx
0x112860:    js      0x112af0
0x112a66:    xor     eax, eax
0x112a68:    and     ebx, 0xffffffff
0x112a6b:    call    0x10d870

```

```

0x112af0:    mov     edi, dword ptr [rsp + 0x18]
0x112af4:    mov     r12, -2
0x112afb:    call    0x104ab0
0x112866:    mov     rcx, qword ptr [rsp + 0x20]
0x11286b:    movsxd  rax, ebx
0x11286e:    mov     edi, 1
0x112873:    add     qword ptr [rcx], rax
0x112876:    call    0x118b70
0x112a70:    xor     eax, eax
0x112a72:    call    0x10d740
0x112b00:    mov     edi, dword ptr [rsp + 0x1c]
0x112b04:    call    0x104ab0
0x11287b:    mov     edi, dword ptr [rsp + 0x1c]
0x11287f:    mov     edx, 0x8000
0x112884:    mov     rsi, r13
0x112887:    call    0x1047a0
0x112a77:    xor     eax, eax
0x112a79:    call    0x10a820
0x11288c:    xor     edi, edi
0x11288e:    mov     r15, rax
0x112891:    call    0x118b70
0x112a7e:    cmp     bl, 0x4e
0x112a81:    je      0x112839
0x112896:    test    r15d, r15d
0x112899:    js      0x112b20
0x112a87:    mov     rdi, r14
0x112a8a:    mov     r15d, dword ptr [rsp + 0x18]
0x112a8f:    mov     r12, -1
0x112a96:    call    0x118d10
0x112b20:    mov     edi, dword ptr [rsp + 0x18]
0x112b24:    mov     r12, -3
0x112b2b:    call    0x104ab0
0x11289f:    mov     rax, qword ptr [rsp + 0x28]
0x1128a4:    movsxd  rdx, r15d
0x1128a7:    add     qword ptr [rax], rdx
0x1128aa:    cmp     r15d, ebx
0x1128ad:    cmovle  ebx, r15d
0x1128b1:    test    ebx, ebx
0x1128b3:    je      0x112b40
0x112a9b:    mov     rdi, rbp
0x112a9e:    call    0x118d10
0x112b30:    mov     edi, dword ptr [rsp + 0x1c]
0x112b34:    call    0x104ab0
0x112b40:    mov     r15d, dword ptr [rsp + 0x18]
0x112b45:    mov     rdi, r14
0x112b48:    call    0x118d10
0x1128b9:    movsxd  r15, ebx
0x1128bc:    mov     rsi, r13

```

```

0x1128bf:    mov     rdi, rbp
0x1128c2:    mov     rdx, r15
0x1128c5:    call    0x104af0
0x112aa3:    mov     rdi, r13
0x112aa6:    call    0x118d10
0x112b39:    jmp     0x112b09
0x112b4d:    mov     rdi, rbp
0x112b50:    call    0x118d10
0x1128ca:    test    eax, eax
0x1128cc:    jne     0x112b70
0x112aab:    mov     edi, r15d
0x112aae:    call    0x104ab0
0x112b55:    mov     rdi, r13
0x112b58:    call    0x118d10
0x1128d2:    add     r12, r15
0x1128d5:    cmp     qword ptr [rsp + 8], 0
0x1128db:    je      0x1127f0
0x112b70:    mov     r15d, dword ptr [rsp + 0x18]
0x112b75:    xor     eax, eax
0x112b77:    nop     word ptr [rax + rax]
0x112b80:    movzx   ecx, byte ptr [r13 + rax]
0x112b86:    mov     edx, eax
0x112b88:    cmp     byte ptr [rbp + rax], cl
0x112b8c:    jne     0x112b99
0x112ab3:    mov     edi, dword ptr [rsp + 0x1c]
0x112ab7:    call    0x104ab0
0x112b5d:    mov     edi, r15d
0x112b60:    call    0x104ab0
0x1127f0:    mov     r8, qword ptr [rsp + 0x10]
0x1127f5:    mov     esi, 1
0x1127fa:    mov     rdi, r14
0x1127fd:    xor     eax, eax
0x1127ff:    lea     rcx, [rip + 0xeab8]
0x112806:    mov     rdx, -1
0x11280d:    call    0x104f10
0x1128e1:    cmp     qword ptr [rsp + 8], 0x1869f
0x1128ea:    jle     0x112a10
0x112b8e:    lea     edx, [rax + 1]
0x112b91:    add     rax, 1
0x112b95:    cmp     ebx, eax
0x112b97:    jg      0x112b80
0x112b99:    movsxd  rdx, edx
0x112b9c:    add     r12, rdx
0x112b9f:    mov     edx, dword ptr [rsp + 0x40]
0x112ba3:    test    edx, edx
0x112ba5:    jne     0x112bea
0x112abc:    jmp     0x112b09
0x112b65:    mov     edi, dword ptr [rsp + 0x1c]

```

```

0x112b69:    call    0x104ab0
0x112812:    mov     edx, 1
0x112817:    mov     esi, 1
0x11281c:    mov     rdi, r14
0x11281f:    call    0x10e480
0x112a10:    lea     rax, [r12 + r12*4]
0x112a14:    lea     rax, [rax + rax*4]
0x112a18:    shl     rax, 2
0x112a1c:    cqo
0x112a1e:    idiv    qword ptr [rsp + 8]
0x112a23:    mov     r8, rax
0x112a26:    jmp     0x1128fd
0x1128f0:    mov     rax, r12
0x1128f3:    cqo
0x1128f5:    idiv    qword ptr [rsp + 0x30]
0x1128fa:    mov     r8, rax
0x1128fd:    mov     r9, qword ptr [rsp + 0x10]
0x112902:    mov     esi, 1
0x112907:    mov     rdi, r14
0x11290a:    xor     eax, eax
0x11290c:    lea     rcx, [rip + 0xe9b4]
0x112913:    mov     rdx, -1
0x11291a:    call    0x104f10
0x112b80:    movzx   ecx, byte ptr [r13 + rax]
0x112b86:    mov     edx, eax
0x112b88:    cmp     byte ptr [rbp + rax], cl
0x112b8c:    jne     0x112b99
0x112ba7:    mov     rax, qword ptr [rsp + 0x48]
0x112bac:    mov     rcx, qword ptr [rsp + 0x50]
0x112bb1:    mov     rax, qword ptr [rax + 0x38]
0x112bb5:    mov     rax, qword ptr [rax + rcx + 8]
0x112bba:    mov     rcx, qword ptr [rsp + 0x20]
0x112bbf:    mov     qword ptr [rcx], rax
0x112bc2:    mov     eax, dword ptr [rsp + 0x44]
0x112bc6:    test    eax, eax
0x112bc8:    jne     0x112c24
0x112bea:    mov     edi, 1
0x112bef:    call    0x118b70
0x112b6e:    jmp     0x112b09
0x112824:    xor     eax, eax
0x112826:    call    0x10a820
0x1128fd:    mov     r9, qword ptr [rsp + 0x10]
0x112902:    mov     esi, 1
0x112907:    mov     rdi, r14
0x11290a:    xor     eax, eax
0x11290c:    lea     rcx, [rip + 0xe9b4]
0x112913:    mov     rdx, -1
0x11291a:    call    0x104f10

```

```

0x11291f:      jmp      0x112812
0x112bca:      mov     rax, qword ptr [rsp + 0x38]
0x112bcf:      mov     rcx, qword ptr [rsp + 0x58]
0x112bd4:      mov     rax, qword ptr [rax + 0x38]
0x112bd8:      mov     rax, qword ptr [rax + rcx + 8]
0x112bdd:      mov     rcx, qword ptr [rsp + 0x28]
0x112be2:      mov     qword ptr [rcx], rax
0x112be5:      jmp     0x112b45
0x112c24:      mov     edi, 1
0x112c29:      call    0x118b70
0x112bf4:      mov     rbx, qword ptr [rsp + 0x20]
0x112bf9:      jmp     0x112c07
0x112b45:      mov     rdi, r14
0x112b48:      call    0x118d10
0x112c2e:      mov     qword ptr [rsp + 8], rbp
0x112c33:      mov     rbx, qword ptr [rsp + 0x28]
0x112c38:      mov     ebp, dword ptr [rsp + 0x1c]
0x112c3c:      jmp     0x112c47
0x112c07:      mov     edx, 0x8000
0x112c0c:      mov     rsi, rbp
0x112c0f:      mov     edi, r15d
0x112c12:      call    0x1047a0
0x112c47:      mov     edx, 0x8000
0x112c4c:      mov     rsi, r13
0x112c4f:      mov     edi, ebp
0x112c51:      call    0x1047a0
0x112c17:      test    eax, eax
0x112c19:      jg      0x112c00
0x112c56:      test    eax, eax
0x112c58:      jg      0x112c40
0x112c1b:      xor     edi, edi
0x112c1d:      call    0x118b70
0x112c00:      cdqe
0x112c02:      add     qword ptr [rbx], rax
0x112c05:      je      0x112c1b
0x112c5a:      xor     edi, edi
0x112c5c:      mov     rbp, qword ptr [rsp + 8]
0x112c61:      call    0x118b70
0x112c40:      cdqe
0x112c42:      add     qword ptr [rbx], rax
0x112c45:      je      0x112c5a
0x112c22:      jmp     0x112bc2
0x112c66:      jmp     0x112b45
0x112bc2:      mov     eax, dword ptr [rsp + 0x44]
0x112bc6:      test    eax, eax
0x112bc8:      jne     0x112c24
0x1127ef:      nop
0x112924:      nop     dword ptr [rax]

```



```

0x1129e7:      nop      word ptr [rax + rax]
0x112a0e:      nop
0x112a2b:      nop      dword ptr [rax + rax]
0x112abe:      nop
0x112aea:      nop      word ptr [rax + rax]
0x112b1b:      nop      dword ptr [rax + rax]
0x112b3b:      nop      dword ptr [rax + rax]
0x112bfb:      nop      dword ptr [rax + rax]
0x112c3e:      nop
0x112c98:      nop      dword ptr [rax + rax]
0x112ca0:      endbr64
0x112ca4:      push     r15
0x112ca6:      push     r14
0x112ca8:      push     r13
0x112caa:      push     r12
0x112cac:      push     rbp
0x112cad:      mov      rbp, rsi
0x112cb0:      push     rbx
0x112cb1:      mov      rbx, rdi
0x112cb4:      sub      rsp, 0x38
0x112cb8:      mov      r13d, dword ptr [rdi + 0x28]
0x112cbc:      mov      rdi, qword ptr [rip + 0x1780d]
0x112cc3:      mov      rax, qword ptr fs:[0x28]
0x112ccc:      mov      qword ptr [rsp + 0x28], rax
0x112cd1:      xor      eax, eax
0x112cd3:      mov      r14d, dword ptr [rsi + 0x28]
0x112cd7:      call     0x10db60
0x112cdc:      xor      eax, eax
0x112cde:      call     0x10a820
0x112ce3:      mov      r15, qword ptr [rbx + 0x18]
0x112ce7:      mov      rsi, qword ptr [rbp + 0x18]
0x112ceb:      mov      rdi, r15
0x112cee:      call     0x104b30
0x112cf3:      mov      rdx, qword ptr [rbx + 0x38]
0x112cf7:      mov      r8d, eax
0x112cfa:      movsxd   rax, r13d
0x112cfd:      lea      r12, [rax + rax*4]
0x112d01:      shl      r12, 4
0x112d05:      add      rdx, r12
0x112d08:      test     r8d, r8d
0x112d0b:      jne      0x112d39
0x112d0d:      movsxd   rax, r14d
0x112d10:      mov      rdi, qword ptr [rdx]
0x112d13:      mov      qword ptr [rsp + 8], rdx
0x112d18:      lea      rax, [rax + rax*4]
0x112d1c:      shl      rax, 4
0x112d20:      add      rax, qword ptr [rbp + 0x38]
0x112d24:      mov      rsi, qword ptr [rax]

```

```

0x112d27:    call    0x104b30
0x112d39:    cmp     byte ptr [rdx + 0x48], 1
0x112d3d:    jne     0x112d57
0x112d2c:    mov     rdx, qword ptr [rsp + 8]
0x112d31:    test    eax, eax
0x112d33:    je      0x112e40
0x112d3f:    movsxd  rax, r14d
0x112d42:    lea     r10, [rax + rax*4]
0x112d46:    mov     rax, qword ptr [rbp + 0x38]
0x112d4a:    shl     r10, 4
0x112d4e:    add     rax, r10
0x112d51:    cmp     byte ptr [rax + 0x48], 1
0x112d55:    je      0x112da0
0x112d57:    mov     ecx, 0x1e
0x112d5c:    xor     edx, edx
0x112d5e:    mov     rsi, r15
0x112d61:    xor     eax, eax
0x112d63:    lea     rdi, [rip + 0xede6]
0x112d6a:    call    0x10f760
0x112e40:    mov     rax, qword ptr [rsp + 0x28]
0x112e45:    sub     rax, qword ptr fs:[0x28]
0x112e4e:    jne     0x112fbc
0x112da0:    mov     rcx, qword ptr [rax + 8]
0x112da4:    cmp     qword ptr [rdx + 8], rcx
0x112da8:    je      0x112df2
0x112d6f:    xor     eax, eax
0x112d71:    call    0x10e4e0
0x112e54:    add     rsp, 0x38
0x112e58:    mov     edx, 0x1c
0x112e5d:    xor     esi, esi
0x112e5f:    pop     rbx
0x112e60:    lea     rdi, [rip + 0xec11]
0x112e67:    pop     rbp
0x112e68:    pop     r12
0x112e6a:    pop     r13
0x112e6c:    pop     r14
0x112e6e:    pop     r15
0x112e70:    jmp     0x108070
0x112fbc:    call    0x1049d0
0x112df2:    mov     esi, r13d
0x112df5:    lea     rdx, [rsp + 0x18]
0x112dfa:    lea     r9, [rsp + 0x20]
0x112dff:    mov     r8d, r14d
0x112e02:    mov     rcx, rbp
0x112e05:    mov     rdi, rbx
0x112e08:    xor     eax, eax
0x112e0a:    mov     qword ptr [rsp + 8], r10
0x112e0f:    call    0x1125c0

```

```

0x112daa:    mov     edx, dword ptr [rdx + 0x10]
0x112dad:    mov     ecx, edx
0x112daf:    and     ecx, 0xf000
0x112db5:    cmp     ecx, 0x4000
0x112dbb:    je      0x112dcb
0x112d76:    mov     rax, qword ptr [rsp + 0x28]
0x112d7b:    sub     rax, qword ptr fs:[0x28]
0x112d84:    jne     0x112fbc
0x112e14:    mov     r13, rax
0x112e17:    lea     eax, [rax + 5]
0x112e1a:    cmp     r13d, -5
0x112e1e:    jb      0x112f17
0x112dcb:    mov     eax, dword ptr [rax + 0x10]
0x112dce:    mov     edx, eax
0x112dd0:    and     edx, 0xf000
0x112dd6:    cmp     edx, 0x4000
0x112ddc:    je      0x112ee8
0x112dbd:    and     edx, 0xd000
0x112dc3:    cmp     edx, 0x8000
0x112dc9:    jne     0x112df2
0x112d8a:    add     rsp, 0x38
0x112d8e:    xor     eax, eax
0x112d90:    pop     rbx
0x112d91:    pop     rbp
0x112d92:    pop     r12
0x112d94:    pop     r13
0x112d96:    pop     r14
0x112d98:    pop     r15
0x112d9a:    jmp     0x10a820
0x112f17:    mov     rax, qword ptr [rsp + 0x18]
0x112f1c:    mov     rdx, qword ptr [rsp + 0x20]
0x112f21:    cmp     rax, r13
0x112f24:    je      0x112f87
0x112e24:    lea     rdx, [rip + 0xeac5]
0x112e2b:    mov     r10, qword ptr [rsp + 8]
0x112e30:    movsxd  rax, dword ptr [rdx + rax*4]
0x112e34:    add     rax, rdx
0x112e37:    jmp     rax
0x112ee8:    mov     edx, 0xc
0x112eed:    lea     rsi, [rip + 0xd63e]
0x112ef4:    mov     qword ptr [rsp + 8], r10
0x112ef9:    lea     rdi, [rip + 0xebb0]
0x112f00:    call    0x108070
0x112de2:    and     eax, 0xd000
0x112de7:    cmp     eax, 0x8000
0x112dec:    je      0x112ee8
0x112f87:    cmp     rax, rdx
0x112f8a:    jne     0x112f26

```

```

0x112f26:    cmp     rax, rdx
0x112f29:    cmovg   rax, rdx
0x112f2d:    cmp     rax, r13
0x112f30:    je      0x112fa4
0x112ea0:    mov     rax, qword ptr [rbp + 0x38]
0x112ea4:    mov     rsi, qword ptr [rax + r10]
0x112ea8:    jmp     0x112e80
0x112e78:    mov     rax, qword ptr [rbx + 0x38]
0x112e7c:    mov     rsi, qword ptr [rax + r12]
0x112e80:    mov     ecx, 0x1e
0x112e85:    xor     edx, edx
0x112e87:    lea     rdi, [rip + 0xec4a]
0x112e8e:    xor     eax, eax
0x112e90:    call    0x10f760
0x112eb0:    mov     rax, qword ptr [rbx + 0x38]
0x112eb4:    mov     rsi, qword ptr [rax + r12]
0x112eb8:    mov     ecx, 0x1e
0x112ebd:    xor     edx, edx
0x112ebf:    lea     rdi, [rip + 0xe423]
0x112ec6:    xor     eax, eax
0x112ec8:    call    0x10f760
0x112ed8:    mov     rax, qword ptr [rbp + 0x38]
0x112edc:    mov     rsi, qword ptr [rax + r10]
0x112ee0:    jmp     0x112eb8
0x112f05:    mov     r10, qword ptr [rsp + 8]
0x112f0a:    cmp     al, 0x79
0x112f0c:    jne     0x112d6f
0x112f8c:    mov     edx, 0x14
0x112f91:    xor     esi, esi
0x112f93:    lea     rdi, [rip + 0xe365]
0x112f9a:    call    0x108070
0x112fa4:    mov     edx, 0x14
0x112fa9:    xor     esi, esi
0x112fab:    lea     rdi, [rip + 0xeb4e]
0x112fb2:    call    0x108070
0x112f32:    mov     edi, 0x80
0x112f37:    call    0x11d670
0x112e80:    mov     ecx, 0x1e
0x112e85:    xor     edx, edx
0x112e87:    lea     rdi, [rip + 0xec4a]
0x112e8e:    xor     eax, eax
0x112e90:    call    0x10f760
0x112e95:    jmp     0x112d6f
0x112ecd:    jmp     0x112d6f
0x112eb8:    mov     ecx, 0x1e
0x112ebd:    xor     edx, edx
0x112ebf:    lea     rdi, [rip + 0xe423]
0x112ec6:    xor     eax, eax

```

```

0x112ec8:      call    0x10f760
0x112f12:      jmp     0x112df2
0x112f9f:      jmp     0x112d6f
0x112fb7:      jmp     0x112d6f
0x112f3c:      mov     r9, r13
0x112f3f:      mov     r8, r13
0x112f42:      mov     esi, 1
0x112f47:      mov     rbp, rax
0x112f4a:      mov     rdi, rax
0x112f4d:      xor     eax, eax
0x112f4f:      mov     rdx, -1
0x112f56:      lea     rcx, [rip + 0xe3af]
0x112f5d:      call    0x104f10
0x112f62:      mov     ecx, 0x14
0x112f67:      xor     edx, edx
0x112f69:      mov     rsi, rbp
0x112f6c:      lea     rdi, [rip + 0xe3a5]
0x112f73:      xor     eax, eax
0x112f75:      call    0x10f760
0x112f7a:      mov     rdi, rbp
0x112f7d:      call    0x118d10
0x112f82:      jmp     0x112d6f
0x112d9f:      nop
0x112e3a:      nop     word ptr [rax + rax]
0x112e75:      nop     dword ptr [rax]
0x112e9a:      nop     word ptr [rax + rax]
0x112eaa:      nop     word ptr [rax + rax]
0x112ed2:      nop     word ptr [rax + rax]
0x112ee2:      nop     word ptr [rax + rax]
0x112fc1:      nop     word ptr cs:[rax + rax]
0x112fcc:      nop     dword ptr [rax]
0x112fd0:      endbr64
0x112fd4:      push    r15
0x112fd6:      push    r14
0x112fd8:      push    r13
0x112fda:      push    r12
0x112fdc:      mov     r12, rdi
0x112fdf:      push    rbp
0x112fe0:      mov     rbp, rsi
0x112fe3:      push    rbx
0x112fe4:      sub     rsp, 0x48
0x112fe8:      mov     rdi, qword ptr [rip + 0x174d9]
0x112fef:      mov     dword ptr [rsp + 0x14], edx
0x112ff3:      mov     rax, qword ptr fs:[0x28]
0x112ffc:      mov     qword ptr [rsp + 0x38], rax
0x113001:      xor     eax, eax
0x113003:      call    0x10db60
0x113008:      xor     eax, eax

```

```

0x11300a:    call    0x10a820
0x11300f:    mov     rsi, qword ptr [rbp + 0x18]
0x113013:    mov     rdi, qword ptr [r12 + 0x18]
0x113018:    call    0x104b30
0x11301d:    test    eax, eax
0x11301f:    je      0x1132d0
0x1132d0:    mov     edx, 0x1c
0x1132d5:    xor     esi, esi
0x1132d7:    lea     rdi, [rip + 0xe89a]
0x1132de:    call    0x108070
0x113025:    mov     ecx, dword ptr [r12 + 0x68]
0x11302a:    test    ecx, ecx
0x11302c:    jle     0x11306d
0x1132e3:    mov     edx, dword ptr [r12 + 0x68]
0x1132e8:    test    edx, edx
0x1132ea:    jle     0x11331c
0x11306d:    mov     edx, dword ptr [rbp + 0x68]
0x113070:    test    edx, edx
0x113072:    jle     0x1130ab
0x11302e:    lea     edx, [rcx - 1]
0x113031:    mov     rsi, qword ptr [r12 + 0x38]
0x113036:    lea     rdx, [rdx + rdx*4]
0x11303a:    shl     rdx, 4
0x11303e:    lea     rax, [rsi + 0x49]
0x113042:    lea     rdx, [rsi + rdx + 0x99]
0x11304a:    nop     word ptr [rax + rax]
0x113050:    cmp     byte ptr [rax], 0
0x113053:    jne     0x113064
0x11331c:    mov     dword ptr [r12 + 0x70], 0
0x113325:    mov     edx, dword ptr [rbp + 0x68]
0x113328:    test    edx, edx
0x11332a:    jle     0x11335c
0x1132ec:    sub     edx, 1
0x1132ef:    mov     rcx, qword ptr [r12 + 0x38]
0x1132f4:    lea     rdx, [rdx + rdx*4]
0x1132f8:    shl     rdx, 4
0x1132fc:    lea     rax, [rcx + 0x49]
0x113300:    lea     rdx, [rcx + rdx + 0x99]
0x113308:    nop     dword ptr [rax + rax]
0x113310:    mov     byte ptr [rax], 0
0x113313:    add     rax, 0x50
0x113317:    cmp     rdx, rax
0x11331a:    jne     0x113310
0x1130ab:    lea     rax, [rsp + 0x30]
0x1130b0:    mov     dword ptr [rsp + 0x10], 1
0x1130b8:    mov     ebx, 0x50
0x1130bd:    mov     qword ptr [rsp + 0x18], rax
0x1130c2:    cmp     ecx, 1

```

```

0x1130c5:      jg      0x1130e8
0x113074:      sub     edx, 1
0x113077:      mov     rsi, qword ptr [rbp + 0x38]
0x11307b:      lea     rdx, [rdx + rdx*4]
0x11307f:      shl     rdx, 4
0x113083:      lea     rax, [rsi + 0x49]
0x113087:      lea     rdx, [rsi + rdx + 0x99]
0x11308f:      nop
0x113090:      cmp     byte ptr [rax], 0
0x113093:      jne     0x1130a2
0x113055:      cmp     byte ptr [rax - 1], 0
0x113059:      je      0x113064
0x113064:      add     rax, 0x50
0x113068:      cmp     rax, rdx
0x11306b:      jne     0x113050
0x11335c:      mov     dword ptr [rbp + 0x70], 0
0x113363:      xor     eax, eax
0x113365:      mov     rdi, r12
0x113368:      call    0x110ec0
0x11332c:      sub     edx, 1
0x11332f:      mov     rcx, qword ptr [rbp + 0x38]
0x113333:      lea     rdx, [rdx + rdx*4]
0x113337:      shl     rdx, 4
0x11333b:      lea     rax, [rcx + 0x49]
0x11333f:      lea     rdx, [rcx + rdx + 0x99]
0x113347:      nop     word ptr [rax + rax]
0x113350:      mov     byte ptr [rax], 0
0x113353:      add     rax, 0x50
0x113357:      cmp     rdx, rax
0x11335a:      jne     0x113350
0x113310:      mov     byte ptr [rax], 0
0x113313:      add     rax, 0x50
0x113317:      cmp     rdx, rax
0x11331a:      jne     0x113310
0x1130c7:      jmp     0x113188
0x1130e8:      mov     rax, qword ptr [r12 + 0x38]
0x1130ed:      cmp     byte ptr [rax + rbx + 0x48], 1
0x1130f2:      jne     0x1130d0
0x113095:      cmp     byte ptr [rax - 1], 0
0x113099:      je      0x1130a2
0x1130a2:      add     rax, 0x50
0x1130a6:      cmp     rax, rdx
0x1130a9:      jne     0x113090
0x11305b:      mov     byte ptr [rax], 1
0x11305e:      add     dword ptr [r12 + 0x70], 1
0x113064:      add     rax, 0x50
0x113068:      cmp     rax, rdx
0x11306b:      jne     0x113050

```

```

0x113050:    cmp     byte ptr [rax], 0
0x113053:    jne     0x113064
0x11336d:    xor     eax, eax
0x11336f:    mov     rdi, r12
0x113372:    call    0x110160
0x113350:    mov     byte ptr [rax], 0
0x113353:    add     rax, 0x50
0x113357:    cmp     rdx, rax
0x11335a:    jne     0x113350
0x113188:    xor     eax, eax
0x11318a:    call    0x10e4e0
0x1130f4:    cmp     dword ptr [rbp + 0x68], 1
0x1130f8:    jle     0x1130d0
0x1130d0:    add     dword ptr [rsp + 0x10], 1
0x1130d5:    add     rbx, 0x50
0x1130d9:    mov     eax, dword ptr [rsp + 0x10]
0x1130dd:    cmp     dword ptr [r12 + 0x68], eax
0x1130e2:    jle     0x113188
0x11309b:    mov     byte ptr [rax], 1
0x11309e:    add     dword ptr [rbp + 0x70], 1
0x1130a2:    add     rax, 0x50
0x1130a6:    cmp     rax, rdx
0x1130a9:    jne     0x113090
0x113090:    cmp     byte ptr [rax], 0
0x113093:    jne     0x1130a2
0x113377:    cmp     byte ptr [r12 + 0x6d], 0
0x11337d:    jne     0x113431
0x11318f:    xor     eax, eax
0x113191:    mov     rdi, r12
0x113194:    call    0x110ec0
0x1130fa:    mov     r14d, 0x50
0x113100:    mov     r13d, 1
0x113106:    jmp     0x11311e
0x113383:    xor     eax, eax
0x113385:    mov     rdi, r12
0x113388:    call    0x110240
0x113431:    mov     rdi, r12
0x113434:    xor     eax, eax
0x113436:    call    0x10ec50
0x113199:    xor     eax, eax
0x11319b:    mov     rdi, r12
0x11319e:    call    0x110160
0x11311e:    xor     eax, eax
0x113120:    call    0x118ab0
0x11338d:    cmp     byte ptr [r12 + 0x6d], 0
0x113393:    jne     0x113422
0x11343b:    jmp     0x113383
0x1131a3:    cmp     byte ptr [r12 + 0x6d], 0

```



```

0x1131a9:    jne     0x113290
0x113125:    mov     r15, qword ptr [rbp + 0x38]
0x113129:    add     r15, r14
0x11312c:    cmp     byte ptr [r15 + 0x48], 1
0x113131:    jne     0x113110
0x113399:    xor     eax, eax
0x11339b:    mov     rdi, rbp
0x11339e:    call    0x110ec0
0x113422:    mov     rdi, r12
0x113425:    xor     eax, eax
0x113427:    call    0x110d70
0x1131af:    xor     eax, eax
0x1131b1:    mov     rdi, r12
0x1131b4:    call    0x110240
0x113290:    mov     rdi, r12
0x113293:    xor     eax, eax
0x113295:    call    0x10ec50
0x113133:    mov     rdx, qword ptr [r12 + 0x38]
0x113138:    mov     rsi, qword ptr [r15]
0x11313b:    add     rdx, rbx
0x11313e:    mov     rdi, qword ptr [rdx]
0x113141:    mov     qword ptr [rsp + 8], rdx
0x113146:    call    0x104b30
0x113110:    add     r13d, 1
0x113114:    add     r14, 0x50
0x113118:    cmp     dword ptr [rbp + 0x68], r13d
0x11311c:    jle     0x1130d0
0x1133a3:    xor     eax, eax
0x1133a5:    mov     rdi, rbp
0x1133a8:    call    0x110160
0x11342c:    jmp     0x113399
0x1131b9:    cmp     byte ptr [r12 + 0x6d], 0
0x1131bf:    jne     0x113280
0x11329a:    jmp     0x1131af
0x11314b:    mov     rdx, qword ptr [rsp + 8]
0x113150:    test    eax, eax
0x113152:    jne     0x113110
0x1133ad:    cmp     byte ptr [rbp + 0x6d], 0
0x1133b1:    jne     0x113402
0x1131c5:    xor     eax, eax
0x1131c7:    mov     rdi, rbp
0x1131ca:    call    0x110ec0
0x113280:    mov     rdi, r12
0x113283:    xor     eax, eax
0x113285:    call    0x110d70
0x113154:    mov     rax, qword ptr [r15 + 8]
0x113158:    cmp     qword ptr [rdx + 8], rax
0x11315c:    je      0x113440

```

```

0x1133b3:    xor     eax, eax
0x1133b5:    mov     rdi, rbp
0x1133b8:    call    0x110240
0x113402:    xor     eax, eax
0x113404:    mov     rdi, rbp
0x113407:    call    0x10ec50
0x1131cf:    xor     eax, eax
0x1131d1:    mov     rdi, rbp
0x1131d4:    call    0x110160
0x11328a:    jmp     0x1131c5
0x113440:    mov     ecx, dword ptr [rsp + 0x14]
0x113444:    test    ecx, ecx
0x113446:    je      0x113468
0x113162:    mov     rsi, qword ptr [rdx + 0x20]
0x113166:    mov     rax, qword ptr [r15 + 0x20]
0x11316a:    cmp     rsi, rax
0x11316d:    jl      0x1133f3
0x1133bd:    cmp     byte ptr [rbp + 0x6d], 0
0x1133c1:    je      0x11320e
0x11340c:    xor     eax, eax
0x11340e:    mov     rdi, rbp
0x113411:    call    0x110240
0x1131d9:    cmp     byte ptr [rbp + 0x6d], 0
0x1131dd:    jne     0x113258
0x113468:    mov     r9, qword ptr [rsp + 0x18]
0x11346d:    mov     esi, dword ptr [rsp + 0x10]
0x113471:    xor     eax, eax
0x113473:    lea     rdx, [rsp + 0x28]
0x113478:    mov     r8d, r13d
0x11347b:    mov     rcx, rbp
0x11347e:    mov     rdi, r12
0x113481:    call    0x1125c0
0x113448:    mov     rsi, qword ptr [rdx + 0x20]
0x11344c:    mov     rax, qword ptr [r15 + 0x20]
0x113450:    cmp     rsi, rax
0x113453:    jne     0x11316a
0x1133f3:    mov     byte ptr [rdx + 0x49], 0
0x1133f7:    sub     dword ptr [r12 + 0x70], 1
0x1133fd:    jmp     0x1130d0
0x113173:    mov     byte ptr [r15 + 0x49], 0
0x113178:    sub     dword ptr [rbp + 0x70], 1
0x11317c:    jmp     0x1130d0
0x11320e:    mov     rax, qword ptr [rsp + 0x38]
0x113213:    sub     rax, qword ptr fs:[0x28]
0x11321c:    jne     0x1134c3
0x1133c7:    mov     rax, qword ptr [rsp + 0x38]
0x1133cc:    sub     rax, qword ptr fs:[0x28]
0x1133d5:    jne     0x1134c3

```

```

0x113416:    cmp     byte ptr [rbp + 0x6d], 0
0x11341a:    je      0x11320e
0x1131df:    xor     eax, eax
0x1131e1:    mov     rdi, rbp
0x1131e4:    call    0x110240
0x113258:    xor     eax, eax
0x11325a:    mov     rdi, rbp
0x11325d:    call    0x10ec50
0x113486:    cmp     rax, -1
0x11348a:    je      0x113188
0x113459:    mov     byte ptr [rdx + 0x49], 0
0x11345d:    sub     dword ptr [r12 + 0x70], 1
0x113463:    jmp     0x113173
0x11316a:    cmp     rsi, rax
0x11316d:    jl      0x1133f3
0x113222:    add     rsp, 0x48
0x113226:    pop     rbx
0x113227:    pop     rbp
0x113228:    pop     r12
0x11322a:    pop     r13
0x11322c:    pop     r14
0x11322e:    pop     r15
0x113230:    ret
0x1134c3:    call    0x1049d0
0x1133db:    add     rsp, 0x48
0x1133df:    mov     rdi, rbp
0x1133e2:    xor     eax, eax
0x1133e4:    pop     rbx
0x1133e5:    pop     rbp
0x1133e6:    pop     r12
0x1133e8:    pop     r13
0x1133ea:    pop     r14
0x1133ec:    pop     r15
0x1133ee:    jmp     0x110d70
0x113420:    jmp     0x1133c7
0x1131e9:    cmp     byte ptr [rbp + 0x6d], 0
0x1131ed:    jne     0x113238
0x113262:    xor     eax, eax
0x113264:    mov     rdi, rbp
0x113267:    call    0x110240
0x113490:    mov     rdx, qword ptr [r12 + 0x38]
0x113495:    add     rdx, rbx
0x113498:    cmp     qword ptr [rsp + 0x28], rax
0x11349d:    je      0x1134b3
0x1131ef:    xor     eax, eax
0x1131f1:    call    0x10a820
0x113238:    mov     rdi, rbp
0x11323b:    xor     eax, eax

```

```

0x11323d:    call    0x110d70
0x11326c:    cmp     byte ptr [rbp + 0x6d], 0
0x113270:    je      0x1131ef
0x1134b3:    cmp     qword ptr [rsp + 0x30], rax
0x1134b8:    jne     0x11349f
0x11349f:    add     r14, qword ptr [rbp + 0x38]
0x1134a3:    mov     rsi, qword ptr [rdx + 0x20]
0x1134a7:    mov     rax, qword ptr [r14 + 0x20]
0x1134ab:    mov     r15, r14
0x1134ae:    jmp     0x11316a
0x1131f6:    mov     edx, dword ptr [r12 + 0x70]
0x1131fb:    test    edx, edx
0x1131fd:    jne     0x1132a0
0x113242:    xor     eax, eax
0x113244:    call    0x10a820
0x113276:    jmp     0x113238
0x1134ba:    add     r14, qword ptr [rbp + 0x38]
0x1134be:    mov     r15, r14
0x1134c1:    jmp     0x113459
0x113203:    mov     eax, dword ptr [rbp + 0x70]
0x113206:    test    eax, eax
0x113208:    jne     0x1132a0
0x1132a0:    mov     rax, qword ptr [rsp + 0x38]
0x1132a5:    sub     rax, qword ptr fs:[0x28]
0x1132ae:    jne     0x1134c3
0x113249:    mov     edx, dword ptr [r12 + 0x70]
0x11324e:    test    edx, edx
0x113250:    je      0x113203
0x1132b4:    add     rsp, 0x48
0x1132b8:    xor     eax, eax
0x1132ba:    pop     rbx
0x1132bb:    pop     rbp
0x1132bc:    pop     r12
0x1132be:    pop     r13
0x1132c0:    pop     r14
0x1132c2:    pop     r15
0x1132c4:    jmp     0x10a5a0
0x113252:    jmp     0x1132a0
0x1130cc:    nop     dword ptr [rax]
0x113108:    nop     dword ptr [rax + rax]
0x113181:    nop     dword ptr [rax]
0x113231:    nop     dword ptr [rax]
0x113254:    nop     dword ptr [rax]
0x113278:    nop     dword ptr [rax + rax]
0x11328f:    nop
0x11329f:    nop
0x1132c9:    nop     dword ptr [rax]
0x1134c8:    nop     dword ptr [rax + rax]

```

```

0x1134d0:      endbr64
0x1134d4:      push     r15
0x1134d6:      movsxd  rsi, esi
0x1134d9:      mov     r15d, edx
0x1134dc:      push     r14
0x1134de:      push     r13
0x1134e0:      lea     r13, [rsi + rsi*4]
0x1134e4:      push     r12
0x1134e6:      shl     r13, 4
0x1134ea:      mov     r12, rdi
0x1134ed:      push     rbp
0x1134ee:      push     rbx
0x1134ef:      sub     rsp, 8
0x1134f3:      mov     rax, qword ptr [rdi + 0x38]
0x1134f7:      mov     r14, qword ptr [rax + r13]
0x1134fb:      mov     rdi, r14
0x1134fe:      call    0x11d7a0
0x113503:      mov     rdi, r14
0x113506:      mov     rbp, rax
0x113509:      call    0x1049b0
0x11350e:      mov     rbx, rax
0x113511:      test    r15d, r15d
0x113514:      jne     0x113580
0x113516:      test    eax, eax
0x113518:      jle     0x113543
0x113580:      test    eax, eax
0x113582:      jle     0x113543
0x113543:      mov     rsi, rbp
0x113546:      mov     rdi, r14
0x113549:      call    0x104b30
0x11351a:      call    0x104ef0
0x113584:      call    0x1047c0
0x11354e:      test    eax, eax
0x113550:      jne     0x1135ba
0x11351f:      lea     edi, [rbx - 1]
0x113522:      xor     edx, edx
0x113524:      nop     dword ptr [rax]
0x113528:      movsx   rsi, byte ptr [r14 + rdx]
0x11352d:      mov     rcx, qword ptr [rax]
0x113530:      mov     ecx, dword ptr [rcx + rsi*4]
0x113533:      mov     byte ptr [rbp + rdx], cl
0x113537:      mov     rcx, rdx
0x11353a:      add     rdx, 1
0x11353e:      cmp     rdi, rcx
0x113541:      jne     0x113528
0x113589:      lea     edi, [rbx - 1]
0x11358c:      xor     edx, edx
0x11358e:      nop

```

```

0x113590:    movsx    rsi, byte ptr [r14 + rdx]
0x113595:    mov      rcx, qword ptr [rax]
0x113598:    mov      ecx, dword ptr [rcx + rsi*4]
0x11359b:    mov      byte ptr [rbp + rdx], cl
0x11359f:    mov      rcx, rdx
0x1135a2:    add      rdx, 1
0x1135a6:    cmp      rcx, rdi
0x1135a9:    jne      0x113590
0x113552:    mov      rax, qword ptr [r12 + 0x38]
0x113557:    mov      byte ptr [rax + r13 + 0x49], 0
0x11355d:    mov      eax, 1
0x113562:    sub      dword ptr [r12 + 0x70], 1
0x113568:    add      rsp, 8
0x11356c:    pop      rbx
0x11356d:    pop      rbp
0x11356e:    pop      r12
0x113570:    pop      r13
0x113572:    pop      r14
0x113574:    pop      r15
0x113576:    ret
0x1135ba:    lea      edi, [rbx + 0x21]
0x1135bd:    mov      ebx, 1
0x1135c2:    movsxd   rdi, edi
0x1135c5:    call     0x11d670
0x113528:    movsx    rsi, byte ptr [r14 + rdx]
0x11352d:    mov      rcx, qword ptr [rax]
0x113530:    mov      ecx, dword ptr [rcx + rsi*4]
0x113533:    mov      byte ptr [rbp + rdx], cl
0x113537:    mov      rcx, rdx
0x11353a:    add      rdx, 1
0x11353e:    cmp      rdi, rcx
0x113541:    jne      0x113528
0x1135ab:    mov      rsi, rbp
0x1135ae:    mov      rdi, r14
0x1135b1:    call     0x104b30
0x113590:    movsx    rsi, byte ptr [r14 + rdx]
0x113595:    mov      rcx, qword ptr [rax]
0x113598:    mov      ecx, dword ptr [rcx + rsi*4]
0x11359b:    mov      byte ptr [rbp + rdx], cl
0x11359f:    mov      rcx, rdx
0x1135a2:    add      rdx, 1
0x1135a6:    cmp      rcx, rdi
0x1135a9:    jne      0x113590
0x1135ca:    mov      r8, r14
0x1135cd:    mov      esi, 1
0x1135d2:    lea      rcx, [rip + 0xdd5b]
0x1135d9:    mov      r15, rax
0x1135dc:    mov      rdi, rax

```

```

0x1135df:    mov     rdx, -1
0x1135e6:    xor     eax, eax
0x1135e8:    call    0x104f10
0x1135b6:    test    eax, eax
0x1135b8:    je      0x113552
0x1135ed:    mov     edx, 1
0x1135f2:    mov     esi, 1
0x1135f7:    mov     rdi, r15
0x1135fa:    call    0x10e480
0x1135ff:    xor     eax, eax
0x113601:    call    0x10a820
0x113606:    mov     rdi, r15
0x113609:    call    0x118d10
0x11360e:    mov     rax, qword ptr [r12 + 0x38]
0x113613:    cmp     byte ptr [rax + r13 + 0x48], 0
0x113619:    je      0x113698
0x113698:    xor     esi, esi
0x11369a:    mov     rdi, rbp
0x11369d:    call    0x104d90
0x11361b:    mov     rsi, rbp
0x11361e:    mov     rdi, r14
0x113621:    call    0x104e10
0x1136a2:    test    eax, eax
0x1136a4:    jne     0x113732
0x113626:    test    eax, eax
0x113628:    je      0x1137a5
0x1136aa:    cmp     dword ptr [r12 + 0x90], 1
0x1136b3:    je      0x11378b
0x113732:    mov     rsi, rbp
0x113735:    mov     rdi, r14
0x113738:    call    0x104dc0
0x1137a5:    mov     rdi, r14
0x1137a8:    call    0x104870
0x11362e:    call    0x104860
0x11378b:    xor     eax, eax
0x11378d:    mov     rsi, rbp
0x113790:    mov     rdi, r12
0x113793:    call    0x111220
0x1136b9:    mov     rdi, rbp
0x1136bc:    call    0x1049b0
0x11373d:    test    eax, eax
0x11373f:    je      0x113552
0x1137ad:    test    eax, eax
0x1137af:    je      0x113552
0x113633:    mov     r15d, dword ptr [rax]
0x113636:    xor     eax, eax
0x113638:    call    0x10e4e0
0x113798:    test    eax, eax

```

```

0x11379a:    je     0x1136b9
0x1136c1:    lea    rdi, [rax + 0x21]
0x1136c5:    call   0x11d670
0x113745:    add    rsp, 8
0x113749:    mov    eax, 6
0x11374e:    pop    rbx
0x11374f:    pop    rbp
0x113750:    pop    r12
0x113752:    pop    r13
0x113754:    pop    r14
0x113756:    pop    r15
0x113758:    ret
0x1137b5:    mov    eax, 4
0x1137ba:    jmp     0x113568
0x11363d:    xor    eax, eax
0x11363f:    call   0x10a820
0x1137a0:    jmp     0x113568
0x1136ca:    mov    rdx, -1
0x1136d1:    mov    r8, rbp
0x1136d4:    lea    rcx, [rip + 0xdc6c]
0x1136db:    mov    r15, rax
0x1136de:    mov    rdi, rax
0x1136e1:    mov    esi, 1
0x1136e6:    xor    eax, eax
0x1136e8:    call   0x104f10
0x113568:    add    rsp, 8
0x11356c:    pop    rbx
0x11356d:    pop    rbp
0x11356e:    pop    r12
0x113570:    pop    r13
0x113572:    pop    r14
0x113574:    pop    r15
0x113576:    ret
0x113644:    cmp    r15d, 1
0x113648:    sete   al
0x11364b:    cmp    r15d, 0xd
0x11364f:    sete   dl
0x113652:    or     al, dl
0x113654:    jne     0x113745
0x1136ed:    mov    esi, 1
0x1136f2:    mov    rdi, r15
0x1136f5:    call   0x1193f0
0x11365a:    and    ebx, 1
0x11365d:    je     0x113745
0x1136fa:    mov    rdi, r15
0x1136fd:    mov    ebx, eax
0x1136ff:    call   0x118d10
0x113663:    cmp    dword ptr [r12 + 0x90], 1

```



```

0x11366c:    je      0x113760
0x113704:    mov     rdi, qword ptr [r12 + 0x18]
0x113709:    call    0x10c4b0
0x113760:    xor     eax, eax
0x113762:    mov     rsi, rbp
0x113765:    mov     rdi, r12
0x113768:    call    0x111220
0x113672:    mov     rdi, rbp
0x113675:    call    0x104870
0x11370e:    mov     edx, ebx
0x113710:    mov     eax, 7
0x113715:    and     edx, 0x7f
0x113718:    add     edx, 1
0x11371b:    cmp     dl, 1
0x11371e:    jg      0x113568
0x11376d:    test    eax, eax
0x11376f:    je      0x113672
0x11367a:    mov     ebx, eax
0x11367c:    test    eax, eax
0x11367e:    je      0x11361b
0x113724:    test    bl, 0x7f
0x113727:    jne     0x113680
0x113775:    jmp     0x113568
0x113680:    add     rsp, 8
0x113684:    mov     eax, 5
0x113689:    pop     rbx
0x11368a:    pop     rbp
0x11368b:    pop     r12
0x11368d:    pop     r13
0x11368f:    pop     r14
0x113691:    pop     r15
0x113693:    ret
0x11372d:    and     bh, 0xff
0x113730:    jne     0x11377a
0x11377a:    lea     rdi, [rip + 0xd469]
0x113781:    call    0x1195e0
0x113786:    jmp     0x113680
0x113577:    nop     word ptr [rax + rax]
0x113694:    nop     dword ptr [rax]
0x113759:    nop     dword ptr [rax]
0x1137bf:    nop
0x1137c0:    endbr64
0x1137c4:    push    r15
0x1137c6:    push    r14
0x1137c8:    mov     r14, rdi
0x1137cb:    push    r13
0x1137cd:    push    r12
0x1137cf:    mov     r12, rdx

```

```

0x1137d2:    push    rbp
0x1137d3:    push    rbx
0x1137d4:    sub     rsp, 0x118
0x1137db:    mov     rax, qword ptr fs:[0x28]
0x1137e4:    mov     qword ptr [rsp + 0x108], rax
0x1137ec:    xor     eax, eax
0x1137ee:    test    rdx, rdx
0x1137f1:    je      0x113be4
0x113be4:    mov     rdi, qword ptr [rsi + 0x18]
0x113be8:    lea     rdx, [rsp + 0x20]
0x113bed:    xor     esi, esi
0x113bef:    mov     qword ptr [rsp + 0x28], -1
0x113bf8:    call    0x11c490
0x1137f7:    movsxd  rcx, dword ptr [r14 + 0x68]
0x1137fb:    test    ecx, ecx
0x1137fd:    jle     0x11382c
0x113bfd:    test    eax, eax
0x113bff:    js      0x1137f7
0x11382c:    mov     dword ptr [r14 + 0x70], 0
0x113834:    lea     rdi, [rcx*8]
0x11383c:    call    0x11d670
0x1137ff:    lea     edx, [rcx - 1]
0x113802:    mov     rsi, qword ptr [r14 + 0x38]
0x113806:    lea     rdx, [rdx + rdx*4]
0x11380a:    shl     rdx, 4
0x11380e:    lea     rax, [rsi + 0x49]
0x113812:    lea     rdx, [rsi + rdx + 0x99]
0x11381a:    nop     word ptr [rax + rax]
0x113820:    mov     byte ptr [rax], 0
0x113823:    add     rax, 0x50
0x113827:    cmp     rax, rdx
0x11382a:    jne     0x113820
0x113c05:    cmp     qword ptr [rsp + 0x28], -1
0x113c0b:    je      0x1137f7
0x113841:    mov     qword ptr [rsp], rax
0x113845:    movsxd  rax, dword ptr [r14 + 0x68]
0x113849:    test    eax, eax
0x11384b:    jle     0x113c59
0x113820:    mov     byte ptr [rax], 0
0x113823:    add     rax, 0x50
0x113827:    cmp     rax, rdx
0x11382a:    jne     0x113820
0x113c11:    call    0x104a60
0x113c59:    mov     rdi, qword ptr [rsp]
0x113c5d:    lea     rcx, [rip - 0x5724]
0x113c64:    xor     esi, esi
0x113c66:    xor     r13d, r13d
0x113c69:    mov     edx, 8

```

```

0x113c6e:    xor     r15d, r15d
0x113c71:    call    0x1048e0
0x113851:    lea     rdi, [rax + rax*4]
0x113855:    xor     r8d, r8d
0x113858:    xor     eax, eax
0x11385a:    xor     r13d, r13d
0x11385d:    shl     rdi, 4
0x113861:    mov     r9d, 0x400
0x113867:    jmp     0x113879
0x113c16:    mov     r12, qword ptr [rsp + 0x30]
0x113c1b:    test    eax, eax
0x113c1d:    cmovne  r12, qword ptr [rsp + 0x38]
0x113c23:    imul    r12, qword ptr [rsp + 0x20]
0x113c29:    shr     r12, 0xa
0x113c2d:    jmp     0x1137f7
0x113c76:    mov     dword ptr [rsp + 0x10], 0
0x113c7e:    xor     r9d, r9d
0x113c81:    jmp     0x1139ea
0x113879:    mov     rcx, qword ptr [r14 + 0x38]
0x11387d:    add     rcx, rax
0x113880:    mov     edx, dword ptr [rcx + 0x10]
0x113883:    and     edx, 0xf000
0x113889:    cmp     edx, 0x8000
0x11388f:    jne     0x113870
0x1139ea:    xor     eax, eax
0x1139ec:    mov     rdi, r14
0x1139ef:    mov     qword ptr [rsp + 8], r9
0x1139f4:    call    0x110ec0
0x113891:    mov     rdx, qword ptr [rcx + 8]
0x113895:    test    edx, 0x3ff
0x11389b:    je      0x1138be
0x113870:    add     rax, 0x50
0x113874:    cmp     rdi, rax
0x113877:    je      0x1138f8
0x1139f9:    xor     eax, eax
0x1139fb:    mov     rdi, r14
0x1139fe:    call    0x110160
0x1138be:    test    rdx, rdx
0x1138c1:    lea     rsi, [rdx + 0x3ff]
0x1138c8:    cmovs   rdx, rsi
0x1138cc:    sar     rdx, 0xa
0x1138d0:    cmp     r12, rdx
0x1138d3:    jl      0x113af0
0x11389d:    mov     r10, rdx
0x1138a0:    sar     r10, 0x3f
0x1138a4:    shr     r10, 0x36
0x1138a8:    lea     rsi, [rdx + r10]
0x1138ac:    and     esi, 0x3ff

```

```

0x1138b2:    sub     rsi, r10
0x1138b5:    mov     r10, r9
0x1138b8:    sub     r10, rsi
0x1138bb:    add     rdx, r10
0x1138be:    test    rdx, rdx
0x1138c1:    lea     rsi, [rdx + 0x3ff]
0x1138c8:    cmovs   rdx, rsi
0x1138cc:    sar     rdx, 0xa
0x1138d0:    cmp     r12, rdx
0x1138d3:    jl      0x113af0
0x1138f8:    mov     rbx, qword ptr [rsp]
0x1138fc:    movsxd  rsi, r13d
0x1138ff:    mov     edx, 8
0x113904:    lea     rcx, [rip - 0x53cb]
0x11390b:    mov     dword ptr [rsp + 0x10], r8d
0x113910:    mov     rdi, rbx
0x113913:    call    0x1048e0
0x113a03:    cmp     byte ptr [r14 + 0x6d], 0
0x113a08:    mov     r9, qword ptr [rsp + 8]
0x113a0d:    je      0x113a1e
0x113af0:    add     r8d, 1
0x113af4:    jmp     0x113870
0x1138d9:    mov     rbx, qword ptr [rsp]
0x1138dd:    movsxd  rdx, r13d
0x1138e0:    add     rax, 0x50
0x1138e4:    add     r13d, 1
0x1138e8:    mov     qword ptr [rbx + rdx*8], rcx
0x1138ec:    cmp     rdi, rax
0x1138ef:    jne     0x113879
0x113918:    test    r13d, r13d
0x11391b:    je      0x113c86
0x113a1e:    xor     eax, eax
0x113a20:    mov     rdi, r14
0x113a23:    mov     qword ptr [rsp + 8], r9
0x113a28:    call    0x110240
0x113a0f:    mov     rdi, r14
0x113a12:    xor     eax, eax
0x113a14:    call    0x10ec50
0x1138f1:    nop     dword ptr [rax]
0x113c86:    xor     r15d, r15d
0x113c89:    xor     r9d, r9d
0x113c8c:    jmp     0x1139ea
0x113921:    lea     eax, [r13 - 1]
0x113925:    mov     r8d, r13d
0x113928:    xor     r15d, r15d
0x11392b:    xor     r11d, r11d
0x11392e:    lea     r10, [rbx + rax*8 + 8]
0x113933:    mov     rax, r12

```

```

0x113936:      xor     r9d, r9d
0x113939:      mov     edx, 0x400
0x11393e:      mov     r13, r10
0x113941:      mov     r12, rbx
0x113944:      mov     rcx, rax
0x113947:      nop     word ptr [rax + rax]
0x113950:      mov     rdi, qword ptr [r12]
0x113954:      mov     rax, qword ptr [rdi + 8]
0x113958:      test    eax, 0x3ff
0x11395d:      je       0x113980
0x113a2d:      cmp     byte ptr [r14 + 0x6d], 0
0x113a32:      mov     r9, qword ptr [rsp + 8]
0x113a37:      je       0x113a48
0x113a19:      mov     r9, qword ptr [rsp + 8]
0x113a1e:      xor     eax, eax
0x113a20:      mov     rdi, r14
0x113a23:      mov     qword ptr [rsp + 8], r9
0x113a28:      call    0x110240
0x113980:      test    rax, rax
0x113983:      lea     rbp, [rax + 0x3ff]
0x11398a:      cmovns  rbp, rax
0x11398e:      sar     rbp, 0xa
0x113992:      test    r15d, r15d
0x113995:      jle     0x113af9
0x11395f:      mov     r10, rax
0x113962:      sar     r10, 0x3f
0x113966:      shr     r10, 0x36
0x11396a:      lea     rsi, [rax + r10]
0x11396e:      and     esi, 0x3ff
0x113974:      sub     rsi, r10
0x113977:      mov     r10, rdx
0x11397a:      sub     r10, rsi
0x11397d:      add     rax, r10
0x113980:      test    rax, rax
0x113983:      lea     rbp, [rax + 0x3ff]
0x11398a:      cmovns  rbp, rax
0x11398e:      sar     rbp, 0xa
0x113992:      test    r15d, r15d
0x113995:      jle     0x113af9
0x113a48:      test    r9, r9
0x113a4b:      je       0x113a55
0x113a39:      mov     rdi, r14
0x113a3c:      xor     eax, eax
0x113a3e:      call    0x110d70
0x113af9:      xor     ebx, ebx
0x113afb:      nop     dword ptr [rax + rax]
0x113b00:      cmp     r15d, ebx
0x113b03:      jne     0x1139d7

```

```

0x11399b:    mov     rax, r9
0x11399e:    xor     ebx, ebx
0x1139a0:    jmp     0x1139b8
0x113a55:    mov     rdi, qword ptr [rsp]
0x113a59:    call    0x118d10
0x113a4d:    mov     rdi, r9
0x113a50:    call    0x118d10
0x113a43:    mov     r9, qword ptr [rsp + 8]
0x113a48:    test    r9, r9
0x113a4b:    je      0x113a55
0x1139d7:    add     r12, 8
0x1139db:    cmp     r13, r12
0x1139de:    jne     0x113950
0x113b09:    nop     dword ptr [rax]
0x1139b8:    mov     rsi, qword ptr [rax]
0x1139bb:    cmp     rsi, rbp
0x1139be:    jl      0x1139a8
0x113a5e:    test    r13d, r13d
0x113a61:    je      0x113bad
0x1139e4:    mov     r13d, r8d
0x1139e7:    mov     r12, rcx
0x1139ea:    xor     eax, eax
0x1139ec:    mov     rdi, r14
0x1139ef:    mov     qword ptr [rsp + 8], r9
0x1139f4:    call    0x110ec0
0x113950:    mov     rdi, qword ptr [r12]
0x113954:    mov     rax, qword ptr [rdi + 8]
0x113958:    test    eax, 0x3ff
0x11395d:    je      0x113980
0x113b10:    cmp     r11d, r15d
0x113b13:    je      0x113b40
0x1139a8:    add     ebx, 1
0x1139ab:    add     rax, 8
0x1139af:    cmp     ebx, r15d
0x1139b2:    je      0x113b10
0x1139c0:    sub     rsi, rbp
0x1139c3:    mov     qword ptr [rax], rsi
0x1139c6:    test    ebx, ebx
0x1139c8:    jne     0x113b00
0x113bad:    mov     eax, dword ptr [rsp + 0x10]
0x113bb1:    lea     rbp, [rsp + 0x60]
0x113bb6:    test    eax, eax
0x113bb8:    je      0x113c32
0x113a67:    mov     eax, dword ptr [rsp + 0x10]
0x113a6b:    lea     rbp, [rsp + 0x60]
0x113a70:    test    eax, eax
0x113a72:    je      0x113b7b
0x113b40:    add     r11d, 0x10

```

```

0x113b44:    mov     rdi, r9
0x113b47:    mov     qword ptr [rsp + 0x18], rcx
0x113b4c:    movsxd  rsi, r11d
0x113b4f:    mov     dword ptr [rsp + 0x14], r8d
0x113b54:    shl     rsi, 3
0x113b58:    mov     dword ptr [rsp + 8], r11d
0x113b5d:    call    0x11d690
0x113b15:    mov     rsi, rcx
0x113b18:    movsxd  rax, r15d
0x113b1b:    add     r15d, 1
0x113b1f:    sub     rsi, rbp
0x113b22:    mov     qword ptr [r9 + rax*8], rsi
0x113b26:    test    ebx, ebx
0x113b28:    jne     0x1139d7
0x1139ce:    mov     byte ptr [rdi + 0x49], 1
0x1139d2:    add     dword ptr [r14 + 0x70], 1
0x1139d7:    add     r12, 8
0x1139db:    cmp     r13, r12
0x1139de:    jne     0x113950
0x113b00:    cmp     r15d, ebx
0x113b03:    jne     0x1139d7
0x113c32:    mov     edx, 0xa0
0x113c37:    mov     esi, 1
0x113c3c:    mov     rdi, rbp
0x113c3f:    xor     eax, eax
0x113c41:    lea     r8, [rip + 0xd70d]
0x113c48:    lea     rcx, [rip + 0xd729]
0x113c4f:    call    0x104f10
0x113bba:    mov     r9, r12
0x113bbd:    lea     r8, [rip + 0xd791]
0x113bc4:    mov     rdi, rbp
0x113bc7:    xor     eax, eax
0x113bc9:    lea     rcx, [rip + 0xe048]
0x113bd0:    mov     edx, 0xa0
0x113bd5:    mov     esi, 1
0x113bda:    call    0x104f10
0x113b7b:    sub     rsp, 8
0x113b7f:    lea     rcx, [rip + 0xe05a]
0x113b86:    mov     r9d, r13d
0x113b89:    xor     eax, eax
0x113b8b:    push    r15
0x113b8d:    mov     edx, 0xa0
0x113b92:    mov     esi, 1
0x113b97:    mov     rdi, rbp
0x113b9a:    lea     r8, [rip + 0xd7b4]
0x113ba1:    call    0x104f10
0x113a78:    sub     rsp, 8
0x113a7c:    mov     r9d, r13d

```

```

0x113a7f:    mov     edx, 0xa0
0x113a84:    mov     rdi, rbp
0x113a87:    push    rax
0x113a88:    lea     rax, [rip + 0xd8d2]
0x113a8f:    mov     esi, 1
0x113a94:    lea     r8, [rip + 0xd8ba]
0x113a9b:    push    r15
0x113a9d:    lea     rcx, [rip + 0xe104]
0x113aa4:    push    rax
0x113aa5:    xor     eax, eax
0x113aa7:    call    0x104f10
0x113b62:    mov     rcx, qword ptr [rsp + 0x18]
0x113b67:    mov     r8d, dword ptr [rsp + 0x14]
0x113b6c:    mov     edx, 0x400
0x113b71:    mov     r11d, dword ptr [rsp + 8]
0x113b76:    mov     r9, rax
0x113b79:    jmp     0x113b15
0x113b2e:    mov     rax, qword ptr [r12]
0x113b32:    mov     byte ptr [rax + 0x49], 1
0x113b36:    add     dword ptr [r14 + 0x70], 1
0x113b3b:    jmp     0x1139d7
0x113c54:    jmp     0x113ab0
0x113bdf:    jmp     0x113ab0
0x113ba6:    pop     rdx
0x113ba7:    pop     rcx
0x113ba8:    jmp     0x113ab0
0x113aac:    add     rsp, 0x20
0x113ab0:    xor     esi, esi
0x113ab2:    mov     edx, 0x36
0x113ab7:    mov     rdi, rbp
0x113aba:    call    0x108070
0x113ab0:    xor     esi, esi
0x113ab2:    mov     edx, 0x36
0x113ab7:    mov     rdi, rbp
0x113aba:    call    0x108070
0x113abf:    mov     rax, qword ptr [rsp + 0x108]
0x113ac7:    sub     rax, qword ptr fs:[0x28]
0x113ad0:    jne     0x113c91
0x113ad6:    add     rsp, 0x118
0x113add:    pop     rbx
0x113ade:    pop     rbp
0x113adf:    pop     r12
0x113ae1:    pop     r13
0x113ae3:    pop     r14
0x113ae5:    pop     r15
0x113ae7:    ret
0x113c91:    call    0x1049d0
0x113869:    nop     dword ptr [rax]

```



```

0x1139a2:      nop      word ptr [rax + rax]
0x113ae8:      nop      dword ptr [rax + rax]
0x113c96:      nop      word ptr cs:[rax + rax]
0x113ca0:      endbr64
0x113ca4:      push     r15
0x113ca6:      xor      r15d, r15d
0x113ca9:      push     r14
0x113cab:      push     r13
0x113cad:      push     r12
0x113caf:      xor      r12d, r12d
0x113cb2:      push     rbp
0x113cb3:      mov      rbp, rdi
0x113cb6:      push     rbx
0x113cb7:      sub      rsp, 0x18
0x113cbb:      call     0x1049b0
0x113cc0:      lea      rdi, [rax + 1]
0x113cc4:      call     0x11d670
0x113cc9:      mov      esi, 8
0x113cce:      mov      edi, 2
0x113cd3:      mov      r13, rax
0x113cd6:      call     0x11d740
0x113cdb:      mov      r14, rax
0x113cde:      xor      eax, eax
0x113ce0:      movzx    ebx, byte ptr [rbp]
0x113ce4:      add      rbp, 1
0x113ce8:      cmp      bl, 0x5c
0x113ceb:      je       0x113d25
0x113d25:      test     r12d, r12d
0x113d28:      je       0x113d36
0x113ced:      jg       0x113cf9
0x113d36:      xor      r12d, 1
0x113d3a:      jmp      0x113ce0
0x113d2a:      movsxd   rcx, eax
0x113d2d:      add      eax, 1
0x113d30:      mov      byte ptr [r13 + rcx], 0x5c
0x113d36:      xor      r12d, 1
0x113d3a:      jmp      0x113ce0
0x113cef:      cmp      bl, 9
0x113cf2:      je       0x113d40
0x113cf9:      test     r12d, r12d
0x113cfc:      je       0x113d0a
0x113ce0:      movzx    ebx, byte ptr [rbp]
0x113ce4:      add      rbp, 1
0x113ce8:      cmp      bl, 0x5c
0x113ceb:      je       0x113d25
0x113d40:      test     bl, bl
0x113d42:      je       0x113d60
0x113cf4:      test     bl, 0xdf

```

```

0x113cf7:      je      0x113d40
0x113d0a:      movsxd  rcx, eax
0x113d0d:      add     rbp, 1
0x113d11:      add     eax, 1
0x113d14:      xor     r12d, r12d
0x113d17:      mov     byte ptr [r13 + rcx], bl
0x113d1c:      movzx   ebx, byte ptr [rbp - 1]
0x113d20:      cmp     bl, 0x5c
0x113d23:      jne     0x113ced
0x113cfe:      movsxd  rcx, eax
0x113d01:      add     eax, 1
0x113d04:      mov     byte ptr [r13 + rcx], 0x5c
0x113d0a:      movsxd  rcx, eax
0x113d0d:      add     rbp, 1
0x113d11:      add     eax, 1
0x113d14:      xor     r12d, r12d
0x113d17:      mov     byte ptr [r13 + rcx], bl
0x113d1c:      movzx   ebx, byte ptr [rbp - 1]
0x113d20:      cmp     bl, 0x5c
0x113d23:      jne     0x113ced
0x113d60:      test    eax, eax
0x113d62:      je      0x113da6
0x113d44:      test    r12b, 1
0x113d48:      je      0x113d60
0x113da6:      test    bl, bl
0x113da8:      je      0x113dd0
0x113d64:      cdqe
0x113d66:      movsxd  rdx, r15d
0x113d69:      mov     rdi, r13
0x113d6c:      mov     byte ptr [r13 + rax], 0
0x113d72:      lea     eax, [r15 + 1]
0x113d76:      lea     r15, [rdx*8]
0x113d7e:      lea     rcx, [r14 + r15]
0x113d82:      mov     dword ptr [rsp + 4], eax
0x113d86:      mov     qword ptr [rsp + 8], rcx
0x113d8b:      call    0x11d7a0
0x113d4a:      movsxd  rcx, eax
0x113d4d:      mov     r12d, 1
0x113d53:      add     eax, 1
0x113d56:      mov     byte ptr [r13 + rcx], bl
0x113d5b:      jmp     0x113ce0
0x113dd0:      mov     rdi, r13
0x113dd3:      call    0x118d10
0x113daa:      lea     esi, [r15 + 2]
0x113dae:      mov     rdi, r14
0x113db1:      movsxd  rsi, esi
0x113db4:      shl     rsi, 3
0x113db8:      call    0x11d690

```

```

0x113d90:    mov     rcx, qword ptr [rsp + 8]
0x113d95:    mov     qword ptr [rcx], rax
0x113d98:    mov     qword ptr [r14 + r15 + 8], 0
0x113da1:    mov     r15d, dword ptr [rsp + 4]
0x113da6:    test    bl, bl
0x113da8:    je      0x113dd0
0x113dd8:    add     rsp, 0x18
0x113ddc:    mov     rax, r14
0x113ddf:    pop     rbx
0x113de0:    pop     rbp
0x113de1:    pop     r12
0x113de3:    pop     r13
0x113de5:    pop     r14
0x113de7:    pop     r15
0x113de9:    ret
0x113dbd:    mov     r14, rax
0x113dc0:    xor     eax, eax
0x113dc2:    jmp     0x113ce0
0x113d3c:    nop     dword ptr [rax]
0x113d5d:    nop     dword ptr [rax]
0x113dc7:    nop     word ptr [rax + rax]
0x113dea:    nop     word ptr [rax + rax]
0x113df0:    endbr64
0x113df4:    push    rbp
0x113df5:    mov     rbp, rdi
0x113df8:    push    rbx
0x113df9:    sub     rsp, 8
0x113dfd:    mov     rdi, qword ptr [rdi]
0x113e00:    test    rdi, rdi
0x113e03:    je      0x113e22
0x113e22:    add     rsp, 8
0x113e26:    mov     rdi, rbp
0x113e29:    pop     rbx
0x113e2a:    pop     rbp
0x113e2b:    jmp     0x118d10
0x113e05:    mov     rbx, rbp
0x113e08:    nop     dword ptr [rax + rax]
0x113e10:    call    0x118d10
0x113e15:    mov     rdi, qword ptr [rbx + 8]
0x113e19:    add     rbx, 8
0x113e1d:    test    rdi, rdi
0x113e20:    jne     0x113e10
0x113e10:    call    0x118d10
0x113e30:    endbr64
0x113e34:    push    r15
0x113e36:    push    r14
0x113e38:    push    r13
0x113e3a:    push    r12

```

```

0x113e3c:    push    rbp
0x113e3d:    push    rbx
0x113e3e:    sub     rsp, 0x48
0x113e42:    mov     rax, qword ptr fs:[0x28]
0x113e4b:    mov     qword ptr [rsp + 0x38], rax
0x113e50:    xor     eax, eax
0x113e52:    cmp     esi, 0x33
0x113e55:    ja      0x114857
0x113e5b:    mov     r12, rdx
0x113e5e:    mov     eax, esi
0x113e60:    lea     rdx, [rip + 0xda9d]
0x113e67:    mov     rbp, rdi
0x113e6a:    movsxd  rax, dword ptr [rdx + rax*4]
0x113e6e:    mov     r15d, esi
0x113e71:    mov     r13, rcx
0x113e74:    add     rax, rdx
0x113e77:    jmp     rax
0x114857:    lea     rdi, [rip + 0xcb3a]
0x11485e:    call    0x108b00
0x113e80:    sub     r15d, 0x10
0x113e84:    movsxd  rax, dword ptr [rbp + 0x28]
0x113e88:    mov     rdx, qword ptr [rbp + 0x18]
0x113e8c:    mov     dword ptr [rbp + 0x84], r15d
0x113e93:    mov     rdi, qword ptr [rbp + 0x38]
0x113e97:    mov     dword ptr [rip + 0x1778a], r15d
0x113e9e:    lea     rax, [rax + rax*4]
0x113ea2:    shl     rax, 4
0x113ea6:    cmp     byte ptr [rdx], 0x2f
0x113ea9:    mov     rbx, qword ptr [rdi + rax]
0x113ead:    mov     eax, dword ptr [rbp + 0x68]
0x113eb0:    jne     0x113ebc
0x114402:    mov     rsi, r12
0x114405:    xor     eax, eax
0x114407:    call    0x114e20
0x114202:    movzx   edx, byte ptr [rdi + 0x6d]
0x114206:    mov     ebx, r8d
0x114209:    xor     r13d, r13d
0x11420c:    xor     r12d, r12d
0x11420f:    test    r8d, r8d
0x114212:    jne     0x114249
0x114187:    mov     rax, qword ptr [rsp + 0x38]
0x11418c:    sub     rax, qword ptr fs:[0x28]
0x114195:    jne     0x114ce9
0x114287:    mov     rsi, r12
0x11428a:    xor     eax, eax
0x11428c:    call    0x115910
0x11438c:    mov     eax, dword ptr [rdi + 0x60]
0x11438f:    mov     edx, dword ptr [r12 + 0x60]

```

```

0x114394:    mov     dword ptr [rdi + 0x60], edx
0x114397:    mov     edx, dword ptr [r12 + 0x64]
0x11439c:    mov     dword ptr [r12 + 0x60], eax
0x1143a1:    mov     eax, dword ptr [rdi + 0x64]
0x1143a4:    mov     dword ptr [rdi + 0x64], edx
0x1143a7:    mov     edx, dword ptr [r12 + 0x58]
0x1143ac:    mov     dword ptr [r12 + 0x64], eax
0x1143b1:    mov     eax, dword ptr [rdi + 0x58]
0x1143b4:    mov     dword ptr [rdi + 0x58], edx
0x1143b7:    mov     edx, dword ptr [r12 + 0x5c]
0x1143bc:    mov     dword ptr [r12 + 0x58], eax
0x1143c1:    mov     eax, dword ptr [rdi + 0x5c]
0x1143c4:    mov     dword ptr [rdi + 0x5c], edx
0x1143c7:    mov     dword ptr [r12 + 0x5c], eax
0x1143cc:    movsxd  rcx, dword ptr [rdi + 0x60]
0x1143d0:    movsxd  rdx, dword ptr [rdi + 0x5c]
0x1143d4:    movsxd  rsi, dword ptr [rdi + 0x58]
0x1143d8:    movsxd  r8, dword ptr [rdi + 0x64]
0x1143dc:    call    0x10f590
0x11437d:    mov     rsi, r12
0x114380:    xor     eax, eax
0x114382:    call    0x112ca0
0x11460e:    mov     ecx, dword ptr [rdi + 0x28]
0x114611:    test    ecx, ecx
0x114613:    je      0x113f50
0x114411:    mov     eax, dword ptr [rdi + 0x84]
0x114417:    add     eax, 1
0x11441a:    movsxd  r15, eax
0x11441d:    cdq
0x11441e:    imul    r15, r15, 0x38e38e39
0x114425:    sar     r15, 0x21
0x114429:    sub     r15d, edx
0x11442c:    lea     edx, [r15 + r15*8]
0x114430:    sub     eax, edx
0x114432:    mov     r15d, eax
0x114435:    jmp     0x113e84
0x114296:    mov     edx, dword ptr [rcx]
0x114298:    mov     rsi, r12
0x11429b:    xor     eax, eax
0x11429d:    call    0x1167a0
0x113f18:    sub     r15d, 9
0x113f1c:    cmp     dword ptr [rbp + 0x64], 0x4f
0x113f20:    mov     dword ptr [rbp + 0x80], r15d
0x113f27:    jg      0x113f40
0x114622:    mov     r9d, dword ptr [rdi + 0x68]
0x114626:    mov     r10d, dword ptr [rdi + 0x28]
0x11462a:    lea     eax, [r9 - 1]
0x11462e:    cmp     r10d, eax

```

```

0x114631:      je      0x113f50
0x1146a4:      mov     r8d, dword ptr [rdi + 0x28]
0x1146a8:      test    r8d, r8d
0x1146ab:      je      0x113f50
0x114525:      mov     rdi, qword ptr [rcx + 8]
0x114529:      mov     qword ptr [rsp + 8], rdi
0x11452e:      call    0x1049b0
0x1142a7:      mov     edx, dword ptr [rcx]
0x1142a9:      mov     rsi, r12
0x1142ac:      xor     eax, eax
0x1142ae:      call    0x112fd0
0x1145b2:      xor     esi, esi
0x1145b4:      mov     edx, 0xa
0x1145b9:      mov     rdi, rcx
0x1145bc:      call    0x104ba0
0x1141b6:      mov     edx, dword ptr [rdi + 0x68]
0x1141b9:      test    edx, edx
0x1141bb:      jle     0x1141ec
0x1142b8:      mov     rdi, qword ptr [rdi + 0x48]
0x1142bc:      call    0x119120
0x11443a:      mov     eax, dword ptr [rdi + 0x80]
0x114440:      add     eax, 1
0x114443:      movsxd   r15, eax
0x114446:      cdq
0x114447:      imul    r15, r15, 0x2aaaaaab
0x11444e:      shr     r15, 0x20
0x114452:      sub     r15d, edx
0x114455:      lea     edx, [r15 + r15*2]
0x114459:      add     edx, edx
0x11445b:      sub     eax, edx
0x11445d:      mov     r15d, eax
0x114460:      jmp     0x113f1c
0x1144ba:      mov     rsi, rcx
0x1144bd:      xor     eax, eax
0x1144bf:      call    0x1161d0
0x1144c9:      mov     rdx, rcx
0x1144cc:      mov     rsi, r12
0x1144cf:      xor     eax, eax
0x1144d1:      call    0x1165f0
0x114349:      sub     dword ptr [rdi + 0x74], r8d
0x11434d:      mov     eax, dword ptr [rdi + 0x74]
0x114350:      xor     edx, edx
0x114352:      test    eax, eax
0x114354:      cmovs    eax, edx
0x114357:      jmp     0x114313
0x1142ce:      mov     eax, dword ptr [rdi + 0x30]
0x1142d1:      mov     edx, dword ptr [rdi + 0x64]
0x1142d4:      mov     esi, 4

```

```

0x1142d9:    add     r8d, dword ptr [rdi + 0x74]
0x1142dd:    add     eax, 1
0x1142e0:    cmp     edx, 0x27
0x1142e3:    jle     0x114300
0x113f78:    movsxd  rax, dword ptr [rdi + 0x28]
0x113f7c:    mov     esi, 0x2e
0x113f81:    lea     rax, [rax + rax*4]
0x113f85:    shl     rax, 4
0x113f89:    add     rax, qword ptr [rdi + 0x38]
0x113f8d:    mov     r12, qword ptr [rax]
0x113f90:    mov     rdi, r12
0x113f93:    call    0x104a20
0x114359:    xor     esi, esi
0x11435b:    mov     rdi, rcx
0x11435e:    mov     edx, 0xa
0x114363:    call    0x104ba0
0x1144db:    mov     rcx, qword ptr [rcx + 8]
0x1144df:    mov     rdi, rcx
0x1144e2:    mov     qword ptr [rsp + 8], rcx
0x1144e7:    call    0x1049b0
0x11415f:    mov     rax, qword ptr [rdi + 0x18]
0x114163:    cmp     byte ptr [rax], 0x2f
0x114166:    je      0x1149a4
0x114060:    xor     eax, eax
0x114062:    mov     rdi, rcx
0x114065:    call    0x113ca0
0x114465:    mov     edx, dword ptr [rdi + 0x68]
0x114468:    mov     dword ptr [rdi + 0x70], 0
0x11446f:    test    edx, edx
0x114471:    jle     0x114132
0x1147e7:    mov     rsi, r12
0x1147ea:    xor     eax, eax
0x1147ec:    call    0x115400
0x1145ea:    mov     eax, dword ptr [rdi + 0x68]
0x1145ed:    sub     eax, 1
0x1145f0:    cmp     dword ptr [rdi + 0x28], eax
0x1145f3:    je      0x113f50
0x1146f2:    movzx   edi, byte ptr [rdi + 0x6d]
0x1146f6:    mov     ebx, r8d
0x1146f9:    xor     r13d, r13d
0x1146fc:    xor     r12d, r12d
0x1146ff:    test    r8d, r8d
0x114702:    jne     0x114739
0x1147f6:    mov     edx, dword ptr [rdi + 0x68]
0x1147f9:    test    edx, edx
0x1147fb:    jle     0x114132
0x114576:    mov     edi, 0x10
0x11457b:    call    0x119100

```

```

0x114278:      mov     rsi, r12
0x11427b:      xor     eax, eax
0x11427d:      call    0x115bb0
0x11477d:      mov     rax, qword ptr [rdi + 0x18]
0x114781:      mov     esi, dword ptr [rdi + 0x28]
0x114784:      cmp     byte ptr [rax], 0x2f
0x114787:      je      0x11497c
0x113eb2:      cmp     byte ptr [rdx + 1], 0
0x113eb6:      je      0x11498b
0x113ebc:      sub     eax, 1
0x113ebf:      add     rdi, 0x50
0x113ec3:      mov     edx, 0x50
0x113ec8:      movsxd   rsi, eax
0x113ecb:      lea     rcx, [rip - 0x5852]
0x113ed2:      call    0x1048e0
0x11440c:      jmp     0x113f50
0x114214:      jmp     0x11488f
0x114249:      mov     esi, dword ptr [rbp + 0x28]
0x11424c:      test    esi, esi
0x11424e:      je      0x114870
0x11419b:      add     rsp, 0x48
0x11419f:      mov     rsi, r12
0x1141a2:      mov     rdi, rbp
0x1141a5:      xor     eax, eax
0x1141a7:      pop     rbx
0x1141a8:      pop     rbp
0x1141a9:      pop     r12
0x1141ab:      pop     r13
0x1141ad:      pop     r14
0x1141af:      pop     r15
0x1141b1:      jmp     0x1162b0
0x114ce9:      call    0x1049d0
0x114291:      jmp     0x113f50
0x1143e1:      movsxd   rcx, dword ptr [r12 + 0x60]
0x1143e6:      movsxd   rdx, dword ptr [r12 + 0x5c]
0x1143eb:      mov     rdi, r12
0x1143ee:      movsxd   rsi, dword ptr [r12 + 0x58]
0x1143f3:      movsxd   r8, dword ptr [r12 + 0x64]
0x1143f8:      call    0x10f590
0x114387:      jmp     0x113f50
0x113f50:      mov     rax, qword ptr [rsp + 0x38]
0x113f55:      sub     rax, qword ptr fs:[0x28]
0x113f5e:      jne     0x114ce9
0x114619:      mov     dword ptr [rdi + 0x28], 0
0x114620:      jmp     0x1145fc
0x113e84:      movsxd   rax, dword ptr [rbp + 0x28]
0x113e88:      mov     rdx, qword ptr [rbp + 0x18]
0x113e8c:      mov     dword ptr [rbp + 0x84], r15d

```



```

0x113e93:    mov     rdi, qword ptr [rbp + 0x38]
0x113e97:    mov     dword ptr [rip + 0x1778a], r15d
0x113e9e:    lea     rax, [rax + rax*4]
0x113ea2:    shl     rax, 4
0x113ea6:    cmp     byte ptr [rdx], 0x2f
0x113ea9:    mov     rbx, qword ptr [rdi + rax]
0x113ead:    mov     eax, dword ptr [rbp + 0x68]
0x113eb0:    jne     0x113ebc
0x1142a2:    jmp     0x113f50
0x113f29:    cmp     r15d, 5
0x113f2d:    jne     0x113f40
0x113f40:    cmp     byte ptr [rbp + 0x6d], 0
0x113f44:    je      0x113f50
0x114637:    mov     r8d, dword ptr [rdi + 0x60]
0x11463b:    mov     ecx, eax
0x11463d:    sub     ecx, dword ptr [rdi + 0x2c]
0x114640:    lea     edx, [r8 - 2]
0x114644:    cmp     ecx, edx
0x114646:    jl      0x114656
0x1146b1:    mov     r9d, dword ptr [rdi + 0x60]
0x1146b5:    lea     eax, [r9 - 2]
0x1146b9:    cmp     r8d, eax
0x1146bc:    jge     0x1149ca
0x114533:    mov     rdi, qword ptr [rsp + 8]
0x114538:    mov     qword ptr [rsp + 0x10], rax
0x11453d:    mov     r14, rax
0x114540:    mov     eax, dword ptr [r13]
0x114544:    cmp     eax, 2
0x114547:    je      0x114a96
0x1142b3:    jmp     0x113f50
0x1145c1:    test    eax, eax
0x1145c3:    jle     0x113f50
0x1141ec:    cmp     byte ptr [rbp + 0x6d], 0
0x1141f0:    mov     dword ptr [rbp + 0x70], 0
0x1141f7:    je      0x113f50
0x1141bd:    sub     edx, 1
0x1141c0:    mov     rcx, qword ptr [rdi + 0x38]
0x1141c4:    lea     rdx, [rdx + rdx*4]
0x1141c8:    shl     rdx, 4
0x1141cc:    lea     rax, [rcx + 0x49]
0x1141d0:    lea     rdx, [rcx + rdx + 0x99]
0x1141d8:    nop     dword ptr [rax + rax]
0x1141e0:    mov     byte ptr [rax], 0
0x1141e3:    add     rax, 0x50
0x1141e7:    cmp     rdx, rax
0x1141ea:    jne     0x1141e0
0x1142c1:    mov     qword ptr [rbp + 0x48], 0
0x1142c9:    jmp     0x113f50

```

```

0x113f1c:    cmp     dword ptr [rbp + 0x64], 0x4f
0x113f20:    mov     dword ptr [rbp + 0x80], r15d
0x113f27:    jg      0x113f40
0x1144c4:    jmp     0x113f50
0x1144d6:    jmp     0x113f50
0x114313:    mov     dword ptr [rbp + 0x74], eax
0x114316:    mov     rdi, rbp
0x114319:    xor     eax, eax
0x11431b:    call    0x110ec0
0x114300:    sub     edx, esi
0x114302:    sub     eax, edx
0x114304:    mov     edx, 0
0x114309:    cmovs   eax, edx
0x11430c:    cmp     eax, r8d
0x11430f:    cmovg   eax, r8d
0x114313:    mov     dword ptr [rbp + 0x74], eax
0x114316:    mov     rdi, rbp
0x114319:    xor     eax, eax
0x11431b:    call    0x110ec0
0x1142e5:    movsxd  rcx, dword ptr [rdi + 0x80]
0x1142ec:    cmp     edx, 0x49
0x1142ef:    jg      0x1142f6
0x113f98:    mov     rbx, rax
0x113f9b:    test    rax, rax
0x113f9e:    je      0x113f50
0x114368:    mov     rsi, r12
0x11436b:    mov     rdi, rbp
0x11436e:    movsxd  rdx, eax
0x114371:    xor     eax, eax
0x114373:    call    0x1137c0
0x1144ec:    mov     rcx, qword ptr [rsp + 8]
0x1144f1:    mov     qword ptr [rsp + 0x10], rax
0x1144f6:    mov     r14, rax
0x1144f9:    mov     eax, dword ptr [r13]
0x1144fd:    cmp     eax, 2
0x114500:    je      0x114bb4
0x1149a4:    cmp     byte ptr [rax + 1], 0
0x1149a8:    je      0x113f50
0x11416c:    mov     esi, dword ptr [rbp + 0x28]
0x11416f:    test    esi, esi
0x114171:    je      0x114187
0x11406a:    cmp     qword ptr [rax], 0
0x11406e:    mov     r14, rax
0x114071:    mov     r13, rax
0x114074:    je      0x11412a
0x114132:    cmp     byte ptr [rbp + 0x6d], 0
0x114136:    je      0x113f50
0x114477:    sub     edx, 1

```

```

0x11447a:    mov     rcx, qword ptr [rdi + 0x38]
0x11447e:    lea     rdx, [rdx + rdx*4]
0x114482:    shl     rdx, 4
0x114486:    lea     rax, [rcx + 0x49]
0x11448a:    lea     rcx, [rcx + rdx + 0x99]
0x114492:    nop     word ptr [rax + rax]
0x114498:    cmp     byte ptr [rax - 1], 0
0x11449c:    je      0x1144ac
0x1147f1:    jmp     0x113f50
0x1145f9:    mov     dword ptr [rdi + 0x28], eax
0x1145fc:    mov     rdi, rbp
0x1145ff:    xor     eax, eax
0x114601:    call    0x10f110
0x114704:    jmp     0x1148c0
0x114739:    mov     eax, dword ptr [rbp + 0x68]
0x11473c:    mov     r9d, dword ptr [rbp + 0x28]
0x114740:    sub     eax, 1
0x114743:    cmp     r9d, eax
0x114746:    jge     0x1148a0
0x114801:    sub     edx, 1
0x114804:    mov     rcx, qword ptr [rdi + 0x38]
0x114808:    lea     rdx, [rdx + rdx*4]
0x11480c:    shl     rdx, 4
0x114810:    lea     rax, [rcx + 0x49]
0x114814:    lea     rdx, [rcx + rdx + 0x99]
0x11481c:    nop     dword ptr [rax]
0x114820:    cmp     byte ptr [rax], 0
0x114823:    jne     0x114832
0x114580:    lea     rsi, [rsp + 0x20]
0x114585:    mov     qword ptr [rsp + 0x28], 0
0x11458e:    mov     qword ptr [rbp + 0x48], rax
0x114592:    mov     rdi, rax
0x114595:    mov     eax, dword ptr [rbp + 0x28]
0x114598:    mov     dword ptr [rsp + 0x20], eax
0x11459c:    call    0x119150
0x114282:    jmp     0x113f50
0x11497c:    cmp     byte ptr [rax + 1], 0
0x114980:    je      0x114791
0x11478d:    test    esi, esi
0x11478f:    je      0x1147d0
0x11498b:    movsxd  rsi, eax
0x11498e:    lea     rcx, [rip - 0x6315]
0x114995:    mov     edx, 0x50
0x11499a:    call    0x1048e0
0x113ed7:    mov     ecx, dword ptr [rbp + 0x68]
0x113eda:    test    ecx, ecx
0x113edc:    jle     0x114cab
0x11488f:    test    dl, dl

```

```

0x114891:    jne    0x114150
0x114870:    test   r13d, r13d
0x114873:    je     0x114931
0x114254:    lea     eax, [rsi - 1]
0x114257:    mov     dword ptr [rbp + 0x28], eax
0x11425a:    cmp     esi, dword ptr [rbp + 0x2c]
0x11425d:    jne     0x114220
0x1143fd:    jmp     0x113f50
0x113f64:    add     rsp, 0x48
0x113f68:    xor     eax, eax
0x113f6a:    pop     rbx
0x113f6b:    pop     rbp
0x113f6c:    pop     r12
0x113f6e:    pop     r13
0x113f70:    pop     r14
0x113f72:    pop     r15
0x113f74:    ret
0x1145fc:    mov     rdi, rbp
0x1145ff:    xor     eax, eax
0x114601:    call    0x10f110
0x113f2f:    mov     dword ptr [rbp + 0x80], 0
0x113f39:    nop     dword ptr [rax]
0x113f40:    cmp     byte ptr [rbp + 0x6d], 0
0x113f44:    je      0x113f50
0x113f46:    mov     rdi, rbp
0x113f49:    xor     eax, eax
0x113f4b:    call    0x110d70
0x114656:    mov     dword ptr [rbp + 0x28], eax
0x114659:    mov     rdi, rbp
0x11465c:    xor     eax, eax
0x11465e:    sub     r8d, 1
0x114662:    call    0x10f110
0x114648:    mov     ecx, eax
0x11464a:    sub     ecx, r10d
0x11464d:    cmp     edx, ecx
0x11464f:    jg      0x114656
0x1146c2:    mov     dword ptr [rdi + 0x28], 0
0x1146c9:    xor     eax, eax
0x1146cb:    call    0x10f110
0x1149ca:    mov     edx, r8d
0x1149cd:    sub     edx, eax
0x1149cf:    xor     eax, eax
0x1149d1:    mov     dword ptr [rdi + 0x28], edx
0x1149d4:    call    0x10f110
0x114a96:    cmp     byte ptr [rbp + 0x6e], 0
0x114a9a:    je      0x114c0e
0x11454d:    cmp     eax, 3
0x114550:    je      0x1149de

```

```

0x1145c9:      mov     ecx, dword ptr [rbp + 0x60]
0x1145cc:      lea     edx, [rcx - 1]
0x1145cf:      cmp     edx, eax
0x1145d1:      jle     0x113f50
0x1141fd:      jmp     0x11413c
0x1141e0:      mov     byte ptr [rax], 0
0x1141e3:      add     rax, 0x50
0x1141e7:      cmp     rdx, rax
0x1141ea:      jne     0x1141e0
0x114320:      xor     eax, eax
0x114322:      mov     rdi, rbp
0x114325:      call    0x110160
0x1142f1:      cmp     ecx, 5
0x1142f4:      je      0x114300
0x1142f6:      lea     rsi, [rip + 0xd6e3]
0x1142fd:      mov     esi, dword ptr [rsi + rcx*4]
0x114300:      sub     edx, esi
0x114302:      sub     eax, edx
0x114304:      mov     edx, 0
0x114309:      cmovs   eax, edx
0x11430c:      cmp     eax, r8d
0x11430f:      cmovg   eax, r8d
0x114313:      mov     dword ptr [rbp + 0x74], eax
0x114316:      mov     rdi, rbp
0x114319:      xor     eax, eax
0x11431b:      call    0x110ec0
0x113fa0:      cmp     r12, rax
0x113fa3:      je      0x113f50
0x114378:      jmp     0x113f50
0x114bb4:      xor     ebx, ebx
0x114bb6:      cmp     byte ptr [rbp + 0x6e], 0
0x114bba:      je      0x114c41
0x114506:      cmp     eax, 3
0x114509:      je      0x114aea
0x1149ae:      jmp     0x11416c
0x114173:      mov     dword ptr [rbp + 0x28], 0
0x11417a:      mov     rdi, rbp
0x11417d:      xor     eax, eax
0x11417f:      call    0x10f110
0x11412a:      mov     rdi, r14
0x11412d:      call    0x118d10
0x11407a:      nop     word ptr [rax + rax]
0x11413c:      mov     rdi, rbp
0x11413f:      xor     eax, eax
0x114141:      call    0x110d70
0x1144ac:      add     rax, 0x50
0x1144b0:      cmp     rcx, rax
0x1144b3:      jne     0x114498

```

```

0x11449e:    cmp     byte ptr [rax], 0
0x1144a1:    sete   dl
0x1144a4:    mov     byte ptr [rax], dl
0x1144a6:    movzx   edx, dl
0x1144a9:    add     dword ptr [rbp + 0x70], edx
0x1144ac:    add     rax, 0x50
0x1144b0:    cmp     rcx, rax
0x1144b3:    jne     0x114498
0x114606:    mov     dword ptr [rbp + 0x2c], eax
0x114609:    jmp     0x114132
0x1148c0:    test    dil, dil
0x1148c3:    jne     0x114150
0x11474c:    mov     esi, dword ptr [rbp + 0x2c]
0x11474f:    mov     ecx, dword ptr [rbp + 0x60]
0x114752:    lea     edx, [r9 + 1]
0x114756:    mov     dword ptr [rbp + 0x28], edx
0x114759:    sub     ecx, 2
0x11475c:    sub     edx, esi
0x11475e:    cmp     edx, ecx
0x114760:    jl      0x114710
0x1148a0:    test    r13d, r13d
0x1148a3:    je      0x114956
0x114825:    cmp     byte ptr [rax - 1], 0
0x114829:    je      0x114832
0x114832:    add     rax, 0x50
0x114836:    cmp     rax, rdx
0x114839:    jne     0x114820
0x1145a1:    mov     qword ptr [rbp + 0x40], 0
0x1145a9:    mov     byte ptr [rbp + 0x6e], 0
0x1145ad:    jmp     0x113f50
0x114791:    movsxd  rax, esi
0x114794:    lea     rax, [rax + rax*4]
0x114798:    shl     rax, 4
0x11479c:    add     rax, qword ptr [rbp + 0x38]
0x1147a0:    movzx   edx, byte ptr [rax + 0x49]
0x1147a4:    test    dl, dl
0x1147a6:    sete   byte ptr [rax + 0x49]
0x1147aa:    cmp     dl, 1
0x1147ad:    sbb     eax, eax
0x1147af:    and     eax, 2
0x1147b2:    sub     eax, 1
0x1147b5:    add     dword ptr [rbp + 0x70], eax
0x1147b8:    cmp     byte ptr [rbp + 0x6d], 0
0x1147bc:    je      0x1147d0
0x114986:    jmp     0x11478d
0x1147d0:    xor     ecx, ecx
0x1147d2:    mov     rdx, r12
0x1147d5:    mov     esi, 0x1c

```

```

0x1147da:    mov     rdi, rbp
0x1147dd:    call   0x113e30
0x11499f:    jmp     0x113ed7
0x114cab:    xor     eax, eax
0x114cad:    jmp     0x113f00
0x113ee2:    mov     rdx, qword ptr [rbp + 0x38]
0x113ee6:    xor     eax, eax
0x113ee8:    jmp     0x113efb
0x114897:    jmp     0x113f50
0x114150:    mov     rdi, rbp
0x114153:    xor     eax, eax
0x114155:    call   0x10ec50
0x114931:    test    r12d, r12d
0x114934:    je      0x11488f
0x114879:    test    dl, dl
0x11487b:    je      0x113f50
0x11425f:    sub     esi, dword ptr [rbp + 0x88]
0x114265:    mov     eax, 0
0x11426a:    mov     r13d, 1
0x114270:    cmovs   esi, eax
0x114273:    mov     dword ptr [rbp + 0x2c], esi
0x114276:    jmp     0x114240
0x114220:    test    r12d, r12d
0x114223:    jne     0x114240
0x114667:    mov     dword ptr [rbp + 0x2c], eax
0x11466a:    movzx   eax, byte ptr [rbp + 0x6d]
0x11466e:    cmp     r9d, r8d
0x114671:    jge     0x1148ce
0x114651:    lea     eax, [r10 + r8 - 2]
0x114656:    mov     dword ptr [rbp + 0x28], eax
0x114659:    mov     rdi, rbp
0x11465c:    xor     eax, eax
0x11465e:    sub     r8d, 1
0x114662:    call   0x10f110
0x1146d0:    mov     dword ptr [rbp + 0x2c], eax
0x1146d3:    sub     r9d, 1
0x1146d7:    movzx   eax, byte ptr [rbp + 0x6d]
0x1146db:    cmp     r9d, dword ptr [rbp + 0x68]
0x1146df:    jle     0x1148ce
0x1149d9:    jmp     0x1146d0
0x114c0e:    mov     eax, dword ptr [rbp + 0x28]
0x114c11:    mov     rdi, qword ptr [rbp + 0x48]
0x114c15:    lea     rsi, [rsp + 0x20]
0x114c1a:    mov     qword ptr [rsp + 0x28], r14
0x114c1f:    mov     dword ptr [rsp + 0x20], eax
0x114c23:    call   0x119150
0x114aa0:    mov     eax, dword ptr [rbp + 0x68]
0x114aa3:    lea     ebx, [rax - 1]

```

```

0x114aa6:    test    ebx, ebx
0x114aa8:    js      0x114a80
0x1149de:    mov     eax, dword ptr [rbp + 0x28]
0x1149e1:    mov     rdi, qword ptr [rbp + 0x48]
0x1149e5:    lea     rsi, [rsp + 0x20]
0x1149ea:    mov     qword ptr [rsp + 0x28], r14
0x1149ef:    mov     dword ptr [rsp + 0x20], eax
0x1149f3:    call    0x119150
0x114556:    cmp     eax, 1
0x114559:    je      0x1148db
0x1145d7:    mov     dword ptr [r12 + 0x88], eax
0x1145df:    mov     dword ptr [rbp + 0x88], eax
0x1145e5:    jmp     0x113f50
0x11432a:    cmp     byte ptr [rbp + 0x6d], 0
0x11432e:    je      0x11433a
0x113fa5:    mov     rdi, rax
0x113fa8:    call    0x1049b0
0x114c41:    mov     eax, dword ptr [rbp + 0x28]
0x114c44:    mov     rdi, qword ptr [rbp + 0x48]
0x114c48:    lea     rsi, [rsp + 0x20]
0x114c4d:    mov     qword ptr [rsp + 0x28], r14
0x114c52:    mov     dword ptr [rsp + 0x20], eax
0x114c56:    call    0x119150
0x114bc0:    mov     eax, dword ptr [rbp + 0x68]
0x114bc3:    mov     dword ptr [rsp + 0x1c], eax
0x114bc7:    cmp     ebx, eax
0x114bc9:    jge     0x114ba3
0x114aea:    mov     eax, dword ptr [rbp + 0x28]
0x114aed:    mov     rdi, qword ptr [rbp + 0x48]
0x114af1:    lea     rsi, [rsp + 0x20]
0x114af6:    mov     qword ptr [rsp + 0x28], r14
0x114afb:    mov     dword ptr [rsp + 0x20], eax
0x114aff:    call    0x119150
0x11450f:    cmp     eax, 1
0x114512:    je      0x1148db
0x114184:    mov     dword ptr [rbp + 0x2c], eax
0x114187:    mov     rax, qword ptr [rsp + 0x38]
0x11418c:    sub     rax, qword ptr fs:[0x28]
0x114195:    jne     0x114ce9
0x114080:    mov     edx, dword ptr [rbp + 0x68]
0x114083:    xor     ebx, ebx
0x114085:    test    edx, edx
0x114087:    jle     0x1140f9
0x114146:    cmp     byte ptr [rbp + 0x6d], 0
0x11414a:    je      0x113f50
0x1144b5:    jmp     0x114132
0x114498:    cmp     byte ptr [rax - 1], 0
0x11449c:    je      0x1144ac

```



```

0x1148c9:      jmp      0x113f50
0x114710:      test     r12d, r12d
0x114713:      jne      0x114730
0x114762:      sub      eax, ecx
0x114764:      add      esi, dword ptr [rbp + 0x88]
0x11476a:      mov      r13d, 1
0x114770:      add      eax, 1
0x114773:      cmp      esi, eax
0x114775:      cmovg     esi, eax
0x114778:      mov      dword ptr [rbp + 0x2c], esi
0x11477b:      jmp      0x114730
0x114956:      test     r12d, r12d
0x114959:      je       0x1148c0
0x1148a9:      test     dil, dil
0x1148ac:      je       0x113f50
0x11482b:      mov      byte ptr [rax], 1
0x11482e:      add      dword ptr [rbp + 0x70], 1
0x114832:      add      rax, 0x50
0x114836:      cmp      rax, rdx
0x114839:      jne      0x114820
0x11483b:      jmp      0x114132
0x114820:      cmp      byte ptr [rax], 0
0x114823:      jne      0x114832
0x1147be:      mov      rdi, rbp
0x1147c1:      mov      dword ptr [rsp + 8], r8d
0x1147c6:      call     0x110760
0x1147e2:      jmp      0x113f50
0x113f00:      mov      dword ptr [rbp + 0x28], eax
0x113f03:      mov      rdi, rbp
0x113f06:      xor      eax, eax
0x113f08:      call     0x10f110
0x113efb:      cmp      qword ptr [rdx], rbx
0x113efe:      jne      0x113ef0
0x11415a:      jmp      0x113f50
0x11493a:      mov      esi, dword ptr [rbp + 0x28]
0x11493d:      test     dl, dl
0x11493f:      je       0x113f50
0x114881:      mov      rdi, rbp
0x114884:      xor      eax, eax
0x114886:      call     0x110d70
0x114240:      sub      ebx, 1
0x114243:      je       0x114870
0x114225:      mov      r12d, 1
0x11422b:      test     dl, dl
0x11422d:      je       0x114240
0x114677:      test     al, al
0x114679:      je       0x113f50
0x1148ce:      test     al, al

```

```

0x1148d0:    je      0x113f50
0x1146e5:    mov     esi, r8d
0x1146e8:    test    al, al
0x1146ea:    je      0x113f50
0x114c28:    cmp     byte ptr [rbp + 0x6e], 0
0x114c2c:    jne     0x114cee
0x114a80:    mov     rax, qword ptr [rbp + 0x40]
0x114a84:    mov     dword ptr [r13], 0xffffffff
0x114a8c:    mov     qword ptr [rsp + 0x10], rax
0x114a91:    jmp     0x114568
0x114aaa:    movsxd  rax, ebx
0x114aad:    lea     r15, [rax + rax*4]
0x114ab1:    shl     r15, 4
0x114ab5:    add     r15, qword ptr [rbp + 0x38]
0x114ab9:    jmp     0x114ad1
0x1149f8:    mov     rax, qword ptr [r13 + 8]
0x1149fc:    mov     ebx, dword ptr [rbp + 0x28]
0x1149ff:    mov     qword ptr [rsp + 8], rax
0x114a04:    test    ebx, ebx
0x114a06:    js      0x114a80
0x1148db:    mov     rdi, qword ptr [rbp + 0x48]
0x1148df:    lea     r14, [rsp + 0x20]
0x1148e4:    mov     rsi, r14
0x1148e7:    call    0x1191a0
0x11455f:    mov     rax, qword ptr [rbp + 0x40]
0x114563:    mov     qword ptr [rsp + 0x10], rax
0x114568:    mov     rax, qword ptr [rsp + 0x10]
0x11456d:    mov     qword ptr [r13 + 0x10], rax
0x114571:    jmp     0x113f50
0x11433a:    mov     rdi, rbp
0x11433d:    xor     eax, eax
0x11433f:    call    0x110240
0x114330:    mov     rdi, rbp
0x114333:    xor     eax, eax
0x114335:    call    0x10ec50
0x113fad:    lea     rdi, [rax + 2]
0x113fb1:    call    0x104c20
0x114c5b:    cmp     byte ptr [rbp + 0x6e], 0
0x114c5f:    jne     0x114c67
0x114bcb:    movsxd  rax, ebx
0x114bce:    lea     r15, [rax + rax*4]
0x114bd2:    shl     r15, 4
0x114bd6:    add     r15, qword ptr [rbp + 0x38]
0x114bda:    jmp     0x114bf2
0x114ba3:    mov     dword ptr [r13], 0xffffffff
0x114bab:    mov     rax, qword ptr [rbp + 0x40]
0x114baf:    jmp     0x11451c
0x114b04:    mov     rax, qword ptr [r13 + 8]

```

```

0x114b08:    mov     ebx, dword ptr [rbp + 0x28]
0x114b0b:    mov     qword ptr [rsp + 8], rax
0x114b10:    mov     eax, dword ptr [rbp + 0x68]
0x114b13:    mov     dword ptr [rsp + 0x1c], eax
0x114b17:    cmp     ebx, eax
0x114b19:    jge     0x114ba3
0x114518:    mov     rax, qword ptr [rbp + 0x40]
0x11451c:    mov     qword ptr [r13 + 0x10], rax
0x114520:    jmp     0x113f50
0x1140f9:    add     r13, 8
0x1140fd:    cmp     qword ptr [r13], 0
0x114102:    jne     0x114080
0x114089:    nop     dword ptr [rax]
0x114715:    mov     r12d, 1
0x11471b:    test    dil, dil
0x11471e:    je      0x114730
0x114730:    sub     ebx, 1
0x114733:    je      0x1148a0
0x11495f:    mov     esi, dword ptr [rbp + 0x28]
0x114962:    test    dil, dil
0x114965:    je      0x113f50
0x1148b2:    mov     rdi, rbp
0x1148b5:    xor     eax, eax
0x1148b7:    call    0x110d70
0x113f0d:    mov     dword ptr [rbp + 0x2c], eax
0x113f10:    jmp     0x113f40
0x113ef0:    add     eax, 1
0x113ef3:    add     rdx, 0x50
0x113ef7:    cmp     eax, ecx
0x113ef9:    je      0x113f00
0x114945:    mov     rdi, rbp
0x114948:    call    0x110760
0x11488b:    movzx   edx, byte ptr [rbp + 0x6d]
0x11488f:    test    dl, dl
0x114891:    jne     0x114150
0x11422f:    mov     rdi, rbp
0x114232:    call    0x110760
0x11467f:    mov     esi, r10d
0x114682:    mov     rdi, rbp
0x114685:    call    0x110760
0x1148d6:    jmp     0x11413c
0x1146f0:    jmp     0x114682
0x114c32:    mov     eax, dword ptr [rbp + 0x28]
0x114c35:    mov     rdi, qword ptr [r13 + 8]
0x114c39:    lea     ebx, [rax - 1]
0x114c3c:    jmp     0x114aa6
0x114cee:    mov     rdi, qword ptr [r13 + 8]
0x114cf2:    jmp     0x114aa0

```

```

0x114568:    mov     rax, qword ptr [rsp + 0x10]
0x11456d:    mov     qword ptr [r13 + 0x10], rax
0x114571:    jmp     0x113f50
0x114ad1:    mov     rsi, qword ptr [r15]
0x114ad4:    mov     rdx, r14
0x114ad7:    mov     qword ptr [rsp + 8], rdi
0x114adc:    call    0x104c30
0x114a08:    movsxd  rax, ebx
0x114a0b:    lea     r8, [rax + rax*4]
0x114a0f:    shl     r8, 4
0x114a13:    add     r8, qword ptr [rbp + 0x38]
0x114a17:    mov     r15, r8
0x114a1a:    jmp     0x114a2c
0x1148ec:    mov     edx, dword ptr [rsp + 0x20]
0x1148f0:    mov     rax, qword ptr [rbp + 0x40]
0x1148f4:    cmp     rax, qword ptr [rsp + 0x10]
0x1148f9:    jne     0x11451c
0x114344:    jmp     0x113f40
0x113fb6:    mov     r8, rbx
0x113fb9:    mov     esi, 1
0x113fbe:    xor     ebx, ebx
0x113fc0:    mov     rdi, rax
0x113fc3:    mov     r13, rax
0x113fc6:    xor     eax, eax
0x113fc8:    mov     rdx, -1
0x113fcf:    lea     rcx, [rip + 0xd3be]
0x113fd6:    call    0x104f10
0x114c61:    mov     eax, dword ptr [rbp + 0x28]
0x114c64:    lea     ebx, [rax + 1]
0x114c67:    mov     rcx, qword ptr [r13 + 8]
0x114c6b:    jmp     0x114bc0
0x114c67:    mov     rcx, qword ptr [r13 + 8]
0x114c6b:    jmp     0x114bc0
0x114bf2:    mov     rsi, qword ptr [r15]
0x114bf5:    mov     rdx, r14
0x114bf8:    mov     rdi, rcx
0x114bfb:    mov     qword ptr [rsp + 8], rcx
0x114c00:    call    0x104c30
0x11451c:    mov     qword ptr [r13 + 0x10], rax
0x114520:    jmp     0x113f50
0x114b1f:    movsxd  rax, ebx
0x114b22:    lea     rcx, [rax + rax*4]
0x114b26:    shl     rcx, 4
0x114b2a:    add     rcx, qword ptr [rbp + 0x38]
0x114b2e:    mov     r15, rcx
0x114b31:    jmp     0x114b45
0x114108:    mov     rdi, qword ptr [r14]
0x11410b:    test    rdi, rdi

```

```

0x11410e:    je      0x11412a
0x114090:    mov     rax, qword ptr [rbp + 0x38]
0x114094:    lea     r12, [rbx + rbx*4]
0x114098:    shl     r12, 4
0x11409c:    add     rax, r12
0x11409f:    cmp     byte ptr [rax + 0x48], 0
0x1140a3:    je      0x1140f0
0x114720:    mov     rdi, rbp
0x114723:    mov     esi, r9d
0x114726:    call    0x110760
0x11496b:    mov     rdi, rbp
0x11496e:    call    0x110760
0x1148bc:    movzx   edi, byte ptr [rbp + 0x6d]
0x1148c0:    test    dil, dil
0x1148c3:    jne     0x114150
0x1147cb:    mov     r8d, dword ptr [rsp + 8]
0x1147d0:    xor     ecx, ecx
0x1147d2:    mov     rdx, r12
0x1147d5:    mov     esi, 0x1c
0x1147da:    mov     rdi, rbp
0x1147dd:    call    0x113e30
0x114682:    mov     rdi, rbp
0x114685:    call    0x110760
0x114aa6:    test    ebx, ebx
0x114aa8:    js      0x114a80
0x114ae1:    test    eax, eax
0x114ae3:    jne     0x114ac0
0x114a2c:    mov     rsi, qword ptr [r15]
0x114a2f:    mov     rdi, qword ptr [rsp + 8]
0x114a34:    mov     rdx, r14
0x114a37:    call    0x104c30
0x1148ff:    mov     r8d, dword ptr [rbp + 0x28]
0x114903:    cmp     r8d, edx
0x114906:    jl      0x114cb2
0x113fdb:    mov     eax, dword ptr [rbp + 0x68]
0x113fde:    test    eax, eax
0x113fe0:    jle     0x114132
0x114c05:    test    eax, eax
0x114c07:    jne     0x114be0
0x114b45:    mov     rsi, qword ptr [r15]
0x114b48:    mov     rdi, qword ptr [rsp + 8]
0x114b4d:    mov     rdx, r14
0x114b50:    call    0x104c30
0x114110:    mov     rbx, r14
0x114113:    nop     dword ptr [rax + rax]
0x114118:    call    0x118d10
0x1140f0:    add     rbx, 1
0x1140f4:    cmp     dword ptr [rbp + 0x68], ebx

```

```

0x1140f7:    jg      0x114090
0x1140a5:    cmp     dword ptr [rip + 0x17578], 1
0x1140ac:    mov     rsi, qword ptr [rax]
0x1140af:    sbb     edx, edx
0x1140b1:    mov     rdi, qword ptr [r13]
0x1140b5:    and     edx, 4
0x1140b8:    add     edx, 1
0x1140bb:    call    0x104a40
0x11494d:    movzx   edx, byte ptr [rbp + 0x6d]
0x114951:    jmp     0x11488f
0x114237:    movzx   edx, byte ptr [rbp + 0x6d]
0x11423b:    nop     dword ptr [rax + rax]
0x114240:    sub     ebx, 1
0x114243:    je      0x114870
0x11468a:    cmp     byte ptr [rbp + 0x6d], 0
0x11468e:    mov     esi, dword ptr [rbp + 0x28]
0x114691:    je      0x113f50
0x114697:    mov     rdi, rbp
0x11469a:    call    0x110760
0x114ae5:    jmp     0x114a40
0x114ac0:    sub     ebx, 1
0x114ac3:    sub     r15, 0x50
0x114ac7:    mov     rdi, qword ptr [rsp + 8]
0x114acc:    cmp     ebx, -1
0x114acf:    je      0x114a80
0x114a3c:    test    eax, eax
0x114a3e:    jne     0x114a20
0x114cb2:    sub     edx, r8d
0x114cb5:    xor     ecx, ecx
0x114cb7:    mov     esi, 0x1c
0x114cbc:    mov     rdi, rbp
0x114cbf:    mov     r8d, edx
0x114cc2:    mov     rdx, r12
0x114cc5:    call    0x113e30
0x11490c:    jg      0x114ccf
0x113fe6:    nop     word ptr cs:[rax + rax]
0x114c09:    jmp     0x114b59
0x114be0:    add     ebx, 1
0x114be3:    add     r15, 0x50
0x114be7:    cmp     ebx, dword ptr [rsp + 0x1c]
0x114beb:    mov     rcx, qword ptr [rsp + 8]
0x114bf0:    je      0x114ba3
0x114b55:    test    eax, eax
0x114b57:    jne     0x114b38
0x11411d:    mov     rdi, qword ptr [rbx + 8]
0x114121:    add     rbx, 8
0x114125:    test    rdi, rdi
0x114128:    jne     0x114118

```

```

0x1140c0:    test    eax, eax
0x1140c2:    jne     0x1140f0
0x11472b:    movzx   edi, byte ptr [rbp + 0x6d]
0x11472f:    nop
0x114730:    sub     ebx, 1
0x114733:    je      0x1148a0
0x114973:    movzx   edi, byte ptr [rbp + 0x6d]
0x114977:    jmp     0x1148c0
0x114a40:    cmp     byte ptr [rbp + 0x6e], 0
0x114a44:    mov     qword ptr [rbp + 0x40], r14
0x114a48:    je      0x114c8d
0x114a20:    sub     ebx, 1
0x114a23:    sub     r15, 0x50
0x114a27:    cmp     ebx, -1
0x114a2a:    je      0x114a80
0x114cca:    jmp     0x114912
0x114912:    mov     rdi, qword ptr [rbp + 0x48]
0x114916:    mov     edx, 1
0x11491b:    mov     rsi, r14
0x11491e:    call    0x119200
0x114ccf:    sub     r8d, edx
0x114cd2:    xor     ecx, ecx
0x114cd4:    mov     rdx, r12
0x114cd7:    mov     esi, 0x1b
0x114cdc:    mov     rdi, rbp
0x114cdf:    call    0x113e30
0x113ff0:    mov     rax, qword ptr [rbp + 0x38]
0x113ff4:    lea     r12, [rbx + rbx*4]
0x113ff8:    shl     r12, 4
0x113ffc:    add     rax, r12
0x113fff:    cmp     byte ptr [rax + 0x48], 0
0x114003:    je      0x114048
0x114b59:    cmp     ebx, -1
0x114b5c:    je      0x114ba3
0x114b38:    add     ebx, 1
0x114b3b:    add     r15, 0x50
0x114b3f:    cmp     ebx, dword ptr [rsp + 0x1c]
0x114b43:    je      0x114ba3
0x114118:    call    0x118d10
0x1140c4:    add     r12, qword ptr [rbp + 0x38]
0x1140c8:    movzx   eax, byte ptr [r12 + 0x49]
0x1140ce:    cmp     r15d, 0x24
0x1140d2:    je      0x114840
0x11469f:    jmp     0x114146
0x114c8d:    mov     r8d, dword ptr [rbp + 0x28]
0x114c91:    xor     ecx, ecx
0x114c93:    mov     rdx, r12
0x114c96:    mov     esi, 0x1b

```

```

0x114c9b:    mov     rdi, rbp
0x114c9e:    sub     r8d, ebx
0x114ca1:    call    0x113e30
0x114a4e:    mov     dword ptr [rbp + 0x28], ebx
0x114a51:    xor     eax, eax
0x114a53:    mov     rdi, rbp
0x114a56:    call    0x10f110
0x114923:    mov     rax, qword ptr [rsp + 0x28]
0x114928:    mov     qword ptr [rbp + 0x40], rax
0x11492c:    jmp     0x11451c
0x114ce4:    jmp     0x114912
0x114048:    add     rbx, 1
0x11404c:    cmp     dword ptr [rbp + 0x68], ebx
0x11404f:    jg      0x113ff0
0x114005:    cmp     dword ptr [rip + 0x17618], 1
0x11400c:    mov     rsi, qword ptr [rax]
0x11400f:    mov     rdi, r13
0x114012:    sbb     edx, edx
0x114014:    and     edx, 4
0x114017:    add     edx, 1
0x11401a:    call    0x104a40
0x114b5e:    cmp     byte ptr [rbp + 0x6e], 0
0x114b62:    mov     qword ptr [rbp + 0x40], r14
0x114b66:    je      0x114c70
0x114840:    test    al, al
0x114842:    jne     0x1140f0
0x1140d8:    test    al, al
0x1140da:    je      0x1140f0
0x114ca6:    jmp     0x11455f
0x114a5b:    cmp     byte ptr [rbp + 0x6d], 0
0x114a5f:    mov     dword ptr [rbp + 0x2c], eax
0x114a62:    je      0x114a77
0x114051:    jmp     0x114132
0x11401f:    test    eax, eax
0x114021:    jne     0x114048
0x114c70:    sub     ebx, dword ptr [rbp + 0x28]
0x114c73:    xor     ecx, ecx
0x114c75:    mov     rdx, r12
0x114c78:    mov     esi, 0x1c
0x114c7d:    mov     r8d, ebx
0x114c80:    mov     rdi, rbp
0x114c83:    call    0x113e30
0x114b6c:    mov     dword ptr [rbp + 0x28], ebx
0x114b6f:    xor     eax, eax
0x114b71:    mov     rdi, rbp
0x114b74:    call    0x10f110
0x114848:    mov     byte ptr [r12 + 0x49], 1
0x11484e:    add     dword ptr [rbp + 0x70], 1

```



```

0x114852:      jmp      0x1140f0
0x1140dc:      mov     byte ptr [r12 + 0x49], 0
0x1140e2:      sub     dword ptr [rbp + 0x70], 1
0x1140e6:      nop     word ptr cs:[rax + rax]
0x1140f0:      add     rbx, 1
0x1140f4:      cmp     dword ptr [rbp + 0x68], ebx
0x1140f7:      jg       0x114090
0x114a77:      mov     byte ptr [rbp + 0x6e], 0
0x114a7b:      jmp     0x114568
0x114a64:      xor     eax, eax
0x114a66:      mov     rdi, rbp
0x114a69:      call    0x110d70
0x114023:      add     r12, qword ptr [rbp + 0x38]
0x114027:      movzx   eax, byte ptr [r12 + 0x49]
0x11402d:      cmp     r15d, 0x32
0x114031:      je       0x1149b3
0x114c88:      jmp     0x114518
0x114b79:      cmp     byte ptr [rbp + 0x6d], 0
0x114b7d:      mov     dword ptr [rbp + 0x2c], eax
0x114b80:      je       0x114b95
0x114a6e:      mov     rax, qword ptr [rbp + 0x40]
0x114a72:      mov     qword ptr [rsp + 0x10], rax
0x114a77:      mov     byte ptr [rbp + 0x6e], 0
0x114a7b:      jmp     0x114568
0x1149b3:      test    al, al
0x1149b5:      jne      0x114048
0x114037:      test    al, al
0x114039:      je       0x114048
0x114b95:      mov     byte ptr [rbp + 0x6e], 0
0x114b99:      mov     rax, qword ptr [rsp + 0x10]
0x114b9e:      jmp     0x11451c
0x114b82:      xor     eax, eax
0x114b84:      mov     rdi, rbp
0x114b87:      call    0x110d70
0x1149bb:      mov     byte ptr [r12 + 0x49], 1
0x1149c1:      add     dword ptr [rbp + 0x70], 1
0x1149c5:      jmp     0x114048
0x11403b:      mov     byte ptr [r12 + 0x49], 0
0x114041:      sub     dword ptr [rbp + 0x70], 1
0x114045:      nop     dword ptr [rax]
0x114048:      add     rbx, 1
0x11404c:      cmp     dword ptr [rbp + 0x68], ebx
0x11404f:      jg       0x113ff0
0x114b8c:      mov     rax, qword ptr [rbp + 0x40]
0x114b90:      mov     qword ptr [rsp + 0x10], rax
0x114b95:      mov     byte ptr [rbp + 0x6e], 0
0x114b99:      mov     rax, qword ptr [rsp + 0x10]
0x114b9e:      jmp     0x11451c

```

```

0x113e7a:      nop      word ptr [rax + rax]
0x113eea:      nop      word ptr [rax + rax]
0x113f12:      nop      word ptr [rax + rax]
0x113f75:      nop      dword ptr [rax]
0x114056:      nop      word ptr cs:[rax + rax]
0x114219:      nop      dword ptr [rax]
0x114709:      nop      dword ptr [rax]
0x114863:      jmp      0x113f50
0x114868:      nop      dword ptr [rax + rax]
0x11489c:      nop      dword ptr [rax]
0x114a1c:      nop      dword ptr [rax]
0x114abb:      nop      dword ptr [rax + rax]
0x114b33:      nop      dword ptr [rax + rax]
0x114bdc:      nop      dword ptr [rax]
0x114cf7:      nop      word ptr [rax + rax]
0x114d00:      endbr64
0x114d04:      push     rbp
0x114d05:      xor      eax, eax
0x114d07:      mov      rbp, rdi
0x114d0a:      mov      dword ptr [rdi + 0x28], 0
0x114d11:      call     0x10f110
0x114d16:      mov      rsi, qword ptr [rbp + 0x18]
0x114d1a:      xor      edx, edx
0x114d1c:      mov      ecx, 0x1e
0x114d21:      mov      dword ptr [rbp + 0x2c], eax
0x114d24:      lea      rdi, [rip + 0xc677]
0x114d2b:      xor      eax, eax
0x114d2d:      call     0x10f760
0x114d32:      mov      rax, qword ptr [rbp + 0x18]
0x114d36:      cmp      byte ptr [rax], 0x2f
0x114d39:      jne      0x114d51
0x114d3b:      cmp      byte ptr [rax + 1], 0
0x114d3f:      jne      0x114d51
0x114d51:      mov      edx, 0x2f
0x114d56:      mov      word ptr [rax], dx
0x114d59:      mov      rdi, qword ptr [rbp + 0x18]
0x114d5d:      mov      qword ptr [rbp + 0x78], 1
0x114d65:      call     0x1049c0
0x114d41:      lea      rdi, [rip + 0xc672]
0x114d48:      call     0x108b00
0x114d6a:      mov      rdi, rbp
0x114d6d:      mov      r8d, 1
0x114d73:      xor      ecx, ecx
0x114d75:      xor      edx, edx
0x114d77:      mov      esi, 0x22
0x114d7c:      pop      rbp
0x114d7d:      jmp      0x113e30
0x114d4d:      mov      rax, qword ptr [rbp + 0x18]

```

```

0x114d51:    mov     edx, 0x2f
0x114d56:    mov     word ptr [rax], dx
0x114d59:    mov     rdi, qword ptr [rbp + 0x18]
0x114d5d:    mov     qword ptr [rbp + 0x78], 1
0x114d65:    call    0x1049c0
0x114d82:    nop     word ptr cs:[rax + rax]
0x114d8d:    nop     dword ptr [rax]
0x114d90:    endbr64
0x114d94:    push    rbp
0x114d95:    cmp     byte ptr [rdi + 0x6d], 0
0x114d99:    mov     rbp, rdi
0x114d9c:    mov     eax, esi
0x114d9e:    mov     byte ptr [rdi + 0x6c], sil
0x114da2:    je      0x114db0
0x114db0:    test    al, al
0x114db2:    jne     0x114dc0
0x114da4:    mov     esi, dword ptr [rdi + 0x28]
0x114da7:    call    0x110760
0x114db4:    pop     rbp
0x114db5:    ret
0x114dc0:    mov     rdi, qword ptr [rbp + 0x18]
0x114dc4:    call    0x1049c0
0x114dc9:    cmp     eax, -1
0x114dcc:    jne     0x114db4
0x114dac:    movzx   eax, byte ptr [rbp + 0x6c]
0x114db0:    test    al, al
0x114db2:    jne     0x114dc0
0x114dce:    xor     eax, eax
0x114dd0:    mov     rdi, rbp
0x114dd3:    call    0x114d00
0x114dd8:    xor     eax, eax
0x114dda:    mov     rdi, rbp
0x114ddd:    call    0x110ec0
0x114de2:    xor     eax, eax
0x114de4:    mov     rdi, rbp
0x114de7:    call    0x110160
0x114dec:    cmp     byte ptr [rbp + 0x6d], 0
0x114df0:    jne     0x114e10
0x114df2:    xor     eax, eax
0x114df4:    mov     rdi, rbp
0x114df7:    call    0x110240
0x114e10:    mov     rdi, rbp
0x114e13:    xor     eax, eax
0x114e15:    call    0x10ec50
0x114dfc:    cmp     byte ptr [rbp + 0x6d], 0
0x114e00:    je      0x114db4
0x114e1a:    jmp     0x114df2
0x114e02:    mov     rdi, rbp

```

```

0x114e05:    xor     eax, eax
0x114e07:    pop     rbp
0x114e08:    jmp     0x110d70
0x114db6:    nop     word ptr cs:[rax + rax]
0x114e0d:    nop     dword ptr [rax]
0x114e1c:    nop     dword ptr [rax]
0x114e20:    endbr64
0x114e24:    push    r15
0x114e26:    mov     r15, rsi
0x114e29:    push    r14
0x114e2b:    push    r13
0x114e2d:    push    r12
0x114e2f:    push    rbp
0x114e30:    mov     rbp, rdi
0x114e33:    push    rbx
0x114e34:    sub     rsp, 0x58
0x114e38:    mov     edx, dword ptr [rdi + 0x70]
0x114e3b:    mov     rax, qword ptr fs:[0x28]
0x114e44:    mov     qword ptr [rsp + 0x48], rax
0x114e49:    xor     eax, eax
0x114e4b:    mov     qword ptr [rsp + 0x38], 0
0x114e54:    mov     qword ptr [rsp + 0x40], 0
0x114e5d:    mov     dword ptr [rdi + 0x90], 1
0x114e67:    test    edx, edx
0x114e69:    jne     0x115138
0x114e6f:    movsxd  rax, dword ptr [rdi + 0x28]
0x114e73:    mov     rdx, rax
0x114e76:    lea     rax, [rax + rax*4]
0x114e7a:    shl     rax, 4
0x114e7e:    add     rax, qword ptr [rdi + 0x38]
0x114e82:    mov     r12, qword ptr [rax]
0x114e85:    test    edx, edx
0x114e87:    jne     0x114ea0
0x115138:    mov     rdx, qword ptr [rsi + 0x18]
0x11513c:    mov     rcx, qword ptr [rip + 0x164d5]
0x115143:    lea     rsi, [rsp + 0x38]
0x115148:    lea     rdi, [rip + 0xc28d]
0x11514f:    call    0x1083e0
0x114e89:    mov     rax, qword ptr [rdi + 0x18]
0x114e8d:    cmp     byte ptr [rax], 0x2f
0x114e90:    jne     0x115110
0x114ea0:    mov     rdi, r12
0x114ea3:    call    0x1049b0
0x115154:    test    rax, rax
0x115157:    je      0x115110
0x114e96:    cmp     byte ptr [rax + 1], 0
0x114e9a:    jne     0x115110
0x115110:    mov     rax, qword ptr [rsp + 0x48]

```

```

0x115115:    sub     rax, qword ptr fs:[0x28]
0x11511e:    jne     0x1153ed
0x114ea8:    lea     rdi, [rax + 0x11]
0x114eac:    call    0x11d670
0x115159:    mov     rdi, qword ptr [rbp + 0x18]
0x11515d:    mov     rsi, qword ptr [rsp + 0x38]
0x115162:    xor     eax, eax
0x115164:    call    0x111190
0x115124:    add     rsp, 0x58
0x115128:    pop     rbx
0x115129:    pop     rbp
0x11512a:    pop     r12
0x11512c:    pop     r13
0x11512e:    pop     r14
0x115130:    pop     r15
0x115132:    ret
0x1153ed:    call    0x1049d0
0x114eb1:    mov     edx, 0xe
0x114eb6:    mov     rsi, r12
0x114eb9:    lea     rdi, [rip + 0x16720]
0x114ec0:    mov     r13, rax
0x114ec3:    call    0x104880
0x115169:    test    eax, eax
0x11516b:    jne     0x115308
0x114ec8:    lea     rcx, [rip + 0xc500]
0x114ecf:    mov     esi, 1
0x114ed4:    mov     rdi, r13
0x114ed7:    mov     r8, rax
0x114eda:    mov     rdx, -1
0x114ee1:    xor     eax, eax
0x114ee3:    call    0x104f10
0x115171:    mov     r12, qword ptr [rsp + 0x38]
0x115176:    mov     rdi, r12
0x115179:    call    0x1049b0
0x115308:    mov     edx, 0xc
0x11530d:    xor     esi, esi
0x11530f:    lea     rdi, [rip + 0x14fea]
0x115316:    call    0x108070
0x114ee8:    mov     rdi, r12
0x114eeb:    call    0x1049b0
0x11517e:    mov     rdi, r12
0x115181:    mov     rbx, rax
0x115184:    lea     rax, [rax + 1]
0x115188:    mov     qword ptr [rsp + 0x18], rax
0x11518d:    lea     rax, [rbx + 2]
0x115191:    mov     rsi, rax
0x115194:    mov     qword ptr [rsp + 0x10], rax
0x115199:    call    0x11d690

```

```

0x11531b:      jmp      0x115110
0x114ef0:      mov     rdi, qword ptr [r15 + 0x18]
0x114ef4:      mov     rbx, rax
0x114ef7:      call    0x1049b0
0x11519e:      mov     qword ptr [rsp + 0x38], rax
0x1151a3:      mov     byte ptr [rax + rbx], 0x2f
0x1151a7:      mov     rax, qword ptr [rsp + 0x38]
0x1151ac:      mov     byte ptr [rax + rbx + 1], 0
0x1151b1:      mov     eax, dword ptr [rbp + 0x70]
0x1151b4:      mov     dword ptr [rbp + 0x50], 0xffffffff
0x1151bb:      mov     dword ptr [rbp + 0x54], eax
0x1151be:      mov     rdi, rbp
0x1151c1:      call    0x1100e0
0x114efc:      lea     rdi, [rbx + rax + 2]
0x114f01:      call    0x11d670
0x1151c6:      cmp     eax, -1
0x1151c9:      je      0x1152ce
0x114f06:      mov     r8, qword ptr [r15 + 0x18]
0x114f0a:      mov     r9, r12
0x114f0d:      lea     rcx, [rip + 0xb7ed]
0x114f14:      mov     rdi, rax
0x114f17:      mov     r14, rax
0x114f1a:      mov     esi, 1
0x114f1f:      xor     eax, eax
0x114f21:      mov     rdx, -1
0x114f28:      call    0x104f10
0x1152ce:      mov     rdi, qword ptr [rsp + 0x38]
0x1152d3:      test    rdi, rdi
0x1152d6:      je      0x1152dd
0x1151cf:      cdqe
0x1151d1:      lea     rax, [rax + rax*4]
0x1151d5:      shl     rax, 4
0x1151d9:      mov     qword ptr [rsp + 8], rax
0x1151de:      add     rax, qword ptr [rbp + 0x38]
0x1151e2:      mov     r12, qword ptr [rax]
0x1151e5:      mov     r14d, dword ptr [rax + 0x10]
0x1151e9:      call    0x1049e0
0x114f2d:      mov     rdi, r13
0x114f30:      lea     rsi, [rsp + 0x40]
0x114f35:      mov     rdx, r14
0x114f38:      mov     rcx, qword ptr [rip + 0x166d9]
0x114f3f:      call    0x1083e0
0x1152dd:      xor     eax, eax
0x1152df:      call    0x10e4e0
0x1152d8:      call    0x118d10
0x1151ee:      mov     ebx, eax
0x1151f0:      call    0x104bf0
0x114f44:      mov     rdi, r13

```

```

0x114f47:    test    rax, rax
0x114f4a:    je      0x1153c3
0x1152e4:    xor     eax, eax
0x1152e6:    call   0x10a820
0x1151f5:    mov     rdi, r12
0x1151f8:    mov     r13d, eax
0x1151fb:    call   0x1049b0
0x1153c3:    call   0x118d10
0x114f50:    call   0x118d10
0x1152eb:    jmp     0x11505b
0x115200:    mov     rcx, qword ptr [rsp + 0x10]
0x115205:    mov     rdi, qword ptr [rsp + 0x38]
0x11520a:    lea     rsi, [rcx + rax]
0x11520e:    call   0x11d690
0x1153c8:    jmp     0x115110
0x114f55:    movsxd  rax, dword ptr [rbp + 0x28]
0x114f59:    lea     rax, [rax + rax*4]
0x114f5d:    shl     rax, 4
0x114f61:    add     rax, qword ptr [rbp + 0x38]
0x114f65:    mov     eax, dword ptr [rax + 0x10]
0x114f68:    and     eax, 0xf000
0x114f6d:    cmp     eax, 0x4000
0x114f72:    je      0x1152f0
0x11505b:    mov     rsi, qword ptr [r15 + 0x18]
0x11505f:    xor     eax, eax
0x115061:    mov     edx, 1
0x115066:    mov     rdi, r15
0x115069:    call   0x10f880
0x115213:    mov     rcx, qword ptr [rsp + 0x18]
0x115218:    mov     rsi, r12
0x11521b:    mov     qword ptr [rsp + 0x38], rax
0x115220:    lea     rdi, [rax + rcx]
0x115224:    call   0x1048b0
0x1152f0:    mov     rdi, qword ptr [rip + 0x151a1]
0x1152f7:    call   0x10db60
0x114f78:    mov     rdi, qword ptr [rip + 0x15509]
0x114f7f:    call   0x10db60
0x11506e:    test    eax, eax
0x115070:    je      0x1153e0
0x115229:    mov     eax, dword ptr [rip + 0x16881]
0x11522f:    test    eax, eax
0x115231:    jne     0x115338
0x1152fc:    jmp     0x114f84
0x114f84:    xor     eax, eax
0x114f86:    call   0x10a820
0x1153e0:    mov     rdi, r15
0x1153e3:    call   0x114d00
0x115076:    xor     eax, eax

```

```

0x115078:    mov     rdi, r15
0x11507b:    call   0x110ec0
0x115237:    mov     rdi, qword ptr [rip + 0x15252]
0x11523e:    call   0x10db60
0x115338:    mov     dword ptr [rip + 0x1676e], 0
0x115342:    xor     eax, eax
0x115344:    call   0x10c8c0
0x114f8b:    mov     rdi, qword ptr [rsp + 0x40]
0x114f90:    xor     eax, eax
0x114f92:    call   0x119940
0x1153e8:    jmp     0x115076
0x115080:    xor     eax, eax
0x115082:    mov     rdi, r15
0x115085:    call   0x110160
0x115243:    xor     eax, eax
0x115245:    call   0x10a820
0x115349:    mov     edx, 4
0x11534e:    lea     rsi, [rip + 0xb1dd]
0x115355:    lea     rdi, [rip + 0xbe9d]
0x11535c:    mov     qword ptr [rsp + 0x28], rax
0x115361:    call   0x108070
0x114f97:    mov     rdi, qword ptr [rsp + 0x40]
0x114f9c:    mov     r13, rax
0x114f9f:    call   0x118d10
0x11508a:    cmp     byte ptr [r15 + 0x6d], 0
0x11508f:    je      0x11509b
0x11524a:    mov     rdx, qword ptr [rsp + 0x38]
0x11524f:    xor     eax, eax
0x115251:    mov     r9d, r13d
0x115254:    mov     r8d, ebx
0x115257:    mov     ecx, r14d
0x11525a:    mov     rsi, r12
0x11525d:    mov     rdi, rbp
0x115260:    call   0x111330
0x115366:    mov     rdi, qword ptr [rsp + 0x28]
0x11536b:    mov     byte ptr [rsp + 0x27], al
0x11536f:    call   0x10c970
0x114fa4:    mov     rdi, r12
0x114fa7:    xor     eax, eax
0x114fa9:    mov     rsi, r13
0x114fac:    mov     qword ptr [rsp + 0x40], r13
0x114fb1:    call   0x111190
0x11509b:    xor     eax, eax
0x11509d:    mov     rdi, r15
0x1150a0:    call   0x110240
0x115091:    mov     rdi, r15
0x115094:    xor     eax, eax
0x115096:    call   0x10ec50

```



```

0x115265:    cmp     eax, 1
0x115268:    je      0x115320
0x115374:    xor     eax, eax
0x115376:    call    0x10d870
0x114fb6:    mov     rdi, r14
0x114fb9:    mov     ebx, eax
0x114fbb:    call    0x118d10
0x1150a5:    cmp     byte ptr [r15 + 0x6d], 0
0x1150aa:    je      0x1150b6
0x115320:    mov     rax, qword ptr [rbp + 0x38]
0x115324:    mov     rbx, qword ptr [rsp + 8]
0x115329:    mov     byte ptr [rax + rbx + 0x49], 0
0x11532e:    jmp     0x1152a9
0x11526e:    cmp     eax, 2
0x115271:    je      0x1152ce
0x11537b:    xor     eax, eax
0x11537d:    call    0x10d740
0x114fc0:    test    ebx, ebx
0x114fc2:    jne     0x1153a0
0x1150b6:    mov     rsi, qword ptr [rbp + 0x18]
0x1150ba:    xor     eax, eax
0x1150bc:    mov     edx, 1
0x1150c1:    mov     rdi, rbp
0x1150c4:    call    0x10f880
0x1150ac:    mov     rdi, r15
0x1150af:    xor     eax, eax
0x1150b1:    call    0x110d70
0x1152a9:    xor     eax, eax
0x1152ab:    mov     rdi, rbp
0x1152ae:    call    0x110240
0x115273:    cmp     eax, 3
0x115276:    je      0x1151be
0x115382:    xor     eax, eax
0x115384:    call    0x10a820
0x114fc8:    call    0x104bf0
0x1153a0:    mov     rdx, qword ptr [rsp + 0x40]
0x1153a5:    lea     rdi, [rip + 0xc8a4]
0x1153ac:    xor     ecx, ecx
0x1153ae:    xor     eax, eax
0x1153b0:    mov     r8d, 0x1e
0x1153b6:    mov     rsi, r12
0x1153b9:    call    0x10f7e0
0x1150c9:    test    eax, eax
0x1150cb:    je      0x1153d0
0x1152b3:    mov     rdi, r15
0x1152b6:    xor     eax, eax
0x1152b8:    call    0x110240
0x1151be:    mov     rdi, rbp

```

```

0x1151c1:      call    0x1100e0
0x11527c:      sub     eax, 1
0x11527f:      lea     rbx, [rip + 0x1519a]
0x115286:      xor     ecx, ecx
0x115288:      mov     rsi, r12
0x11528b:      cdqe
0x11528d:      mov     r8d, 0xe
0x115293:      lea     rdi, [rip + 0xc00f]
0x11529a:      mov     rdx, qword ptr [rbx + rax*8]
0x11529e:      xor     eax, eax
0x1152a0:      call    0x10f7e0
0x115389:      movzx   eax, byte ptr [rsp + 0x27]
0x11538e:      and     eax, 0xffffffff
0x115391:      cmp     al, 0x4e
0x115393:      je      0x115237
0x114fcd:      mov     ebx, eax
0x114fcf:      call    0x1049e0
0x1153be:      mov     rdi, qword ptr [rsp + 0x40]
0x1153c3:      call    0x118d10
0x1153d0:      mov     rdi, rbp
0x1153d3:      call    0x114d00
0x1150d1:      xor     eax, eax
0x1150d3:      mov     rdi, rbp
0x1150d6:      call    0x110ec0
0x1152bd:      mov     rdi, rbp
0x1152c0:      call    0x1100e0
0x1152a5:      test    al, al
0x1152a7:      je      0x1152ce
0x115399:      jmp     0x1152ce
0x114fd4:      movsxd  rdx, dword ptr [rbp + 0x28]
0x114fd8:      mov     r9d, ebx
0x114fdb:      mov     rsi, r12
0x114fde:      mov     r8d, eax
0x114fe1:      mov     rdi, rbp
0x114fe4:      xor     eax, eax
0x114fe6:      lea     rdx, [rdx + rdx*4]
0x114fea:      shl     rdx, 4
0x114fee:      add     rdx, qword ptr [rbp + 0x38]
0x114ff2:      mov     ecx, dword ptr [rdx + 0x10]
0x114ff5:      mov     rdx, qword ptr [rsp + 0x40]
0x114ffa:      call    0x111330
0x1153d8:      jmp     0x1150d1
0x1150db:      xor     eax, eax
0x1150dd:      mov     rdi, rbp
0x1150e0:      call    0x110160
0x1152c5:      cmp     eax, -1
0x1152c8:      jne     0x1151cf
0x114fff:      mov     rdi, qword ptr [rsp + 0x40]

```

```

0x115004:    mov     ebx, eax
0x115006:    call   0x118d10
0x1150e5:    cmp     byte ptr [rbp + 0x6d], 0
0x1150e9:    je      0x1150f5
0x11500b:    lea     eax, [rbx - 1]
0x11500e:    cmp     eax, 1
0x115011:    jbe     0x115039
0x1150f5:    xor     eax, eax
0x1150f7:    mov     rdi, rbp
0x1150fa:    call   0x110240
0x1150eb:    mov     rdi, rbp
0x1150ee:    xor     eax, eax
0x1150f0:    call   0x10ec50
0x115039:    xor     eax, eax
0x11503b:    call   0x10e4e0
0x115013:    cdqe
0x115015:    lea     rdx, [rip + 0x15404]
0x11501c:    xor     ecx, ecx
0x11501e:    mov     rsi, r12
0x115021:    mov     rdx, qword ptr [rdx + rax*8]
0x115025:    mov     r8d, 0x1e
0x11502b:    lea     rdi, [rip + 0xc277]
0x115032:    xor     eax, eax
0x115034:    call   0x10f7e0
0x1150ff:    cmp     byte ptr [rbp + 0x6d], 0
0x115103:    je      0x115110
0x115040:    mov     rdi, rbp
0x115043:    xor     eax, eax
0x115045:    call   0x110240
0x115105:    mov     rdi, rbp
0x115108:    xor     eax, eax
0x11510a:    call   0x110d70
0x11504a:    xor     eax, eax
0x11504c:    mov     rdi, r15
0x11504f:    call   0x110240
0x11510f:    nop
0x115054:    xor     eax, eax
0x115056:    call   0x10a820
0x115133:    nop     dword ptr [rax + rax]
0x115301:    nop     dword ptr [rax]
0x115333:    nop     dword ptr [rax + rax]
0x11539e:    nop
0x1153cd:    nop     dword ptr [rax]
0x1153dd:    nop     dword ptr [rax]
0x1153f2:    nop     word ptr cs:[rax + rax]
0x1153fd:    nop     dword ptr [rax]
0x115400:    endbr64
0x115404:    push    r15

```

```

0x115406:    push    r14
0x115408:    push    r13
0x11540a:    push    r12
0x11540c:    push    rbp
0x11540d:    mov     rbp, rdi
0x115410:    push    rbx
0x115411:    sub     rsp, 0x28
0x115415:    mov     qword ptr [rsp + 0x10], rsi
0x11541a:    mov     esi, dword ptr [rdi + 0x70]
0x11541d:    test    esi, esi
0x11541f:    jne     0x115450
0x115421:    mov     ecx, dword ptr [rdi + 0x28]
0x115424:    test    ecx, ecx
0x115426:    jne     0x115450
0x115450:    xor     edx, edx
0x115452:    lea     rsi, [rip + 0xb0d9]
0x115459:    lea     rdi, [rip + 0xbf97]
0x115460:    call    0x108070
0x115428:    mov     rax, qword ptr [rdi + 0x18]
0x11542c:    cmp     byte ptr [rax], 0x2f
0x11542f:    je      0x115440
0x115465:    cmp     al, 0x79
0x115467:    jne     0x115431
0x115440:    cmp     byte ptr [rax + 1], 0
0x115444:    jne     0x115431
0x115431:    add     rsp, 0x28
0x115435:    pop     rbx
0x115436:    pop     rbp
0x115437:    pop     r12
0x115439:    pop     r13
0x11543b:    pop     r14
0x11543d:    pop     r15
0x11543f:    ret
0x115469:    mov     edx, dword ptr [rbp + 0x68]
0x11546c:    test    edx, edx
0x11546e:    jle     0x1158fc
0x115446:    nop     word ptr cs:[rax + rax]
0x1158fc:    mov     dword ptr [rsp + 0x1c], 0
0x115904:    jmp     0x115494
0x115474:    mov     rax, qword ptr [rbp + 0x38]
0x115478:    xor     ecx, ecx
0x11547a:    add     rax, 0x49
0x11547e:    jmp     0x11548b
0x115494:    mov     eax, dword ptr [rbp + 0x70]
0x115497:    mov     dword ptr [rbp + 0x50], 0xffffffff
0x11549e:    xor     r13d, r13d
0x1154a1:    lea     r15, [rip + 0xbf69]
0x1154a8:    mov     dword ptr [rsp + 0x18], 1

```

```

0x1154b0:    mov     dword ptr [rbp + 0x54], eax
0x1154b3:    nop     dword ptr [rax + rax]
0x1154b8:    mov     rdi, rbp
0x1154bb:    call    0x1100e0
0x11548b:    cmp     byte ptr [rax], 0
0x11548e:    je      0x115480
0x1154c0:    cmp     eax, -1
0x1154c3:    je      0x1155c6
0x115480:    add     ecx, 1
0x115483:    add     rax, 0x50
0x115487:    cmp     ecx, edx
0x115489:    je      0x115490
0x115490:    mov     dword ptr [rsp + 0x1c], ecx
0x115494:    mov     eax, dword ptr [rbp + 0x70]
0x115497:    mov     dword ptr [rbp + 0x50], 0xffffffff
0x11549e:    xor     r13d, r13d
0x1154a1:    lea     r15, [rip + 0xbf69]
0x1154a8:    mov     dword ptr [rsp + 0x18], 1
0x1154b0:    mov     dword ptr [rbp + 0x54], eax
0x1154b3:    nop     dword ptr [rax + rax]
0x1154b8:    mov     rdi, rbp
0x1154bb:    call    0x1100e0
0x1155c6:    mov     eax, dword ptr [rsp + 0x1c]
0x1155ca:    cmp     dword ptr [rbp + 0x68], eax
0x1155cd:    je      0x1155e3
0x1154c9:    cdqe
0x1154cb:    lea     rbx, [rax + rax*4]
0x1154cf:    mov     rax, qword ptr [rbp + 0x38]
0x1154d3:    shl     rbx, 4
0x1154d7:    mov     r12, qword ptr [rax + rbx]
0x1154db:    xor     eax, eax
0x1154dd:    call    0x118ab0
0x1155e3:    xor     eax, eax
0x1155e5:    mov     rdi, rbp
0x1155e8:    call    0x110240
0x1155cf:    mov     eax, dword ptr [rsp + 0x1c]
0x1155d3:    mov     rdi, rbp
0x1155d6:    mov     dword ptr [rbp + 0x28], eax
0x1155d9:    xor     eax, eax
0x1155db:    call    0x10f110
0x1154e2:    mov     rdi, r12
0x1154e5:    call    0x1049b0
0x1155ed:    mov     rdi, qword ptr [rsp + 0x10]
0x1155f2:    xor     eax, eax
0x1155f4:    call    0x110240
0x1155e0:    mov     dword ptr [rbp + 0x2c], eax
0x1155e3:    xor     eax, eax
0x1155e5:    mov     rdi, rbp

```

```

0x1155e8:      call    0x110240
0x1154ea:      lea     rdi, [rax + 0x20]
0x1154ee:      call    0x11d670
0x1155f9:      xor     eax, eax
0x1155fb:      call    0x10e4e0
0x1154f3:      mov     r8, r12
0x1154f6:      mov     rcx, r15
0x1154f9:      mov     esi, 1
0x1154fe:      mov     r14, rax
0x115501:      mov     rdi, rax
0x115504:      mov     rdx, -1
0x11550b:      xor     eax, eax
0x11550d:      call    0x104f10
0x115600:      xor     eax, eax
0x115602:      call    0x10a820
0x115512:      mov     edx, 1
0x115517:      mov     esi, 2
0x11551c:      mov     rdi, r14
0x11551f:      call    0x10e480
0x115607:      mov     rsi, qword ptr [rbp + 0x18]
0x11560b:      xor     eax, eax
0x11560d:      mov     edx, 1
0x115612:      mov     rdi, rbp
0x115615:      call    0x10f880
0x115524:      xor     eax, eax
0x115526:      call    0x10a820
0x11561a:      test    eax, eax
0x11561c:      je      0x1158e0
0x11552b:      mov     rdi, r14
0x11552e:      call    0x118d10
0x1158e0:      mov     rdi, rbp
0x1158e3:      call    0x114d00
0x115622:      xor     eax, eax
0x115624:      mov     rdi, rbp
0x115627:      call    0x110ec0
0x115533:      mov     edx, dword ptr [rip + 0x16577]
0x115539:      test    edx, edx
0x11553b:      jne     0x115760
0x1158e8:      jmp     0x115622
0x11562c:      xor     eax, eax
0x11562e:      mov     rdi, rbp
0x115631:      call    0x110160
0x115541:      mov     eax, dword ptr [rsp + 0x18]
0x115545:      test    eax, eax
0x115547:      jne     0x115738
0x115760:      mov     dword ptr [rip + 0x16346], 0
0x11576a:      xor     eax, eax
0x11576c:      call    0x10c8c0

```

```

0x115636:    cmp     byte ptr [rbp + 0x6d], 0
0x11563a:    jne     0x115890
0x11554d:    mov     rdi, qword ptr [rip + 0x14f54]
0x115554:    call    0x10db60
0x115738:    mov     ecx, 2
0x11573d:    lea     rdx, [rip + 0xbcd9]
0x115744:    mov     rsi, r12
0x115747:    xor     eax, eax
0x115749:    lea     rdi, [rip + 0xc528]
0x115750:    call    0x10f760
0x115771:    mov     edx, 4
0x115776:    lea     rsi, [rip + 0xad5]
0x11577d:    lea     rdi, [rip + 0xba75]
0x115784:    mov     qword ptr [rsp + 8], rax
0x115789:    call    0x108070
0x115640:    xor     eax, eax
0x115642:    mov     rdi, rbp
0x115645:    call    0x110240
0x115890:    xor     eax, eax
0x115892:    mov     rdi, rbp
0x115895:    call    0x10ec50
0x115559:    xor     eax, eax
0x11555b:    call    0x10a820
0x115755:    movsx   r13d, al
0x115759:    jmp     0x11554d
0x11578e:    mov     rdi, qword ptr [rsp + 8]
0x115793:    mov     r14d, eax
0x115796:    call    0x10c970
0x11564a:    cmp     byte ptr [rbp + 0x6d], 0
0x11564e:    jne     0x115880
0x11589a:    xor     eax, eax
0x11589c:    mov     rdi, rbp
0x11589f:    call    0x110240
0x115560:    cmp     r13d, 0x61
0x115564:    je      0x115728
0x11579b:    xor     eax, eax
0x11579d:    and     r14d, 0xffffffff
0x1157a1:    call    0x10d870
0x115654:    mov     rax, qword ptr [rsp + 0x10]
0x115659:    mov     rdi, qword ptr [rbp + 0x18]
0x11565d:    mov     r12, qword ptr [rax + 0x18]
0x115661:    mov     rsi, r12
0x115664:    call    0x104b30
0x115880:    mov     rdi, rbp
0x115883:    xor     eax, eax
0x115885:    call    0x110d70
0x1158a4:    cmp     byte ptr [rbp + 0x6d], 0
0x1158a8:    je      0x115654

```

```

0x115728:    mov     dword ptr [rsp + 0x18], 0
0x115730:    jmp     0x11557a
0x11556a:    cmp     r13d, 0x73
0x11556e:    je      0x1154b8
0x1157a6:    xor     eax, eax
0x1157a8:    call    0x10d740
0x115669:    test    eax, eax
0x11566b:    jne     0x115431
0x11588a:    jmp     0x115654
0x1158ae:    jmp     0x115880
0x11557a:    mov     rax, qword ptr [rbp + 0x38]
0x11557e:    cmp     byte ptr [rax + rbx + 0x48], 0
0x115583:    jne     0x1156e8
0x1154b8:    mov     rdi, rbp
0x1154bb:    call    0x1100e0
0x115574:    cmp     r13d, 0x79
0x115578:    jne     0x1155c6
0x1157ad:    xor     eax, eax
0x1157af:    call    0x10a820
0x115671:    mov     rdi, qword ptr [rsp + 0x10]
0x115676:    mov     edx, 1
0x11567b:    mov     rsi, r12
0x11567e:    call    0x10f880
0x115589:    mov     rdi, qword ptr [rip + 0x14f10]
0x115590:    call    0x10db60
0x1156e8:    mov     rdi, r12
0x1156eb:    call    0x104870
0x1157b4:    cmp     r14b, 0x4e
0x1157b8:    je      0x115541
0x115683:    test    eax, eax
0x115685:    je      0x1158ed
0x115595:    xor     eax, eax
0x115597:    call    0x10a820
0x1156f0:    test    eax, eax
0x1156f2:    je      0x1155ac
0x1157be:    jmp     0x1155c6
0x1158ed:    mov     rdi, qword ptr [rsp + 0x10]
0x1158f2:    call    0x114d00
0x11568b:    mov     rbx, qword ptr [rsp + 0x10]
0x115690:    xor     eax, eax
0x115692:    mov     rdi, rbx
0x115695:    call    0x110ec0
0x11559c:    mov     rdi, r12
0x11559f:    call    0x104990
0x1155ac:    mov     rax, qword ptr [rbp + 0x38]
0x1155b0:    mov     rdi, rbp
0x1155b3:    mov     byte ptr [rax + rbx + 0x49], 0
0x1155b8:    call    0x1100e0

```



```

0x1156f8:    xor     eax, eax
0x1156fa:    mov     ecx, 0xe
0x1156ff:    mov     rsi, r12
0x115702:    lea     rdx, [rip + 0xae29]
0x115709:    lea     rdi, [rip + 0xc5e8]
0x115710:    call    0x10f760
0x1158f7:    jmp     0x11568b
0x11569a:    xor     eax, eax
0x11569c:    mov     rdi, rbx
0x11569f:    call    0x110160
0x1155a4:    test    eax, eax
0x1155a6:    jne     0x1157c8
0x1155bd:    cmp     eax, -1
0x1155c0:    jne     0x1154c9
0x115715:    cmp     al, 0x79
0x115717:    je      0x1154b8
0x1156a4:    cmp     byte ptr [rbx + 0x6d], 0
0x1156a8:    je      0x1156b4
0x1157c8:    xor     eax, eax
0x1157ca:    mov     ecx, 0x12
0x1157cf:    mov     rsi, r12
0x1157d2:    lea     rdx, [rip + 0xad59]
0x1157d9:    lea     rdi, [rip + 0xc4c0]
0x1157e0:    call    0x10f760
0x11571d:    jmp     0x1155c6
0x1156b4:    mov     rbx, qword ptr [rsp + 0x10]
0x1156b9:    xor     eax, eax
0x1156bb:    mov     rdi, rbx
0x1156be:    call    0x110240
0x1156aa:    mov     rdi, rbx
0x1156ad:    xor     eax, eax
0x1156af:    call    0x10ec50
0x1157e5:    cmp     al, 0x79
0x1157e7:    jne     0x1156f8
0x1156c3:    cmp     byte ptr [rbx + 0x6d], 0
0x1156c7:    je      0x115431
0x1157ed:    mov     rdi, r12
0x1157f0:    call    0x1049b0
0x1156cd:    add     rsp, 0x28
0x1156d1:    mov     rdi, rbx
0x1156d4:    xor     eax, eax
0x1156d6:    pop     rbx
0x1156d7:    pop     rbp
0x1156d8:    pop     r12
0x1156da:    pop     r13
0x1156dc:    pop     r14
0x1156de:    pop     r15
0x1156e0:    jmp     0x110d70

```

```

0x1157f5:    lea     rdi, [rax + 0x21]
0x1157f9:    call   0x11d670
0x1157fe:    mov     rdx, -1
0x115805:    mov     r8, r12
0x115808:    lea     rcx, [rip + 0xbb38]
0x11580f:    mov     r14, rax
0x115812:    mov     rdi, rax
0x115815:    mov     esi, 1
0x11581a:    xor     eax, eax
0x11581c:    call   0x104f10
0x115821:    mov     esi, 1
0x115826:    mov     rdi, r14
0x115829:    call   0x1193f0
0x11582e:    mov     rdi, r14
0x115831:    mov     dword ptr [rsp + 8], eax
0x115835:    call   0x118d10
0x11583a:    mov     rdi, qword ptr [rbp + 0x18]
0x11583e:    call   0x10c4b0
0x115843:    mov     eax, dword ptr [rsp + 8]
0x115847:    mov     edx, eax
0x115849:    and     edx, 0x7f
0x11584c:    add     edx, 1
0x11584f:    cmp     dl, 1
0x115852:    jg      0x1158b0
0x115854:    test    al, 0x7f
0x115856:    jne     0x1156f8
0x1158b0:    xor     eax, eax
0x1158b2:    mov     ecx, 0xe
0x1158b7:    mov     rsi, r12
0x1158ba:    lea     rdx, [rip + 0xac71]
0x1158c1:    lea     rdi, [rip + 0xc408]
0x1158c8:    call   0x10f760
0x11585c:    test    ah, 0xff
0x11585f:    je      0x1155ac
0x1158cd:    cmp     al, 0x79
0x1158cf:    je      0x1154b8
0x115865:    lea     rdi, [rip + 0xb37e]
0x11586c:    call   0x1195e0
0x1158d5:    jmp     0x1155c6
0x115871:    jmp     0x1156f8
0x1156e5:    nop     dword ptr [rax]
0x115722:    nop     word ptr [rax + rax]
0x115735:    nop     dword ptr [rax]
0x11575e:    nop
0x1157c3:    nop     dword ptr [rax + rax]
0x115876:    nop     word ptr cs:[rax + rax]
0x11588f:    nop
0x1158da:    nop     word ptr [rax + rax]

```

```

0x115909:      nop      dword ptr [rax]
0x115910:      endbr64
0x115914:      push     r13
0x115916:      xor      edx, edx
0x115918:      push     r12
0x11591a:      push     rbp
0x11591b:      mov      rbp, rsi
0x11591e:      push     rbx
0x11591f:      mov      rbx, rdi
0x115922:      lea      rdi, [rip + 0xbaf9]
0x115929:      sub      rsp, 0x18
0x11592d:      mov      rcx, qword ptr [rip + 0x15cd4]
0x115934:      mov      rax, qword ptr fs:[0x28]
0x11593d:      mov      qword ptr [rsp + 8], rax
0x115942:      xor      eax, eax
0x115944:      mov      rsi, rsp
0x115947:      mov      qword ptr [rsp], 0
0x11594f:      call     0x1083e0
0x115954:      test     rax, rax
0x115957:      je       0x115a0b
0x115a0b:      mov      rax, qword ptr [rsp + 8]
0x115a10:      sub      rax, qword ptr fs:[0x28]
0x115a19:      jne      0x115ba4
0x11595d:      mov      rdi, qword ptr [rsp]
0x115961:      cmp      byte ptr [rdi], 0
0x115964:      je       0x115a06
0x115a1f:      add      rsp, 0x18
0x115a23:      pop      rbx
0x115a24:      pop      rbp
0x115a25:      pop      r12
0x115a27:      pop      r13
0x115a29:      ret
0x115ba4:      call     0x1049d0
0x115a06:      call     0x118d10
0x11596a:      xor      eax, eax
0x11596c:      call     0x119940
0x115971:      mov      rdi, qword ptr [rsp]
0x115975:      mov      r13, rax
0x115978:      call     0x118d10
0x11597d:      mov      rdi, r13
0x115980:      mov      qword ptr [rsp], r13
0x115984:      call     0x1049b0
0x115989:      mov      rdi, r13
0x11598c:      lea      rsi, [rax + 2]
0x115990:      mov      r12, rax
0x115993:      call     0x104ce0
0x115998:      mov      esi, 0x41ff
0x11599d:      mov      qword ptr [rsp], rax

```

```

0x1159a1:    mov     byte ptr [rax + r12], 0x2f
0x1159a6:    mov     rax, qword ptr [rsp]
0x1159aa:    mov     byte ptr [rax + r12 + 1], 0
0x1159b0:    mov     rdi, qword ptr [rsp]
0x1159b4:    xor     eax, eax
0x1159b6:    call    0x1124d0
0x1159bb:    cmp     eax, -1
0x1159be:    je      0x115a90
0x115a90:    mov     rsi, qword ptr [rsp]
0x115a94:    mov     ecx, 0x1c
0x115a99:    xor     edx, edx
0x115a9b:    xor     eax, eax
0x115a9d:    lea     rdi, [rip + 0xb993]
0x115aa4:    call    0x10f760
0x1159c4:    mov     rsi, qword ptr [rbx + 0x18]
0x1159c8:    xor     eax, eax
0x1159ca:    mov     edx, 1
0x1159cf:    mov     rdi, rbx
0x1159d2:    call    0x10f880
0x115aa9:    mov     rdi, qword ptr [rsp]
0x115aad:    call    0x118d10
0x1159d7:    test    eax, eax
0x1159d9:    jne     0x115a30
0x115ab2:    jmp     0x115a0b
0x1159db:    mov     rdi, rbx
0x1159de:    call    0x114d00
0x115a30:    mov     rsi, qword ptr [rsp]
0x115a34:    mov     rdi, rbx
0x115a37:    xor     eax, eax
0x115a39:    call    0x1122d0
0x1159e3:    mov     rsi, qword ptr [rbp + 0x18]
0x1159e7:    mov     rdi, qword ptr [rbx + 0x18]
0x1159eb:    call    0x104b30
0x115a3e:    mov     edx, dword ptr [rbx + 0x60]
0x115a41:    mov     esi, dword ptr [rbx + 0x68]
0x115a44:    mov     edi, 0
0x115a49:    mov     dword ptr [rbx + 0x28], eax
0x115a4c:    sub     edx, 2
0x115a4f:    mov     ecx, edx
0x115a51:    shr     ecx, 0x1f
0x115a54:    add     ecx, edx
0x115a56:    sar     ecx, 1
0x115a58:    sub     eax, ecx
0x115a5a:    lea     ecx, [rdx + rax]
0x115a5d:    cmovs   eax, edi
0x115a60:    cmp     esi, ecx
0x115a62:    jg      0x115a70
0x1159f0:    test    eax, eax

```

```

0x1159f2:    je      0x115af0
0x115a64:    mov     eax, esi
0x115a66:    sub     eax, edx
0x115a68:    mov     edx, 0
0x115a6d:    cmovs   eax, edx
0x115a70:    cmp     byte ptr [rbx + 0x6d], 0
0x115a74:    mov     dword ptr [rbx + 0x2c], eax
0x115a77:    jne     0x115ac0
0x115a70:    cmp     byte ptr [rbx + 0x6d], 0
0x115a74:    mov     dword ptr [rbx + 0x2c], eax
0x115a77:    jne     0x115ac0
0x115af0:    movsxd  rax, dword ptr [rbp + 0x28]
0x115af4:    lea     rax, [rax + rax*4]
0x115af8:    shl     rax, 4
0x115afc:    add     rax, qword ptr [rbp + 0x38]
0x115b00:    mov     rdi, qword ptr [rax]
0x115b03:    call    0x11d7a0
0x1159f8:    mov     rdi, rbp
0x1159fb:    xor     eax, eax
0x1159fd:    call    0x110240
0x115a79:    mov     rdi, rbx
0x115a7c:    xor     eax, eax
0x115a7e:    call    0x110240
0x115ac0:    xor     eax, eax
0x115ac2:    mov     rdi, rbx
0x115ac5:    call    0x110d70
0x115b08:    mov     rsi, qword ptr [rbp + 0x18]
0x115b0c:    mov     edx, 1
0x115b11:    mov     rdi, rbp
0x115b14:    mov     r12, rax
0x115b17:    xor     eax, eax
0x115b19:    call    0x10f880
0x115a02:    mov     rdi, qword ptr [rsp]
0x115a06:    call    0x118d10
0x115a83:    jmp     0x1159e3
0x115aca:    cmp     byte ptr [rbx + 0x6d], 0
0x115ace:    je      0x115a79
0x115b1e:    test    eax, eax
0x115b20:    jne     0x115b40
0x115ad0:    mov     rdi, rbx
0x115ad3:    xor     eax, eax
0x115ad5:    call    0x10ec50
0x115b22:    mov     rdi, rbp
0x115b25:    call    0x114d00
0x115b40:    mov     rsi, r12
0x115b43:    mov     rdi, rbp
0x115b46:    xor     eax, eax
0x115b48:    call    0x1122d0

```

```

0x115ada:    mov     rdi, rbx
0x115add:    xor     eax, eax
0x115adf:    call   0x110240
0x115b2a:    mov     rdi, r12
0x115b2d:    call   0x118d10
0x115b4d:    mov     edx, dword ptr [rbp + 0x60]
0x115b50:    mov     esi, dword ptr [rbp + 0x68]
0x115b53:    mov     edi, 0
0x115b58:    mov     dword ptr [rbp + 0x28], eax
0x115b5b:    sub     edx, 2
0x115b5e:    mov     ecx, edx
0x115b60:    shr     ecx, 0x1f
0x115b63:    add     ecx, edx
0x115b65:    sar     ecx, 1
0x115b67:    sub     eax, ecx
0x115b69:    lea     ecx, [rdx + rax]
0x115b6c:    cmovs   eax, edi
0x115b6f:    cmp     esi, ecx
0x115b71:    jg      0x115b7f
0x115ae4:    jmp     0x1159e3
0x115b32:    jmp     0x1159f8
0x115b73:    mov     eax, esi
0x115b75:    sub     eax, edx
0x115b77:    mov     edx, 0
0x115b7c:    cmovs   eax, edx
0x115b7f:    cmp     byte ptr [rbp + 0x6d], 0
0x115b83:    mov     dword ptr [rbp + 0x2c], eax
0x115b86:    je      0x115b2a
0x115b7f:    cmp     byte ptr [rbp + 0x6d], 0
0x115b83:    mov     dword ptr [rbp + 0x2c], eax
0x115b86:    je      0x115b2a
0x115b88:    xor     eax, eax
0x115b8a:    mov     rdi, rbp
0x115b8d:    call   0x110d70
0x115b92:    cmp     byte ptr [rbp + 0x6d], 0
0x115b96:    je      0x115b2a
0x115b98:    mov     rdi, rbp
0x115b9b:    xor     eax, eax
0x115b9d:    call   0x10ec50
0x115ba2:    jmp     0x115b2a
0x115a2a:    nop     word ptr [rax + rax]
0x115a88:    nop     dword ptr [rax + rax]
0x115ab7:    nop     word ptr [rax + rax]
0x115ae9:    nop     dword ptr [rax]
0x115b37:    nop     word ptr [rax + rax]
0x115ba9:    nop     dword ptr [rax]
0x115bb0:    endbr64
0x115bb4:    push    r15

```

```

0x115bb6:    mov     r15, rdi
0x115bb9:    push    r14
0x115bbb:    push    r13
0x115bbd:    push    r12
0x115bbf:    mov     r12, rsi
0x115bc2:    push    rbp
0x115bc3:    push    rbx
0x115bc4:    sub     rsp, 0x48
0x115bc8:    mov     edx, dword ptr [rdi + 0x70]
0x115bcb:    mov     rax, qword ptr fs:[0x28]
0x115bd4:    mov     qword ptr [rsp + 0x38], rax
0x115bd9:    xor     eax, eax
0x115bdb:    mov     qword ptr [rsp + 0x28], 0
0x115be4:    mov     qword ptr [rsp + 0x30], 0
0x115bed:    mov     dword ptr [rdi + 0x90], 1
0x115bf7:    test    edx, edx
0x115bf9:    jne     0x115ec8
0x115bff:    movsxd  rax, dword ptr [rdi + 0x28]
0x115c03:    mov     rdx, rax
0x115c06:    lea     rax, [rax + rax*4]
0x115c0a:    shl     rax, 4
0x115c0e:    add     rax, qword ptr [rdi + 0x38]
0x115c12:    mov     rbp, qword ptr [rax]
0x115c15:    test    edx, edx
0x115c17:    jne     0x115c30
0x115ec8:    mov     rdx, qword ptr [rsi + 0x18]
0x115ecc:    mov     rcx, qword ptr [rip + 0x1573d]
0x115ed3:    lea     rsi, [rsp + 0x28]
0x115ed8:    lea     rdi, [rip + 0xb591]
0x115edf:    call    0x1083e0
0x115c19:    mov     rax, qword ptr [rdi + 0x18]
0x115c1d:    cmp     byte ptr [rax], 0x2f
0x115c20:    jne     0x115ea0
0x115c30:    mov     rdi, rbp
0x115c33:    call    0x1049b0
0x115ee4:    test    rax, rax
0x115ee7:    je      0x115ea0
0x115c26:    cmp     byte ptr [rax + 1], 0
0x115c2a:    jne     0x115ea0
0x115ea0:    mov     rax, qword ptr [rsp + 0x38]
0x115ea5:    sub     rax, qword ptr fs:[0x28]
0x115eae:    jne     0x1161bd
0x115c38:    lea     rdi, [rax + 0x11]
0x115c3c:    call    0x11d670
0x115ee9:    mov     rdi, qword ptr [r15 + 0x18]
0x115eed:    mov     rsi, qword ptr [rsp + 0x28]
0x115ef2:    xor     eax, eax
0x115ef4:    call    0x111190

```

```

0x115eb4:    add     rsp, 0x48
0x115eb8:    pop     rbx
0x115eb9:    pop     rbp
0x115eba:    pop     r12
0x115ebc:    pop     r13
0x115ebe:    pop     r14
0x115ec0:    pop     r15
0x115ec2:    ret
0x1161bd:    call    0x1049d0
0x115c41:    mov     edx, 0xe
0x115c46:    mov     rsi, rbp
0x115c49:    lea     rdi, [rip + 0x15990]
0x115c50:    mov     r13, rax
0x115c53:    call    0x104880
0x115ef9:    mov     r13d, eax
0x115efc:    test    eax, eax
0x115efe:    jne     0x1160b8
0x115c58:    lea     rcx, [rip + 0xb7ef]
0x115c5f:    mov     esi, 1
0x115c64:    mov     rdi, r13
0x115c67:    mov     r8, rax
0x115c6a:    mov     rdx, -1
0x115c71:    xor     eax, eax
0x115c73:    call    0x104f10
0x115f04:    mov     rbp, qword ptr [rsp + 0x28]
0x115f09:    mov     rdi, rbp
0x115f0c:    call    0x1049b0
0x1160b8:    mov     edx, 0xc
0x1160bd:    xor     esi, esi
0x1160bf:    lea     rdi, [rip + 0x1423a]
0x1160c6:    call    0x108070
0x115c78:    mov     rdi, rbp
0x115c7b:    call    0x1049b0
0x115f11:    mov     rdi, rbp
0x115f14:    mov     rbx, rax
0x115f17:    lea     rax, [rax + 1]
0x115f1b:    lea     r14, [rbx + 2]
0x115f1f:    mov     qword ptr [rsp + 8], rax
0x115f24:    mov     rsi, r14
0x115f27:    call    0x11d690
0x1160cb:    jmp     0x115ea0
0x115c80:    mov     rdi, qword ptr [r12 + 0x18]
0x115c85:    mov     rbx, rax
0x115c88:    call    0x1049b0
0x115f2c:    mov     qword ptr [rsp + 0x28], rax
0x115f31:    mov     byte ptr [rax + rbx], 0x2f
0x115f35:    mov     rax, qword ptr [rsp + 0x28]
0x115f3a:    mov     byte ptr [rax + rbx + 1], 0

```



```

0x115f3f:    mov     edx, dword ptr [r15 + 0x68]
0x115f43:    test    edx, edx
0x115f45:    jle     0x115f6a
0x115c8d:    lea     rdi, [rbx + rax + 2]
0x115c92:    call    0x11d670
0x115f6a:    mov     eax, dword ptr [r15 + 0x70]
0x115f6e:    mov     dword ptr [r15 + 0x50], 0xffffffff
0x115f76:    mov     dword ptr [r15 + 0x54], eax
0x115f7a:    nop     word ptr [rax + rax]
0x115f80:    mov     rdi, r15
0x115f83:    call    0x1100e0
0x115f47:    mov     rax, qword ptr [r15 + 0x38]
0x115f4b:    add     rax, 0x49
0x115f4f:    jmp     0x115f65
0x115c97:    mov     r8, qword ptr [r12 + 0x18]
0x115c9c:    mov     r9, rbp
0x115c9f:    lea     rcx, [rip + 0xaa5b]
0x115ca6:    mov     rdi, rax
0x115ca9:    mov     r14, rax
0x115cac:    mov     esi, 1
0x115cb1:    xor     eax, eax
0x115cb3:    mov     rdx, -1
0x115cba:    call    0x104f10
0x115f88:    cmp     eax, -1
0x115f8b:    je      0x116060
0x115f65:    cmp     byte ptr [rax], 0
0x115f68:    je      0x115f58
0x115cbf:    mov     rdi, r13
0x115cc2:    lea     rsi, [rsp + 0x30]
0x115cc7:    mov     rdx, r14
0x115cca:    mov     rcx, qword ptr [rip + 0x1593f]
0x115cd1:    call    0x1083e0
0x116060:    mov     rdi, qword ptr [rsp + 0x28]
0x116065:    test    rdi, rdi
0x116068:    je      0x11606f
0x115f91:    cdq     rax
0x115f93:    lea     rbx, [rax + rax*4]
0x115f97:    mov     rax, qword ptr [r15 + 0x38]
0x115f9b:    shl     rbx, 4
0x115f9f:    mov     rbp, qword ptr [rax + rbx]
0x115fa3:    xor     eax, eax
0x115fa5:    call    0x118ab0
0x115f58:    add     r13d, 1
0x115f5c:    add     rax, 0x50
0x115f60:    cmp     r13d, edx
0x115f63:    je      0x115f6a
0x115cd6:    mov     rdi, r13
0x115cd9:    test    rax, rax

```

```

0x115cdc:    je      0x116173
0x11606f:    cmp     dword ptr [r15 + 0x68], r13d
0x116073:    je      0x116087
0x11606a:    call    0x118d10
0x115faa:    mov     rdi, rbp
0x115fad:    call    0x1049b0
0x116173:    call    0x118d10
0x115ce2:    call    0x118d10
0x116087:    xor     eax, eax
0x116089:    call    0x10e4e0
0x116075:    mov     dword ptr [r15 + 0x28], r13d
0x116079:    mov     rdi, r15
0x11607c:    xor     eax, eax
0x11607e:    call    0x10f110
0x115fb2:    mov     rdi, qword ptr [rsp + 0x28]
0x115fb7:    lea     rsi, [r14 + rax]
0x115fbb:    call    0x11d690
0x116178:    jmp     0x115ea0
0x115ce7:    movsxd  rax, dword ptr [r15 + 0x28]
0x115ceb:    lea     rax, [rax + rax*4]
0x115cef:    shl     rax, 4
0x115cf3:    add     rax, qword ptr [r15 + 0x38]
0x115cf7:    mov     eax, dword ptr [rax + 0x10]
0x115cfa:    and     eax, 0xf000
0x115cff:    cmp     eax, 0x4000
0x115d04:    je      0x1160a0
0x11608e:    xor     eax, eax
0x116090:    call    0x10a820
0x116083:    mov     dword ptr [r15 + 0x2c], eax
0x116087:    xor     eax, eax
0x116089:    call    0x10e4e0
0x115fc0:    mov     rcx, qword ptr [rsp + 8]
0x115fc5:    mov     rsi, rbp
0x115fc8:    mov     qword ptr [rsp + 0x28], rax
0x115fcd:    lea     rdi, [rax + rcx]
0x115fd1:    call    0x1048b0
0x1160a0:    mov     rdi, qword ptr [rip + 0x14419]
0x1160a7:    call    0x10db60
0x115d0a:    mov     rdi, qword ptr [rip + 0x1479f]
0x115d11:    call    0x10db60
0x116095:    jmp     0x115de2
0x115fd6:    mov     eax, dword ptr [rip + 0x15ad4]
0x115fdc:    test    eax, eax
0x115fde:    jne     0x1160e0
0x1160ac:    jmp     0x115d16
0x115d16:    xor     eax, eax
0x115d18:    call    0x10a820
0x115de2:    mov     rsi, qword ptr [r12 + 0x18]

```

```

0x115de7:    xor     eax, eax
0x115de9:    mov     edx, 1
0x115dee:    mov     rdi, r12
0x115df1:    call    0x10f880
0x115fe4:    mov     rdi, qword ptr [rip + 0x144cd]
0x115feb:    call    0x10db60
0x1160e0:    mov     dword ptr [rip + 0x159c6], 0
0x1160ea:    xor     eax, eax
0x1160ec:    call    0x10c8c0
0x115d1d:    mov     rdi, qword ptr [rsp + 0x30]
0x115d22:    xor     eax, eax
0x115d24:    call    0x119940
0x115df6:    test    eax, eax
0x115df8:    je      0x116190
0x115ff0:    xor     eax, eax
0x115ff2:    call    0x10a820
0x1160f1:    mov     edx, 4
0x1160f6:    lea     rsi, [rip + 0xa435]
0x1160fd:    lea     rdi, [rip + 0xb0f5]
0x116104:    mov     qword ptr [rsp + 0x18], rax
0x116109:    call    0x108070
0x115d29:    mov     rdi, qword ptr [rsp + 0x30]
0x115d2e:    mov     r13, rax
0x115d31:    call    0x118d10
0x116190:    mov     rdi, r12
0x116193:    call    0x114d00
0x115dfe:    xor     eax, eax
0x115e00:    mov     rdi, r12
0x115e03:    call    0x110ec0
0x115ff7:    mov     rax, qword ptr [r15 + 0x38]
0x115ffb:    mov     rsi, rbp
0x115ffe:    mov     rdi, r15
0x116001:    mov     rdx, qword ptr [rsp + 0x28]
0x116006:    mov     ecx, dword ptr [rax + rbx + 0x10]
0x11600a:    xor     eax, eax
0x11600c:    call    0x111d10
0x11610e:    mov     rdi, qword ptr [rsp + 0x18]
0x116113:    mov     byte ptr [rsp + 0x17], al
0x116117:    call    0x10c970
0x115d36:    mov     rdi, rbp
0x115d39:    xor     eax, eax
0x115d3b:    mov     rsi, r13
0x115d3e:    mov     qword ptr [rsp + 0x30], r13
0x115d43:    call    0x111190
0x116198:    jmp     0x115dfe
0x115e08:    xor     eax, eax
0x115e0a:    mov     rdi, r12
0x115e0d:    call    0x110160

```

```

0x116011:    cmp     eax, 1
0x116014:    je      0x1160d0
0x11611c:    xor     eax, eax
0x11611e:    call    0x10d870
0x115d48:    mov     rdi, r14
0x115d4b:    mov     ebx, eax
0x115d4d:    call    0x118d10
0x115e12:    cmp     byte ptr [r12 + 0x6d], 0
0x115e18:    je      0x115e24
0x1160d0:    mov     rax, qword ptr [r15 + 0x38]
0x1160d4:    mov     byte ptr [rax + rbx + 0x49], 0
0x1160d9:    jmp     0x115f80
0x11601a:    cmp     eax, 2
0x11601d:    je      0x116060
0x116123:    xor     eax, eax
0x116125:    call    0x10d740
0x115d52:    test    ebx, ebx
0x115d54:    jne     0x116150
0x115e24:    xor     eax, eax
0x115e26:    mov     rdi, r12
0x115e29:    call    0x110240
0x115e1a:    mov     rdi, r12
0x115e1d:    xor     eax, eax
0x115e1f:    call    0x10ec50
0x115f80:    mov     rdi, r15
0x115f83:    call    0x1100e0
0x11601f:    cmp     eax, 3
0x116022:    je      0x115f80
0x11612a:    xor     eax, eax
0x11612c:    call    0x10a820
0x115d5a:    movsxd  rax, dword ptr [r15 + 0x28]
0x115d5e:    mov     rdx, qword ptr [rsp + 0x30]
0x115d63:    mov     rsi, rbp
0x115d66:    mov     rdi, r15
0x115d69:    lea     rax, [rax + rax*4]
0x115d6d:    shl     rax, 4
0x115d71:    add     rax, qword ptr [r15 + 0x38]
0x115d75:    mov     ecx, dword ptr [rax + 0x10]
0x115d78:    xor     eax, eax
0x115d7a:    call    0x111d10
0x116150:    mov     rdx, qword ptr [rsp + 0x30]
0x116155:    lea     rdi, [rip + 0xbaf4]
0x11615c:    xor     ecx, ecx
0x11615e:    xor     eax, eax
0x116160:    mov     r8d, 0x1e
0x116166:    mov     rsi, rbp
0x116169:    call    0x10f7e0
0x115e2e:    cmp     byte ptr [r12 + 0x6d], 0

```

```

0x115e34:    je      0x115e40
0x116028:    sub     eax, 1
0x11602b:    lea     rdx, [rip + 0x1438e]
0x116032:    xor     ecx, ecx
0x116034:    mov     rsi, rbp
0x116037:    cdqe
0x116039:    mov     r8d, 0xe
0x11603f:    lea     rdi, [rip + 0xb415]
0x116046:    mov     rdx, qword ptr [rdx + rax*8]
0x11604a:    xor     eax, eax
0x11604c:    call    0x10f7e0
0x116131:    movzx   eax, byte ptr [rsp + 0x17]
0x116136:    and     eax, 0xffffffff
0x116139:    cmp     al, 0x4e
0x11613b:    je      0x115fe4
0x115d7f:    cmp     eax, 1
0x115d82:    je      0x115db6
0x11616e:    mov     rdi, qword ptr [rsp + 0x30]
0x116173:    call    0x118d10
0x115e40:    mov     rsi, qword ptr [r15 + 0x18]
0x115e44:    xor     eax, eax
0x115e46:    mov     edx, 1
0x115e4b:    mov     rdi, r15
0x115e4e:    call    0x10f880
0x115e36:    mov     rdi, r12
0x115e39:    xor     eax, eax
0x115e3b:    call    0x110d70
0x116051:    test    al, al
0x116053:    jne     0x115f80
0x116141:    jmp     0x116060
0x115db6:    mov     rdi, qword ptr [rsp + 0x30]
0x115dbb:    call    0x118d10
0x115d84:    cmp     eax, 2
0x115d87:    je      0x1161a0
0x115e53:    test    eax, eax
0x115e55:    je      0x116180
0x116059:    nop     dword ptr [rax]
0x115dc0:    xor     eax, eax
0x115dc2:    call    0x10e4e0
0x1161a0:    mov     rdi, qword ptr [rsp + 0x30]
0x1161a5:    call    0x118d10
0x115d8d:    sub     eax, 1
0x115d90:    lea     rdx, [rip + 0x14629]
0x115d97:    xor     ecx, ecx
0x115d99:    mov     rsi, rbp
0x115d9c:    cdqe
0x115d9e:    mov     r8d, 0xe
0x115da4:    lea     rdi, [rip + 0xb6b0]

```

```

0x115dab:    mov     rdx, qword ptr [rdx + rax*8]
0x115daf:    xor     eax, eax
0x115db1:    call    0x10f7e0
0x116180:    mov     rdi, r15
0x116183:    call    0x114d00
0x115e5b:    xor     eax, eax
0x115e5d:    mov     rdi, r15
0x115e60:    call    0x110ec0
0x115dc7:    mov     rdi, r15
0x115dca:    xor     eax, eax
0x115dcc:    call    0x110240
0x1161aa:    xor     eax, eax
0x1161ac:    call    0x10e4e0
0x116188:    jmp     0x115e5b
0x115e65:    xor     eax, eax
0x115e67:    mov     rdi, r15
0x115e6a:    call    0x110160
0x115dd1:    xor     eax, eax
0x115dd3:    mov     rdi, r12
0x115dd6:    call    0x110240
0x1161b1:    xor     eax, eax
0x1161b3:    call    0x10a820
0x115e6f:    cmp     byte ptr [r15 + 0x6d], 0
0x115e74:    je      0x115e80
0x115ddb:    xor     eax, eax
0x115ddd:    call    0x10a820
0x1161b8:    jmp     0x115ea0
0x115e80:    xor     eax, eax
0x115e82:    mov     rdi, r15
0x115e85:    call    0x110240
0x115e76:    mov     rdi, r15
0x115e79:    xor     eax, eax
0x115e7b:    call    0x10ec50
0x115e8a:    cmp     byte ptr [r15 + 0x6d], 0
0x115e8f:    je      0x115ea0
0x115e91:    mov     rdi, r15
0x115e94:    xor     eax, eax
0x115e96:    call    0x110d70
0x115e9b:    nop     dword ptr [rax + rax]
0x115ec3:    nop     dword ptr [rax + rax]
0x115f51:    nop     dword ptr [rax]
0x11609a:    nop     word ptr [rax + rax]
0x1160b1:    nop     dword ptr [rax]
0x1160de:    nop
0x116146:    nop     word ptr cs:[rax + rax]
0x11617d:    nop     dword ptr [rax]
0x11618d:    nop     dword ptr [rax]
0x11619d:    nop     dword ptr [rax]

```

```

0x1161c2:      nop      word ptr cs:[rax + rax]
0x1161cd:      nop      dword ptr [rax]
0x1161d0:      endbr64
0x1161d4:      push     r13
0x1161d6:      xor      r13d, r13d
0x1161d9:      push     r12
0x1161db:      lea      r12, [rip + 0xc203]
0x1161e2:      push     rbp
0x1161e3:      mov      rbp, rdi
0x1161e6:      push     rbx
0x1161e7:      mov      rbx, rsi
0x1161ea:      sub      rsp, 8
0x1161ee:      mov      rdx, qword ptr [rdi + 0x38]
0x1161f2:      test     rdx, rdx
0x1161f5:      je       0x11621a
0x11621a:      xor      edx, edx
0x11621c:      mov      rsi, qword ptr [rbp + 0x18]
0x116220:      cmp      rbx, -1
0x116224:      mov      rdi, rbp
0x116227:      sete     dl
0x11622a:      xor      eax, eax
0x11622c:      call     0x10f880
0x1161f7:      movsxd   rax, dword ptr [rdi + 0x28]
0x1161fb:      lea      rax, [rax + rax*4]
0x1161ff:      shl      rax, 4
0x116203:      mov      rdi, qword ptr [rdx + rax]
0x116207:      test     rdi, rdi
0x11620a:      je       0x11621a
0x116231:      test     eax, eax
0x116233:      je       0x116278
0x11620c:      call     0x11d7a0
0x116278:      mov      rdi, rbp
0x11627b:      xor      eax, eax
0x11627d:      call     0x114d00
0x116235:      cmp      rbx, -1
0x116239:      je       0x116288
0x116211:      mov      r13d, 1
0x116217:      mov      r12, rax
0x11621a:      xor      edx, edx
0x11621c:      mov      rsi, qword ptr [rbp + 0x18]
0x116220:      cmp      rbx, -1
0x116224:      mov      rdi, rbp
0x116227:      sete     dl
0x11622a:      xor      eax, eax
0x11622c:      call     0x10f880
0x116282:      jmp      0x11624f
0x116288:      mov      rsi, r12
0x11628b:      mov      rdi, rbp

```

```

0x11628e:    xor     eax, eax
0x116290:    call   0x1122d0
0x11623b:    mov     dword ptr [rbp + 0x28], 0
0x116242:    mov     rdi, rbp
0x116245:    xor     eax, eax
0x116247:    call   0x10f110
0x11624f:    test    r13d, r13d
0x116252:    jne     0x116260
0x116295:    mov     rdi, rbp
0x116298:    mov     dword ptr [rbp + 0x28], eax
0x11629b:    xor     eax, eax
0x11629d:    call   0x10f110
0x11624c:    mov     dword ptr [rbp + 0x2c], eax
0x11624f:    test    r13d, r13d
0x116252:    jne     0x116260
0x116254:    add     rsp, 8
0x116258:    pop     rbx
0x116259:    pop     rbp
0x11625a:    pop     r12
0x11625c:    pop     r13
0x11625e:    ret
0x116260:    add     rsp, 8
0x116264:    mov     rdi, r12
0x116267:    pop     rbx
0x116268:    pop     rbp
0x116269:    pop     r12
0x11626b:    pop     r13
0x11626d:    jmp     0x118d10
0x1162a2:    mov     dword ptr [rbp + 0x2c], eax
0x1162a5:    jmp     0x11624f
0x11625f:    nop
0x116272:    nop     word ptr [rax + rax]
0x116284:    nop     dword ptr [rax]
0x1162a7:    nop     word ptr [rax + rax]
0x1162b0:    endbr64
0x1162b4:    push    r15
0x1162b6:    push    r14
0x1162b8:    push    r13
0x1162ba:    push    r12
0x1162bc:    mov     r12, rdi
0x1162bf:    push    rbp
0x1162c0:    movsxd  rax, dword ptr [rdi + 0x28]
0x1162c4:    mov     rbp, rsi
0x1162c7:    lea     rax, [rax + rax*4]
0x1162cb:    shl     rax, 4
0x1162cf:    add     rax, qword ptr [rdi + 0x38]
0x1162d3:    movzx   edx, byte ptr [rax + 0x48]
0x1162d7:    mov     r14, qword ptr [rax]

```



```

0x1162da:    test    dl, dl
0x1162dc:    je      0x116300
0x116300:    cmp     byte ptr [r14], 0x2e
0x116304:    jne     0x1163c0
0x1162de:    cmp     dl, 1
0x1162e1:    jne     0x1162ed
0x11630a:    cmp     byte ptr [r14 + 1], 0x2e
0x11630f:    jne     0x1163c0
0x1163c0:    mov     rax, qword ptr [r12 + 0x78]
0x1163c5:    lea     rdi, [rax + 1]
0x1163c9:    call    0x11d670
0x1162e3:    cmp     byte ptr [rax + 0x4a], 0
0x1162e7:    jne     0x116420
0x1162ed:    xor     r13d, r13d
0x1162f0:    pop     rbp
0x1162f1:    mov     eax, r13d
0x1162f4:    pop     r12
0x1162f6:    pop     r13
0x1162f8:    pop     r14
0x1162fa:    pop     r15
0x1162fc:    ret
0x116315:    cmp     byte ptr [r14 + 2], 0
0x11631a:    jne     0x1163c0
0x1163ce:    mov     rsi, qword ptr [r12 + 0x18]
0x1163d3:    mov     rdi, rax
0x1163d6:    mov     r15, rax
0x1163d9:    call    0x1048b0
0x116420:    mov     rdi, r14
0x116423:    call    0x1049b0
0x116320:    mov     rax, qword ptr [rdi + 0x18]
0x116324:    cmp     byte ptr [rax], 0x2f
0x116327:    jne     0x116330
0x1163de:    mov     edx, 1
0x1163e3:    mov     rsi, r14
0x1163e6:    mov     rdi, r12
0x1163e9:    xor     eax, eax
0x1163eb:    call    0x10f880
0x116428:    lea     rdi, [rax + 0x21]
0x11642c:    call    0x11d670
0x116329:    cmp     byte ptr [rax + 1], 0
0x11632d:    je      0x1162ed
0x116330:    mov     rax, qword ptr [r12 + 0x78]
0x116335:    lea     rdi, [rax + 1]
0x116339:    call    0x11d670
0x1163f0:    mov     r13d, eax
0x1163f3:    test    eax, eax
0x1163f5:    jne     0x1164b0
0x116431:    mov     r8, r14

```

```

0x116434:      mov     esi, 1
0x116439:      lea     rcx, [rip + 0xb04b]
0x116440:      mov     rdx, -1
0x116447:      mov     r13, rax
0x11644a:      mov     rdi, rax
0x11644d:      xor     eax, eax
0x11644f:      call    0x104f10
0x11632f:      nop
0x11633e:      mov     rsi, qword ptr [r12 + 0x18]
0x116343:      mov     rdi, rax
0x116346:      mov     r15, rax
0x116349:      call    0x1048b0
0x1163fb:      mov     ecx, 0x1c
0x116400:      xor     edx, edx
0x116402:      mov     rsi, r14
0x116405:      xor     eax, eax
0x116407:      lea     rdi, [rip + 0xaf94]
0x11640e:      call    0x10f760
0x1164b0:      mov     dword ptr [r12 + 0x28], 0
0x1164b9:      mov     rdi, r12
0x1164bc:      xor     eax, eax
0x1164be:      call    0x10f110
0x116454:      xor     esi, esi
0x116456:      mov     rdi, r13
0x116459:      call    0x1193f0
0x11634e:      mov     edx, 1
0x116353:      mov     rsi, r14
0x116356:      mov     rdi, r12
0x116359:      xor     eax, eax
0x11635b:      call    0x10f880
0x116413:      jmp     0x1162f0
0x1164c3:      mov     dword ptr [r12 + 0x2c], eax
0x1164c8:      mov     rdi, r15
0x1164cb:      call    0x118d10
0x11645e:      mov     rdi, r13
0x116461:      mov     r13d, 1
0x116467:      mov     dword ptr [rip + 0x1446b], 1
0x116471:      call    0x118d10
0x116360:      mov     r13d, eax
0x116363:      test    eax, eax
0x116365:      jne     0x1165c0
0x1162f0:      pop     rbp
0x1162f1:      mov     eax, r13d
0x1162f4:      pop     r12
0x1162f6:      pop     r13
0x1162f8:      pop     r14
0x1162fa:      pop     r15
0x1162fc:      ret

```

```

0x1164d0:    xor     eax, eax
0x1164d2:    mov     rdi, r12
0x1164d5:    call    0x110ec0
0x116476:    xor     eax, eax
0x116478:    call    0x10a480
0x11636b:    xor     eax, eax
0x11636d:    mov     rdi, r12
0x116370:    call    0x114d00
0x1165c0:    mov     rdi, r15
0x1165c3:    mov     esi, 0x2f
0x1165c8:    call    0x104a20
0x1164da:    xor     eax, eax
0x1164dc:    mov     rdi, r12
0x1164df:    call    0x110160
0x11647d:    mov     rdi, qword ptr [rip + 0x14444]
0x116484:    call    0x10bb10
0x116375:    xor     eax, eax
0x116377:    mov     rdi, r12
0x11637a:    call    0x110ec0
0x1165cd:    mov     rdi, r12
0x1165d0:    lea     rsi, [rax + 1]
0x1165d4:    xor     eax, eax
0x1165d6:    call    0x1122d0
0x1164e4:    cmp     byte ptr [r12 + 0x6d], 0
0x1164ea:    je      0x1164f6
0x116489:    mov     rdi, r14
0x11648c:    mov     dword ptr [r12 + 0x14], 0xffffffff
0x116495:    mov     dword ptr [rbp + 0x14], 0xffffffff
0x11649c:    call    0x10d610
0x11637f:    xor     eax, eax
0x116381:    mov     rdi, r12
0x116384:    call    0x110160
0x1165db:    mov     rdi, r12
0x1165de:    mov     dword ptr [r12 + 0x28], eax
0x1165e3:    xor     eax, eax
0x1165e5:    call    0x10f110
0x1164f6:    xor     eax, eax
0x1164f8:    mov     rdi, r12
0x1164fb:    call    0x110240
0x1164ec:    mov     rdi, r12
0x1164ef:    xor     eax, eax
0x1164f1:    call    0x10ec50
0x1164a1:    mov     rdi, qword ptr [r12 + 0x18]
0x1164a6:    call    0x10c4b0
0x116389:    cmp     byte ptr [r12 + 0x6d], 0
0x11638f:    jne     0x1165b0
0x1165ea:    jmp     0x1164c3
0x116500:    cmp     byte ptr [r12 + 0x6d], 0

```

```

0x116506:      je      0x116512
0x1164ab:      jmp     0x1162f0
0x116395:      xor     eax, eax
0x116397:      mov     rdi, r12
0x11639a:      call    0x110240
0x1165b0:      mov     rdi, r12
0x1165b3:      xor     eax, eax
0x1165b5:      call    0x10ec50
0x116512:      mov     rsi, qword ptr [rbp + 0x18]
0x116516:      mov     rdi, qword ptr [r12 + 0x18]
0x11651b:      call    0x104b30
0x116508:      mov     rdi, r12
0x11650b:      xor     eax, eax
0x11650d:      call    0x110d70
0x11639f:      cmp     byte ptr [r12 + 0x6d], 0
0x1163a5:      je      0x1162ed
0x1165ba:      jmp     0x116395
0x116520:      mov     r13d, eax
0x116523:      test    eax, eax
0x116525:      jne     0x116588
0x1163ab:      mov     rdi, r12
0x1163ae:      xor     eax, eax
0x1163b0:      call    0x110d70
0x116527:      mov     r8d, 1
0x11652d:      mov     rdx, r12
0x116530:      mov     esi, 0x22
0x116535:      mov     rdi, rbp
0x116538:      mov     rcx, -1
0x11653f:      call    0x113e30
0x116588:      mov     rdi, rbp
0x11658b:      xor     eax, eax
0x11658d:      call    0x110240
0x1163b5:      jmp     0x1162f0
0x116544:      xor     eax, eax
0x116546:      mov     rdi, rbp
0x116549:      call    0x110ec0
0x116592:      jmp     0x1162ed
0x11654e:      xor     eax, eax
0x116550:      mov     rdi, rbp
0x116553:      call    0x110160
0x116558:      cmp     byte ptr [rbp + 0x6d], 0
0x11655c:      jne     0x1165a0
0x11655e:      xor     eax, eax
0x116560:      mov     rdi, rbp
0x116563:      call    0x110240
0x1165a0:      mov     rdi, rbp
0x1165a3:      xor     eax, eax
0x1165a5:      call    0x10ec50

```

```

0x116568:    cmp     byte ptr [rbp + 0x6d], 0
0x11656c:    je      0x1162ed
0x1165aa:    jmp     0x11655e
0x116572:    mov     rdi, rbp
0x116575:    xor     eax, eax
0x116577:    call    0x110d70
0x11657c:    jmp     0x1162f0
0x1162fd:    nop     dword ptr [rax]
0x1163ba:    nop     word ptr [rax + rax]
0x116418:    nop     dword ptr [rax + rax]
0x116581:    nop     dword ptr [rax]
0x116597:    nop     word ptr [rax + rax]
0x1165ac:    nop     dword ptr [rax]
0x1165bf:    nop
0x1165ef:    nop
0x1165f0:    endbr64
0x1165f4:    push    r13
0x1165f6:    push    r12
0x1165f8:    mov     r12, rdx
0x1165fb:    push    rbp
0x1165fc:    mov     rbp, rsi
0x1165ff:    push    rbx
0x116600:    mov     rbx, rdi
0x116603:    mov     rdi, rdx
0x116606:    sub     rsp, 8
0x11660a:    call    0x1049b0
0x11660f:    cmp     byte ptr [r12], 0x2f
0x116614:    je      0x1166f0
0x1166f0:    mov     qword ptr [rbx + 0x78], rax
0x1166f4:    mov     rdi, qword ptr [rbx + 0x18]
0x1166f8:    lea     rsi, [rax + 1]
0x1166fc:    call    0x11d690
0x11661a:    add     rax, qword ptr [rbx + 0x78]
0x11661e:    mov     rdi, qword ptr [rbx + 0x18]
0x116622:    lea     rdx, [rax + 1]
0x116626:    lea     rsi, [rax + 2]
0x11662a:    mov     qword ptr [rbx + 0x78], rdx
0x11662e:    call    0x11d690
0x116701:    mov     rsi, r12
0x116704:    mov     qword ptr [rbx + 0x18], rax
0x116708:    mov     rdi, rax
0x11670b:    call    0x1048b0
0x116633:    mov     qword ptr [rbx + 0x18], rax
0x116637:    mov     rdi, rax
0x11663a:    mov     r13, rax
0x11663d:    call    0x1049b0
0x116710:    jmp     0x116662
0x116642:    mov     edx, 0x2f

```

```

0x116647:    mov     rsi, r12
0x11664a:    mov     word ptr [r13 + rax], dx
0x116650:    mov     rdi, qword ptr [rbx + 0x18]
0x116654:    call    0x104df0
0x116662:    mov     dword ptr [rbx + 0x28], 0
0x116669:    mov     rdi, rbx
0x11666c:    xor     eax, eax
0x11666e:    mov     r8d, 1
0x116674:    call    0x10f110
0x116659:    mov     rdi, qword ptr [rbx + 0x18]
0x11665d:    call    0x118040
0x116679:    xor     ecx, ecx
0x11667b:    xor     edx, edx
0x11667d:    mov     esi, 0x22
0x116682:    mov     dword ptr [rbx + 0x2c], eax
0x116685:    mov     rdi, rbx
0x116688:    call    0x113e30
0x11668d:    xor     eax, eax
0x11668f:    mov     rdi, rbx
0x116692:    call    0x110ec0
0x116697:    xor     eax, eax
0x116699:    mov     rdi, rbx
0x11669c:    call    0x110160
0x1166a1:    cmp     byte ptr [rbx + 0x6d], 0
0x1166a5:    je      0x1166b1
0x1166b1:    xor     eax, eax
0x1166b3:    mov     rdi, rbx
0x1166b6:    call    0x110240
0x1166a7:    mov     rdi, rbx
0x1166aa:    xor     eax, eax
0x1166ac:    call    0x10ec50
0x1166bb:    cmp     byte ptr [rbx + 0x6d], 0
0x1166bf:    je      0x1166cb
0x1166cb:    mov     rsi, qword ptr [rbp + 0x18]
0x1166cf:    mov     rdi, qword ptr [rbx + 0x18]
0x1166d3:    call    0x104b30
0x1166c1:    mov     rdi, rbx
0x1166c4:    xor     eax, eax
0x1166c6:    call    0x110d70
0x1166d8:    test    eax, eax
0x1166da:    je      0x116718
0x116718:    xor     edx, edx
0x11671a:    mov     r8d, 1
0x116720:    mov     esi, 0x22
0x116725:    mov     rdi, rbp
0x116728:    mov     rcx, -1
0x11672f:    call    0x113e30
0x1166dc:    add     rsp, 8

```

```

0x1166e0:    pop     rbx
0x1166e1:    pop     rbp
0x1166e2:    pop     r12
0x1166e4:    pop     r13
0x1166e6:    ret
0x116734:    xor     eax, eax
0x116736:    mov     rdi, rbp
0x116739:    call    0x110ec0
0x11673e:    xor     eax, eax
0x116740:    mov     rdi, rbp
0x116743:    call    0x110160
0x116748:    cmp     byte ptr [rbp + 0x6d], 0
0x11674c:    jne     0x116780
0x11674e:    xor     eax, eax
0x116750:    mov     rdi, rbp
0x116753:    call    0x110240
0x116780:    xor     eax, eax
0x116782:    mov     rdi, rbp
0x116785:    call    0x10ec50
0x116758:    cmp     byte ptr [rbp + 0x6d], 0
0x11675c:    je      0x1166dc
0x11678a:    xor     eax, eax
0x11678c:    mov     rdi, rbp
0x11678f:    call    0x110240
0x116762:    add     rsp, 8
0x116766:    mov     rdi, rbp
0x116769:    xor     eax, eax
0x11676b:    pop     rbx
0x11676c:    pop     rbp
0x11676d:    pop     r12
0x11676f:    pop     r13
0x116771:    jmp     0x110d70
0x116794:    cmp     byte ptr [rbp + 0x6d], 0
0x116798:    je      0x1166dc
0x11679e:    jmp     0x116762
0x1166e7:    nop     word ptr [rax + rax]
0x116715:    nop     dword ptr [rax]
0x116776:    nop     word ptr cs:[rax + rax]
0x1167a0:    endbr64
0x1167a4:    push    r15
0x1167a6:    lea     r15, [rip + 0x13bd3]
0x1167ad:    push    r14
0x1167af:    lea     r14, [rip + 0xacdc]
0x1167b6:    push    r13
0x1167b8:    mov     r13, rsi
0x1167bb:    push    r12
0x1167bd:    mov     r12d, edx
0x1167c0:    push    rbp

```

```

0x1167c1:    mov     rbp, rdi
0x1167c4:    push    rbx
0x1167c5:    sub     rsp, 8
0x1167c9:    mov     rdi, qword ptr [rip + 0x13d08]
0x1167d0:    call    0x10db60
0x1167d5:    xor     eax, eax
0x1167d7:    call    0x10a820
0x1167dc:    mov     eax, dword ptr [rbp + 0x70]
0x1167df:    mov     dword ptr [rbp + 0x50], 0xffffffff
0x1167e6:    mov     dword ptr [rbp + 0x90], 1
0x1167f0:    mov     dword ptr [rbp + 0x54], eax
0x1167f3:    nop     dword ptr [rax + rax]
0x1167f8:    mov     rdi, rbp
0x1167fb:    call    0x1100e0
0x116800:    movsxd  rbx, eax
0x116803:    cmp     ebx, -1
0x116806:    je      0x11686b
0x11686b:    xor     eax, eax
0x11686d:    call    0x10e4e0
0x116808:    xor     eax, eax
0x11680a:    call    0x118ab0
0x116872:    xor     eax, eax
0x116874:    call    0x10a820
0x11680f:    xor     eax, eax
0x116811:    mov     edx, r12d
0x116814:    mov     esi, ebx
0x116816:    mov     rdi, rbp
0x116819:    call    0x1134d0
0x116879:    xor     ecx, ecx
0x11687b:    xor     edx, edx
0x11687d:    mov     r8d, 1
0x116883:    mov     esi, 0x22
0x116888:    mov     rdi, rbp
0x11688b:    call    0x113e30
0x11681e:    cmp     eax, 1
0x116821:    je      0x1168f0
0x116890:    xor     eax, eax
0x116892:    mov     rdi, rbp
0x116895:    call    0x110ec0
0x1168f0:    lea     rax, [rbx + rbx*4]
0x1168f4:    shl     rax, 4
0x1168f8:    add     rax, qword ptr [rbp + 0x38]
0x1168fc:    mov     byte ptr [rax + 0x49], 0
0x116900:    jmp     0x1167f8
0x116827:    cmp     eax, 2
0x11682a:    je      0x11686b
0x11689a:    xor     eax, eax
0x11689c:    mov     rdi, rbp

```



```

0x11689f:    call    0x110160
0x1167f8:    mov     rdi, rbp
0x1167fb:    call    0x1100e0
0x11682c:    cmp     eax, 3
0x11682f:    je      0x1167f8
0x1168a4:    cmp     byte ptr [rbp + 0x6d], 0
0x1168a8:    je      0x1168b4
0x116831:    sub     eax, 1
0x116834:    xor     ecx, ecx
0x116836:    mov     rdi, r14
0x116839:    mov     r8d, 0x1e
0x11683f:    cdqe
0x116841:    mov     rdx, qword ptr [r15 + rax*8]
0x116845:    lea     rax, [rbx + rbx*4]
0x116849:    shl     rax, 4
0x11684d:    add     rax, qword ptr [rbp + 0x38]
0x116851:    mov     rsi, qword ptr [rax]
0x116854:    xor     eax, eax
0x116856:    call    0x10f7e0
0x1168b4:    xor     eax, eax
0x1168b6:    mov     rdi, rbp
0x1168b9:    call    0x110240
0x1168aa:    mov     rdi, rbp
0x1168ad:    xor     eax, eax
0x1168af:    call    0x10ec50
0x11685b:    mov     rdi, rbp
0x11685e:    call    0x1100e0
0x1168be:    cmp     byte ptr [rbp + 0x6d], 0
0x1168c2:    je      0x1168ce
0x116863:    movsxd  rbx, eax
0x116866:    cmp     ebx, -1
0x116869:    jne     0x116808
0x1168ce:    mov     rsi, qword ptr [r13 + 0x18]
0x1168d2:    mov     rdi, qword ptr [rbp + 0x18]
0x1168d6:    call    0x104b30
0x1168c4:    mov     rdi, rbp
0x1168c7:    xor     eax, eax
0x1168c9:    call    0x110d70
0x1168db:    test    eax, eax
0x1168dd:    je      0x116908
0x116908:    xor     ecx, ecx
0x11690a:    xor     edx, edx
0x11690c:    mov     r8d, 1
0x116912:    mov     esi, 0x22
0x116917:    mov     rdi, r13
0x11691a:    call    0x113e30
0x1168df:    add     rsp, 8
0x1168e3:    pop     rbx

```

```

0x1168e4:    pop     rbp
0x1168e5:    pop     r12
0x1168e7:    pop     r13
0x1168e9:    pop     r14
0x1168eb:    pop     r15
0x1168ed:    ret
0x11691f:    xor     eax, eax
0x116921:    mov     rdi, r13
0x116924:    call    0x110ec0
0x116929:    xor     eax, eax
0x11692b:    mov     rdi, r13
0x11692e:    call    0x110160
0x116933:    cmp     byte ptr [r13 + 0x6d], 0
0x116938:    jne     0x116968
0x11693a:    xor     eax, eax
0x11693c:    mov     rdi, r13
0x11693f:    call    0x110240
0x116968:    mov     rdi, r13
0x11696b:    xor     eax, eax
0x11696d:    call    0x10ec50
0x116944:    cmp     byte ptr [r13 + 0x6d], 0
0x116949:    je      0x1168df
0x116972:    jmp     0x11693a
0x11694b:    add     rsp, 8
0x11694f:    mov     rdi, r13
0x116952:    xor     eax, eax
0x116954:    pop     rbx
0x116955:    pop     rbp
0x116956:    pop     r12
0x116958:    pop     r13
0x11695a:    pop     r14
0x11695c:    pop     r15
0x11695e:    jmp     0x110d70
0x1168ee:    nop
0x116905:    nop     dword ptr [rax]
0x116963:    nop     dword ptr [rax + rax]
0x116974:    nop     word ptr cs:[rax + rax]
0x11697e:    nop
0x116980:    push    r15
0x116982:    push    r14
0x116984:    push    r13
0x116986:    push    r12
0x116988:    push    rbp
0x116989:    push    rbx
0x11698a:    sub     rsp, 0x8c8
0x116991:    mov     rdi, qword ptr [rip + 0x14ca8]
0x116998:    mov     r13, qword ptr [rip + 0x13cc1]
0x11699f:    mov     rax, qword ptr fs:[0x28]

```

```

0x1169a8:    mov     qword ptr [rsp + 0x8b8], rax
0x1169b0:    xor     eax, eax
0x1169b2:    lea     rax, [rip + 0x14c8f]
0x1169b9:    mov     qword ptr [rip + 0x13ca0], rax
0x1169c0:    test    rdi, rdi
0x1169c3:    je      0x116a78
0x116a78:    xor     eax, eax
0x116a7a:    mov     rdx, qword ptr [rsp + 0x8b8]
0x116a82:    sub     rdx, qword ptr fs:[0x28]
0x116a8b:    jne     0x116c02
0x1169c9:    lea     rsi, [rsp + 0x20]
0x1169ce:    call    0x118f40
0x116a91:    add     rsp, 0x8c8
0x116a98:    pop     rbx
0x116a99:    pop     rbp
0x116a9a:    pop     r12
0x116a9c:    pop     r13
0x116a9e:    pop     r14
0x116aa0:    pop     r15
0x116aa2:    ret
0x116c02:    call    0x1049d0
0x1169d3:    cmp     eax, -1
0x1169d6:    je      0x116a78
0x1169dc:    mov     rax, qword ptr [rsp + 0x68]
0x1169e1:    mov     rdi, qword ptr [rip + 0x14c58]
0x1169e8:    lea     rsi, [rip + 0xb5da]
0x1169ef:    mov     qword ptr [rsp + 0x10], rax
0x1169f4:    mov     rax, qword ptr [rsp + 0x78]
0x1169f9:    mov     qword ptr [rsp + 0x18], rax
0x1169fe:    call    0x104da0
0x116a03:    mov     r14, rax
0x116a06:    test    rax, rax
0x116a09:    je      0x116a78
0x116a0b:    mov     dword ptr [rsp + 4], 0
0x116a13:    xor     ebp, ebp
0x116a15:    xor     ebx, ebx
0x116a17:    xor     r12d, r12d
0x116a1a:    lea     r15, [rsp + 0xb0]
0x116a22:    nop     word ptr [rax + rax]
0x116a28:    mov     rdx, r14
0x116a2b:    mov     esi, 0x800
0x116a30:    mov     rdi, r15
0x116a33:    call    0x104b00
0x116a38:    test    rax, rax
0x116a3b:    je      0x116ad0
0x116ad0:    mov     rdi, r14
0x116ad3:    call    0x104940
0x116a41:    cmp     byte ptr [rsp + 0xb0], 0

```

```

0x116a49:    je      0x116aa8
0x116ad8:    mov     rdi, qword ptr [rip + 0x14b61]
0x116adf:    lea     rsi, [rsp + 0x10]
0x116ae4:    call    0x1047f0
0x116aa8:    xor     ebx, ebx
0x116aaa:    jmp     0x116a28
0x116a4b:    cmp     dword ptr [r15], 0x6d6f7246
0x116a52:    je      0x116ab0
0x116ae9:    mov     rbp, qword ptr [rip + 0x13b70]
0x116af0:    test    r12d, r12d
0x116af3:    je      0x116b0a
0x116a28:    mov     rdx, r14
0x116a2b:    mov     esi, 0x800
0x116a30:    mov     rdi, r15
0x116a33:    call    0x104b00
0x116ab0:    cmp     byte ptr [r15 + 4], 0x20
0x116ab5:    jne     0x116a54
0x116a54:    test    ebp, ebp
0x116a56:    jne     0x116a28
0x116b0a:    mov     rdi, rbp
0x116b0d:    call    0x1049b0
0x116af5:    cmp     r12d, dword ptr [rsp + 4]
0x116afa:    je      0x116b48
0x116ab7:    add     r12d, 1
0x116abb:    xor     ebp, ebp
0x116abd:    mov     ebx, 1
0x116ac2:    jmp     0x116a28
0x116a58:    test    ebx, ebx
0x116a5a:    je      0x116a28
0x116b12:    mov     rsi, r13
0x116b15:    mov     rdi, rbp
0x116b18:    mov     r8, rax
0x116b1b:    mov     eax, dword ptr [rip + 0x14b2f]
0x116b21:    add     eax, dword ptr [rip + 0x14b2d]
0x116b27:    lea     eax, [rax + r8 + 0x15]
0x116b2c:    mov     dword ptr [rip + 0x14b1a], eax
0x116b32:    call    0x104b30
0x116b48:    lea     rbp, [rip + 0x13adc]
0x116b4f:    mov     qword ptr [rip + 0x13b0a], rbp
0x116b56:    jmp     0x116b0a
0x116afc:    lea     rbp, [rip + 0x13b1d]
0x116b03:    mov     qword ptr [rip + 0x13b56], rbp
0x116b0a:    mov     rdi, rbp
0x116b0d:    call    0x1049b0
0x116a5c:    cmp     dword ptr [r15], 0x74617453
0x116a63:    je      0x116b60
0x116b37:    test    eax, eax
0x116b39:    setne   al

```

```

0x116b3c:    movzx    eax, al
0x116b3f:    jmp      0x116a7a
0x116b60:    cmp      word ptr [r15 + 4], 0x7375
0x116b67:    jne      0x116a69
0x116a69:    xor      ebp, ebp
0x116a6b:    mov      ebx, 1
0x116a70:    jmp      0x116a28
0x116a7a:    mov      rdx, qword ptr [rsp + 0x8b8]
0x116a82:    sub      rdx, qword ptr fs:[0x28]
0x116a8b:    jne      0x116c02
0x116b6d:    cmp      byte ptr [r15 + 6], 0x3a
0x116b72:    jne      0x116a69
0x116b78:    mov      rdi, r15
0x116b7b:    mov      esi, 0x3a
0x116b80:    call     0x104a00
0x116b85:    movsx    rbx, byte ptr [rax + 1]
0x116b8a:    lea      rdi, [rax + 1]
0x116b8e:    test     bl, bl
0x116b90:    je       0x116bf8
0x116bf8:    mov      ebx, 1
0x116bfd:    jmp      0x116a28
0x116b92:    mov      qword ptr [rsp + 8], rdi
0x116b97:    call     0x104f00
0x116b9c:    mov      rdi, qword ptr [rsp + 8]
0x116ba1:    mov      rax, qword ptr [rax]
0x116ba4:    jmp      0x116bbd
0x116bbd:    test     byte ptr [rax + rbx*2 + 1], 0x20
0x116bc2:    jne      0x116bb0
0x116bc4:    mov      esi, 0x52
0x116bc9:    mov      ebp, 1
0x116bce:    mov      ebx, 1
0x116bd3:    call     0x104a00
0x116bb0:    movsx    rbx, byte ptr [rdi + 1]
0x116bb5:    add      rdi, 1
0x116bb9:    test     bl, bl
0x116bbb:    je       0x116bf8
0x116bd8:    test     rax, rax
0x116bdb:    je       0x116a28
0x116be1:    add      dword ptr [rsp + 4], 1
0x116be6:    mov      ebp, 1
0x116beb:    mov      ebx, 1
0x116bf0:    jmp      0x116a28
0x116a72:    nop      word ptr [rax + rax]
0x116aa3:    nop      dword ptr [rax + rax]
0x116aaf:    nop
0x116ac7:    nop      word ptr [rax + rax]
0x116b44:    nop      dword ptr [rax]
0x116b58:    nop      dword ptr [rax + rax]

```

```

0x116ba6:      nop      word ptr cs:[rax + rax]
0x116bf5:      nop      dword ptr [rax]
0x116c07:      nop      word ptr [rax + rax]
0x116c10:      endbr64
0x116c14:      sub      rsp, 0xa8
0x116c1b:      mov      edi, 0x1d
0x116c20:      mov      rax, qword ptr fs:[0x28]
0x116c29:      mov      qword ptr [rsp + 0x98], rax
0x116c31:      xor      eax, eax
0x116c33:      call     0x11d670
0x116c38:      lea      r9, [rip + 0x9950]
0x116c3f:      lea      r8, [rip + 0x991f]
0x116c46:      mov      rdx, -1
0x116c4d:      lea      rcx, [rip + 0xb219]
0x116c54:      mov      esi, 1
0x116c59:      mov      rdi, rax
0x116c5c:      mov      qword ptr [rip + 0x149fd], rax
0x116c63:      xor      eax, eax
0x116c65:      call     0x104f10
0x116c6a:      mov      rdi, qword ptr [rip + 0x149ef]
0x116c71:      call     0x1049b0
0x116c76:      mov      rdi, qword ptr [rip + 0x14e1b]
0x116c7d:      mov      dword ptr [rip + 0x149d5], eax
0x116c83:      call     0x1049b0
0x116c88:      mov      rdi, qword ptr [rip + 0x13d01]
0x116c8f:      mov      dword ptr [rip + 0x149bf], eax
0x116c95:      call     0x1049b0
0x116c9a:      lea      rdi, [rip + 0x9999]
0x116ca1:      mov      dword ptr [rip + 0x149a9], eax
0x116ca7:      call     0x1047e0
0x116cac:      mov      qword ptr [rip + 0x1498d], rax
0x116cb3:      test     rax, rax
0x116cb6:      je       0x116cc3
0x116cc3:      xor      eax, eax
0x116cc5:      call     0x116980
0x116cb8:      mov      rdi, rax
0x116cbb:      mov      rsi, rsp
0x116cbe:      call     0x118f40
0x116cca:      mov      rdi, qword ptr [rip + 0x1398f]
0x116cd1:      call     0x1049b0
0x116cd6:      mov      r8, rax
0x116cd9:      mov      eax, dword ptr [rip + 0x14971]
0x116cdf:      add      eax, dword ptr [rip + 0x1496f]
0x116ce5:      lea      eax, [rax + r8 + 0x15]
0x116cea:      mov      dword ptr [rip + 0x1495c], eax
0x116cf0:      xor      eax, eax
0x116cf2:      call     0x10c590
0x116cf7:      mov      qword ptr [rip + 0x1493a], rax

```

```

0x116cfe:    mov     rax, qword ptr [rsp + 0x98]
0x116d06:    sub     rax, qword ptr fs:[0x28]
0x116d0f:    jne     0x116d19
0x116d11:    add     rsp, 0xa8
0x116d18:    ret
0x116d19:    call    0x1049d0
0x116d1e:    nop
0x116d20:    endbr64
0x116d24:    mov     rdi, qword ptr [rip + 0x1490d]
0x116d2b:    jmp     0x10c5c0
0x116d30:    endbr64
0x116d34:    mov     r8d, edi
0x116d37:    mov     rdi, qword ptr [rip + 0x148fa]
0x116d3e:    mov     edx, esi
0x116d40:    mov     ecx, 1
0x116d45:    xor     esi, esi
0x116d47:    jmp     0x10c5e0
0x116d4c:    nop     dword ptr [rax]
0x116d50:    endbr64
0x116d54:    push    r14
0x116d56:    push    r13
0x116d58:    push    r12
0x116d5a:    push    rbp
0x116d5b:    push    rbx
0x116d5c:    sub     rsp, 0x10
0x116d60:    mov     rax, qword ptr fs:[0x28]
0x116d69:    mov     qword ptr [rsp + 8], rax
0x116d6e:    xor     eax, eax
0x116d70:    lea     rbp, [rsp + 7]
0x116d75:    mov     rdi, rbp
0x116d78:    call    0x10a6a0
0x116d7d:    mov     edx, dword ptr [rip + 0x138c1]
0x116d83:    mov     esi, dword ptr [rip + 0x138b7]
0x116d89:    mov     edi, dword ptr [rip + 0x138b9]
0x116d8f:    call    0x10a570
0x116d94:    mov     rdi, qword ptr [rip + 0x1489d]
0x116d9b:    xor     edx, edx
0x116d9d:    xor     esi, esi
0x116d9f:    call    0x10c680
0x116da4:    mov     edx, dword ptr [rip + 0x148ae]
0x116daa:    mov     rsi, qword ptr [rip + 0x148af]
0x116db1:    mov     rdi, qword ptr [rip + 0x14880]
0x116db8:    call    0x10c600
0x116dbd:    mov     rax, qword ptr [rip + 0x14874]
0x116dc4:    mov     edi, dword ptr [rax + 0xc]
0x116dc7:    add     edi, 1
0x116dca:    movsxd  rdi, edi
0x116dcd:    call    0x11d670

```

```

0x116dd2:    mov     r13, rax
0x116dd5:    xor     eax, eax
0x116dd7:    call    0x116980
0x116ddc:    mov     r14, qword ptr [rip + 0x14855]
0x116de3:    mov     eax, dword ptr [rip + 0x1486f]
0x116de9:    mov     ecx, dword ptr [rip + 0x1485d]
0x116def:    mov     edx, dword ptr [r14 + 0xc]
0x116df3:    lea     esi, [rax + rcx + 2]
0x116df7:    cmp     esi, edx
0x116df9:    jl      0x116e58
0x116e58:    sub     edx, eax
0x116e5a:    mov     rdi, r13
0x116e5d:    mov     esi, 0x20
0x116e62:    mov     ebx, edx
0x116e64:    sub     ebx, ecx
0x116e66:    movsxd  rdx, ebx
0x116e69:    call    0x104a50
0x116dfb:    cmp     eax, edx
0x116dfd:    jl      0x116e30
0x116e6e:    mov     edx, ebx
0x116e70:    mov     rsi, r13
0x116e73:    mov     rdi, r14
0x116e76:    call    0x10c600
0x116e30:    sub     edx, eax
0x116e32:    mov     rdi, r13
0x116e35:    mov     esi, 0x20
0x116e3a:    mov     r12d, edx
0x116e3d:    movsxd  rdx, edx
0x116e40:    call    0x104a50
0x116dff:    mov     rdi, r13
0x116e02:    call    0x118d10
0x116e7b:    add     ebx, dword ptr [rip + 0x147d7]
0x116e81:    mov     rdi, qword ptr [rip + 0x147b0]
0x116e88:    xor     esi, esi
0x116e8a:    mov     edx, ebx
0x116e8c:    call    0x10c680
0x116e45:    mov     edx, r12d
0x116e48:    mov     rsi, r13
0x116e4b:    mov     rdi, r14
0x116e4e:    call    0x10c600
0x116e07:    mov     rdi, rbp
0x116e0a:    call    0x10a6b0
0x116e91:    mov     rdi, qword ptr [rip + 0x147a0]
0x116e98:    mov     edx, 5
0x116e9d:    lea     rsi, [rip + 0x13793]
0x116ea4:    call    0x10c600
0x116e53:    jmp     0x116dff
0x116e0f:    mov     rax, qword ptr [rsp + 8]

```



```

0x116e14:    sub     rax, qword ptr fs:[0x28]
0x116e1d:    jne     0x116fc7
0x116ea9:    mov     rdi, qword ptr [rip + 0x14788]
0x116eb0:    mov     esi, 0x20
0x116eb5:    call    0x10c650
0x116e23:    add     rsp, 0x10
0x116e27:    pop     rbx
0x116e28:    pop     rbp
0x116e29:    pop     r12
0x116e2b:    pop     r13
0x116e2d:    pop     r14
0x116e2f:    ret
0x116fc7:    call    0x1049d0
0x116eba:    mov     edi, dword ptr [rip + 0x1378c]
0x116ec0:    call    0x10a4f0
0x116ec5:    mov     edx, dword ptr [rip + 0x14789]
0x116ecb:    mov     rsi, qword ptr [rip + 0x14bc6]
0x116ed2:    mov     rdi, qword ptr [rip + 0x1475f]
0x116ed9:    call    0x10c600
0x116ede:    mov     rdi, qword ptr [rip + 0x14753]
0x116ee5:    mov     esi, 0x20
0x116eea:    call    0x10c650
0x116eef:    mov     r12, qword ptr [rip + 0x1376a]
0x116ef6:    mov     rdi, r12
0x116ef9:    call    0x1049b0
0x116efe:    mov     rdi, qword ptr [rip + 0x14733]
0x116f05:    mov     rsi, r12
0x116f08:    mov     edx, eax
0x116f0a:    call    0x10c600
0x116f0f:    mov     rdi, qword ptr [rip + 0x14722]
0x116f16:    mov     esi, 0x20
0x116f1b:    call    0x10c650
0x116f20:    mov     edi, dword ptr [rip + 0x1371a]
0x116f26:    call    0x10a4f0
0x116f2b:    mov     rdi, qword ptr [rip + 0x14706]
0x116f32:    mov     edx, 4
0x116f37:    lea     rsi, [rip + 0x136f4]
0x116f3e:    call    0x10c600
0x116f43:    mov     rdi, qword ptr [rip + 0x146ee]
0x116f4a:    mov     esi, 0x20
0x116f4f:    call    0x10c650
0x116f54:    mov     edi, dword ptr [rip + 0x136f6]
0x116f5a:    call    0x10a4f0
0x116f5f:    mov     edx, dword ptr [rip + 0x146eb]
0x116f65:    mov     rsi, qword ptr [rip + 0x13a24]
0x116f6c:    mov     rdi, qword ptr [rip + 0x146c5]
0x116f73:    call    0x10c600
0x116f78:    mov     edi, dword ptr [rip + 0x136c2]

```

```

0x116f7e:    call    0x10a4f0
0x116f83:    mov     rdi, qword ptr [rip + 0x146ae]
0x116f8a:    mov     esi, 0x20
0x116f8f:    call    0x10c650
0x116f94:    xor     edi, edi
0x116f96:    xor     eax, eax
0x116f98:    call    0x116fd0
0x116f9d:    mov     rdi, qword ptr [rip + 0x14694]
0x116fa4:    xor     esi, esi
0x116fa6:    mov     eax, dword ptr [rdi + 0xc]
0x116fa9:    lea     edx, [rax - 1]
0x116fac:    call    0x10c680
0x116fb1:    mov     rdi, qword ptr [rip + 0x14680]
0x116fb8:    mov     esi, 0x20
0x116fbd:    call    0x10c650
0x116fc2:    jmp     0x116dff
0x116e55:    nop     dword ptr [rax]
0x116fcc:    nop     dword ptr [rax]
0x116fd0:    endbr64
0x116fd4:    push    r12
0x116fd6:    push    rbp
0x116fd7:    push    rbx
0x116fd8:    mov     ebx, edi
0x116fda:    sub     rsp, 0x30
0x116fde:    mov     rax, qword ptr fs:[0x28]
0x116fe7:    mov     qword ptr [rsp + 0x28], rax
0x116fec:    xor     eax, eax
0x116fee:    call    0x107870
0x116ff3:    test    eax, eax
0x116ff5:    jne     0x117114
0x116ffb:    mov     eax, dword ptr [rip + 0x1464b]
0x117001:    mov     edx, dword ptr [rip + 0x14651]
0x117007:    lea     edx, [rdx + rax + 2]
0x11700b:    mov     rax, qword ptr [rip + 0x14626]
0x117012:    cmp     edx, dword ptr [rax + 0xc]
0x117015:    jge     0x117114
0x117114:    mov     rax, qword ptr [rsp + 0x28]
0x117119:    sub     rax, qword ptr fs:[0x28]
0x117122:    jne     0x11715b
0x11701b:    test    ebx, ebx
0x11701d:    jne     0x117140
0x117124:    add     rsp, 0x30
0x117128:    pop     rbx
0x117129:    pop     rbp
0x11712a:    pop     r12
0x11712c:    ret
0x11715b:    call    0x1049d0
0x117023:    lea     r12, [rsp + 7]

```

```

0x117028:    xor     eax, eax
0x11702a:    call   0x118330
0x117140:    xor     eax, eax
0x117142:    call   0x116980
0x11702f:    mov     rdi, r12
0x117032:    mov     rbp, rax
0x117035:    call   0x10a6a0
0x117147:    test    eax, eax
0x117149:    je      0x117023
0x11703a:    lea     rsi, [rsp + 0xc]
0x11703f:    lea     rdi, [rsp + 8]
0x117044:    call   0x10a4d0
0x11714f:    xor     eax, eax
0x117151:    call   0x116d50
0x117049:    xor     edi, edi
0x11704b:    call   0x10a660
0x117156:    jmp     0x117023
0x117050:    movsxd  r8, dword ptr [rbp + 8]
0x117054:    mov     r9d, dword ptr [rbp + 4]
0x117058:    lea     rbp, [rsp + 0x10]
0x11705d:    lea     rcx, [rip + 0xa0d4]
0x117064:    mov     esi, 1
0x117069:    mov     rdi, rbp
0x11706c:    mov     rax, r8
0x11706f:    imul    r8, r8, 0x2aaaaaab
0x117076:    cdq
0x117077:    sar     r8, 0x21
0x11707b:    sub     r8d, edx
0x11707e:    lea     edx, [r8 + r8*2]
0x117082:    mov     r8d, eax
0x117085:    shl     edx, 2
0x117088:    sub     r8d, edx
0x11708b:    mov     edx, 0xc
0x117090:    cmovbe  r8d, edx
0x117094:    cmp     eax, 0xc
0x117097:    mov     edx, 0x70
0x11709c:    mov     eax, 0x61
0x1170a1:    cmovge  eax, edx
0x1170a4:    sub     rsp, 8
0x1170a8:    mov     edx, 0x10
0x1170ad:    push    rax
0x1170ae:    xor     eax, eax
0x1170b0:    call   0x104f10
0x1170b5:    mov     rdi, qword ptr [rip + 0x1457c]
0x1170bc:    xor     esi, esi
0x1170be:    mov     eax, dword ptr [rdi + 0xc]
0x1170c1:    lea     edx, [rax - 7]
0x1170c4:    call   0x10c680

```

```

0x1170c9:    mov     edx, dword ptr [rip + 0x13589]
0x1170cf:    mov     esi, dword ptr [rip + 0x1357f]
0x1170d5:    mov     edi, dword ptr [rip + 0x13581]
0x1170db:    call    0x10a570
0x1170e0:    mov     rdi, rbp
0x1170e3:    call    0x1049b0
0x1170e8:    mov     rdi, qword ptr [rip + 0x14549]
0x1170ef:    mov     rsi, rbp
0x1170f2:    mov     edx, eax
0x1170f4:    call    0x10c600
0x1170f9:    mov     esi, dword ptr [rsp + 0x1c]
0x1170fd:    mov     edi, dword ptr [rsp + 0x18]
0x117101:    call    0x10a4b0
0x117106:    mov     rdi, r12
0x117109:    call    0x10a6b0
0x11710e:    pop     rax
0x11710f:    pop     rdx
0x117110:    test    ebx, ebx
0x117112:    jne     0x117130
0x117130:    xor     eax, eax
0x117132:    call    0x10a820
0x117137:    jmp     0x117114
0x11712d:    nop     dword ptr [rax]
0x117139:    nop     dword ptr [rax]
0x117160:    push    rbp
0x117161:    mov     esi, 0x400
0x117166:    push    rbx
0x117167:    lea     rbx, [rip + 0x14512]
0x11716e:    mov     rdi, rbx
0x117171:    sub     rsp, 8
0x117175:    mov     rdx, qword ptr [rip + 0x1490c]
0x11717c:    call    0x104b00
0x117181:    test    rax, rax
0x117184:    je      0x1171d8
0x1171d8:    add     rsp, 8
0x1171dc:    xor     eax, eax
0x1171de:    pop     rbx
0x1171df:    pop     rbp
0x1171e0:    ret
0x117186:    mov     rdi, rbx
0x117189:    call    0x1049b0
0x11718e:    mov     rbp, rax
0x117191:    cmp     rax, 0x3ff
0x117197:    je      0x1171e8
0x1171e8:    mov     rcx, qword ptr [rip + 0x136f9]
0x1171ef:    mov     rdi, qword ptr [rip + 0x1368a]
0x1171f6:    lea     rdx, [rip + 0xaceb]
0x1171fd:    xor     eax, eax

```

```

0x1171ff:    mov     esi, 1
0x117204:    call   0x104e60
0x117199:    mov     esi, 0x23
0x11719e:    mov     rdi, rbx
0x1171a1:    call   0x104a00
0x117209:    nop     dword ptr [rax]
0x1171a6:    test    rax, rax
0x1171a9:    je      0x1171c0
0x117210:    mov     rdi, qword ptr [rip + 0x14871]
0x117217:    call   0x104a90
0x1171c0:    cmp     byte ptr [rbx + rbp - 1], 0xa
0x1171c5:    mov     eax, 1
0x1171ca:    jne     0x1171b3
0x1171ab:    mov     byte ptr [rax], 0
0x1171ae:    mov     eax, 1
0x1171b3:    add     rsp, 8
0x1171b7:    pop     rbx
0x1171b8:    pop     rbp
0x1171b9:    ret
0x11721c:    cmp     eax, 0xa
0x11721f:    je      0x117199
0x1171cc:    mov     byte ptr [rbx + rbp - 1], 0
0x1171d1:    jmp     0x1171b3
0x1171b3:    add     rsp, 8
0x1171b7:    pop     rbx
0x1171b8:    pop     rbp
0x1171b9:    ret
0x117225:    cmp     eax, -1
0x117228:    jne     0x117210
0x11722a:    jmp     0x117199
0x1171ba:    nop     word ptr [rax + rax]
0x1171d3:    nop     dword ptr [rax + rax]
0x1171e1:    nop     dword ptr [rax]
0x11722f:    nop
0x117230:    endbr64
0x117234:    test    rdi, rdi
0x117237:    je      0x117270
0x117270:    xor     eax, eax
0x117272:    ret
0x117239:    sub     rsp, 8
0x11723d:    lea     rsi, [rip + 0xad85]
0x117244:    call   0x104da0
0x117249:    mov     qword ptr [rip + 0x14838], rax
0x117250:    test    rax, rax
0x117253:    je      0x117269
0x117269:    xor     eax, eax
0x11726b:    add     rsp, 8
0x11726f:    ret

```

```

0x117255:    mov     dword ptr [rip + 0x14821], 0xffffffff
0x11725f:    mov     eax, 1
0x117264:    add     rsp, 8
0x117268:    ret
0x117273:    nop     word ptr cs:[rax + rax]
0x11727e:    nop
0x117280:    endbr64
0x117284:    mov     rdi, qword ptr [rip + 0x147fd]
0x11728b:    test    rdi, rdi
0x11728e:    je      0x117298
0x117298:    ret
0x117290:    jmp     0x104940
0x117295:    nop     dword ptr [rax]
0x117299:    nop     dword ptr [rax]
0x1172a0:    endbr64
0x1172a4:    push    rbp
0x1172a5:    xor     edx, edx
0x1172a7:    xor     esi, esi
0x1172a9:    lea     rbp, [rip + 0x143d0]
0x1172b0:    push    rbx
0x1172b1:    mov     rbx, rdi
0x1172b4:    sub     rsp, 8
0x1172b8:    mov     rdi, qword ptr [rip + 0x147c9]
0x1172bf:    call    0x104c80
0x1172c4:    jmp     0x1172df
0x1172df:    xor     eax, eax
0x1172e1:    call    0x117160
0x1172e6:    test    eax, eax
0x1172e8:    jne     0x1172d0
0x1172ea:    mov     eax, 0xffffffff
0x1172ef:    mov     dword ptr [rip + 0x1478b], eax
0x1172f5:    add     rsp, 8
0x1172f9:    pop     rbx
0x1172fa:    pop     rbp
0x1172fb:    ret
0x1172d0:    mov     rsi, rbp
0x1172d3:    mov     rdi, rbx
0x1172d6:    call    0x104b30
0x1172db:    test    eax, eax
0x1172dd:    je      0x117300
0x117300:    mov     rdi, qword ptr [rip + 0x14781]
0x117307:    call    0x104b80
0x11730c:    mov     dword ptr [rip + 0x1476e], eax
0x117312:    add     rsp, 8
0x117316:    pop     rbx
0x117317:    pop     rbp
0x117318:    ret
0x1172c6:    nop     word ptr cs:[rax + rax]

```

```

0x1172fc:      nop      dword ptr [rax]
0x117319:      nop      dword ptr [rax]
0x117320:      endbr64
0x117324:      push     r15
0x117326:      push     r14
0x117328:      push     r13
0x11732a:      mov      r13, rdi
0x11732d:      push     r12
0x11732f:      push     rbp
0x117330:      push     rbx
0x117331:      mov      ebx, ecx
0x117333:      sub      rsp, 0x438
0x11733a:      mov      qword ptr [rsp + 0x10], rsi
0x11733f:      mov      dword ptr [rsp + 0xc], edx
0x117343:      mov      rax, qword ptr fs:[0x28]
0x11734c:      mov      qword ptr [rsp + 0x428], rax
0x117354:      xor      eax, eax
0x117356:      test     ecx, ecx
0x117358:      jne      0x117618
0x11735e:      cmp      dword ptr [rsp + 0xc], 1
0x117363:      jne      0x1175e0
0x117618:      movsxd   rsi, dword ptr [rip + 0x14461]
0x11761f:      mov      rdi, qword ptr [rip + 0x14462]
0x117626:      xor      edx, edx
0x117628:      call     0x104c80
0x117369:      mov      rax, qword ptr [rsp + 0x10]
0x11736e:      mov      qword ptr [rax], 0
0x117375:      lea      rbp, [rip + 0x14304]
0x11737c:      lea      r12, [rsp + 0x20]
0x117381:      lea      r14, [rip + 0xa092]
0x117388:      nop      dword ptr [rax + rax]
0x117390:      xor      eax, eax
0x117392:      call     0x117160
0x1175e0:      movsxd   rdx, dword ptr [rsp + 0xc]
0x1175e5:      mov      rdi, qword ptr [rsp + 0x10]
0x1175ea:      xor      esi, esi
0x1175ec:      shl      rdx, 3
0x1175f0:      call     0x104a50
0x11762d:      jmp      0x11735e
0x117397:      test     eax, eax
0x117399:      je       0x117600
0x1175f5:      jmp      0x117375
0x117600:      test     ebx, ebx
0x117602:      jne      0x1173df
0x11739f:      cmp      byte ptr [rbp], 0
0x1173a3:      je       0x117600
0x117375:      lea      rbp, [rip + 0x14304]
0x11737c:      lea      r12, [rsp + 0x20]

```

```

0x117381:    lea     r14, [rip + 0xa092]
0x117388:    nop     dword ptr [rax + rax]
0x117390:    xor     eax, eax
0x117392:    call    0x117160
0x117608:    mov     byte ptr [r13], 0
0x11760d:    jmp     0x1173df
0x1173df:    mov     rax, qword ptr [rsp + 0x428]
0x1173e7:    sub     rax, qword ptr fs:[0x28]
0x1173f0:    jne     0x117657
0x1173a9:    mov     rdx, r12
0x1173ac:    mov     rsi, r14
0x1173af:    mov     rdi, rbp
0x1173b2:    xor     eax, eax
0x1173b4:    mov     byte ptr [rsp + 0x20], 0
0x1173b9:    call    0x104c60
0x1173f6:    add     rsp, 0x438
0x1173fd:    pop     rbx
0x1173fe:    pop     rbp
0x1173ff:    pop     r12
0x117401:    pop     r13
0x117403:    pop     r14
0x117405:    pop     r15
0x117407:    ret
0x117657:    call    0x1049d0
0x1173be:    call    0x104f00
0x1173c3:    mov     r15, rax
0x1173c6:    test    ebx, ebx
0x1173c8:    jne     0x117410
0x1173ca:    movsx    rcx, byte ptr [rsp + 0x20]
0x1173d0:    mov     rax, qword ptr [rax]
0x1173d3:    mov     byte ptr [rsp + 0x40], 0
0x1173d8:    test    byte ptr [rax + rcx*2 + 1], 0x40
0x1173dd:    jne     0x117433
0x117410:    movsx    rcx, byte ptr [rsp + 0x20]
0x117416:    mov     rax, qword ptr [rax]
0x117419:    test    byte ptr [rax + rcx*2 + 1], 0x40
0x11741e:    je      0x1173df
0x117433:    mov     esi, 0x3d
0x117438:    mov     rdi, rbp
0x11743b:    call    0x104a00
0x117420:    mov     rsi, r12
0x117423:    mov     rdi, r13
0x117426:    call    0x104b30
0x117440:    test    rax, rax
0x117443:    je      0x1174e3
0x11742b:    test    eax, eax
0x11742d:    jne     0x117390
0x1174e3:    mov     eax, dword ptr [rsp + 0xc]

```



```

0x1174e7:    test    eax, eax
0x1174e9:    jle     0x1175c0
0x117449:    cmp     byte ptr [rax + 1], 0
0x11744d:    je      0x1174e3
0x117390:    xor     eax, eax
0x117392:    call    0x117160
0x1175c0:    test    ebx, ebx
0x1175c2:    jne     0x1173df
0x1174ef:    mov     rsi, qword ptr [rsp + 0x10]
0x1174f4:    xor     r8d, r8d
0x1174f7:    jmp     0x117513
0x117453:    mov     rdx, qword ptr [rsp + 0x10]
0x117458:    lea     rdi, [rax + 1]
0x11745c:    mov     qword ptr [rdx], rdi
0x11745f:    movzx   ecx, byte ptr [rax + 1]
0x117463:    test    cl, cl
0x117465:    je      0x11763f
0x1175c8:    mov     rsi, r12
0x1175cb:    mov     rdi, r13
0x1175ce:    call    0x1048b0
0x117513:    mov     rbp, qword ptr [rsi]
0x117516:    test    rbp, rbp
0x117519:    je      0x117500
0x11763f:    mov     esi, 0x3b
0x117644:    call    0x104a00
0x11746b:    cmp     dword ptr [rsp + 0xc], 1
0x117470:    jle     0x11763f
0x1175d3:    jmp     0x1173df
0x117500:    add     r8d, 1
0x117504:    add     rsi, 8
0x117508:    cmp     dword ptr [rsp + 0xc], r8d
0x11750d:    jle     0x1175c0
0x11751b:    mov     r14, qword ptr [r15]
0x11751e:    movsx   rax, byte ptr [rbp]
0x117523:    test    byte ptr [r14 + rax*2 + 1], 0x20
0x117529:    je      0x117547
0x117649:    test    rax, rax
0x11764c:    jne     0x1174e0
0x117476:    mov     r8d, 1
0x11747c:    jmp     0x11748e
0x117547:    mov     rdi, rbp
0x11754a:    mov     qword ptr [rsp + 0x18], rsi
0x11754f:    mov     dword ptr [rsp + 0x10], r8d
0x117554:    call    0x1049b0
0x11752b:    lea     rax, [rbp + 1]
0x11752f:    nop
0x117530:    mov     rbp, rax
0x117533:    mov     qword ptr [rsi], rax

```

```

0x117536:      add     rax, 1
0x11753a:      movsx   rcx, byte ptr [rbp]
0x11753f:      test    byte ptr [r14 + rcx*2 + 1], 0x20
0x117545:      jne      0x117530
0x117652:      jmp      0x1174e3
0x1174e0:      mov      byte ptr [rax], 0
0x1174e3:      mov      eax, dword ptr [rsp + 0xc]
0x1174e7:      test     eax, eax
0x1174e9:      jle      0x1175c0
0x11748e:      movzx    eax, byte ptr [rdi + 1]
0x117492:      test     al, al
0x117494:      setne    sil
0x117498:      add      rdi, 1
0x11749c:      cmp      cl, 0x3b
0x11749f:      jne      0x117480
0x117559:      mov      r8d, dword ptr [rsp + 0x10]
0x11755e:      mov      rsi, qword ptr [rsp + 0x18]
0x117563:      lea      rax, [rbp + rax - 1]
0x117568:      movsx    rcx, byte ptr [rax]
0x11756c:      test     byte ptr [r14 + rcx*2 + 1], 0x20
0x117572:      jne      0x117591
0x117530:      mov      rbp, rax
0x117533:      mov      qword ptr [rsi], rax
0x117536:      add      rax, 1
0x11753a:      movsx    rcx, byte ptr [rbp]
0x11753f:      test     byte ptr [r14 + rcx*2 + 1], 0x20
0x117545:      jne      0x117530
0x1174a1:      mov      byte ptr [rdi - 1], 0
0x1174a5:      cmp      al, 0x3b
0x1174a7:      je       0x117638
0x117480:      mov      ecx, eax
0x117482:      cmp      r8d, dword ptr [rsp + 0xc]
0x117487:      jge      0x1174d1
0x117574:      jmp      0x117596
0x117591:      cmp      rax, rbp
0x117594:      jae      0x117580
0x117638:      mov      ecx, eax
0x11763a:      jmp      0x1174cb
0x1174ad:      test     sil, sil
0x1174b0:      je       0x117638
0x117489:      test     sil, sil
0x11748c:      je       0x1174d1
0x1174d1:      mov      esi, 0x3b
0x1174d6:      call     0x104a00
0x117596:      mov      byte ptr [rax + 1], 0
0x11759a:      mov      rax, qword ptr [rsi]
0x11759d:      cmp      byte ptr [rax], 0
0x1175a0:      jne      0x117500

```

```

0x117580:    movsx    rcx, byte ptr [rax - 1]
0x117585:    sub      rax, 1
0x117589:    test     byte ptr [r14 + rcx*2 + 1], 0x20
0x11758f:    je       0x117596
0x1174cb:    add      r8d, 1
0x1174cf:    jmp      0x117482
0x1174b6:    mov      rdx, qword ptr [rsp + 0x10]
0x1174bb:    movsxd   rax, r8d
0x1174be:    mov      qword ptr [rdx + rax*8], rdi
0x1174c2:    movzx    ecx, byte ptr [rdi]
0x1174c5:    test     cl, cl
0x1174c7:    setne    sil
0x1174cb:    add      r8d, 1
0x1174cf:    jmp      0x117482
0x1174db:    test     rax, rax
0x1174de:    je       0x1174ef
0x1175a6:    mov      qword ptr [rsi], 0
0x1175ad:    add      r8d, 1
0x1175b1:    add      rsi, 8
0x1175b5:    cmp      dword ptr [rsp + 0xc], r8d
0x1175ba:    jg       0x117513
0x117482:    cmp      r8d, dword ptr [rsp + 0xc]
0x117487:    jge      0x1174d1
0x117408:    nop      dword ptr [rax + rax]
0x11747e:    nop
0x1174f9:    nop      dword ptr [rax]
0x117576:    nop      word ptr cs:[rax + rax]
0x1175d8:    nop      dword ptr [rax + rax]
0x1175fa:    nop      word ptr [rax + rax]
0x117612:    nop      word ptr [rax + rax]
0x117632:    nop      word ptr [rax + rax]
0x11765c:    nop      dword ptr [rax]
0x117660:    endbr64
0x117664:    push     r12
0x117666:    mov      edi, 0xf
0x11766b:    lea      r12, [rip + 0xa8b4]
0x117672:    call     0x11d670
0x117677:    mov      rcx, r12
0x11767a:    lea      r9, [rip + 0xa895]
0x117681:    lea      r8, [rip + 0xa893]
0x117688:    mov      rdi, rax
0x11768b:    mov      rdx, -1
0x117692:    mov      esi, 1
0x117697:    mov      qword ptr [rip + 0x14402], rax
0x11769e:    xor      eax, eax
0x1176a0:    call     0x104f10
0x1176a5:    mov      edi, 0x17
0x1176aa:    call     0x11d670

```

```

0x1176af:    mov     rcx, r12
0x1176b2:    pop     r12
0x1176b4:    lea     r9, [rip + 0xa870]
0x1176bb:    mov     qword ptr [rip + 0x143e6], rax
0x1176c2:    mov     rdi, rax
0x1176c5:    xor     eax, eax
0x1176c7:    lea     r8, [rip + 0xa84d]
0x1176ce:    mov     rdx, -1
0x1176d5:    mov     esi, 1
0x1176da:    jmp     0x104f10
0x1176df:    nop
0x1176e0:    endbr64
0x1176e4:    push    rbp
0x1176e5:    mov     rbp, rdi
0x1176e8:    xor     edi, edi
0x1176ea:    push    rbx
0x1176eb:    sub     rsp, 0x18
0x1176ef:    mov     rax, qword ptr fs:[0x28]
0x1176f8:    mov     qword ptr [rsp + 8], rax
0x1176fd:    xor     eax, eax
0x1176ff:    call    0x104be0
0x117704:    mov     rdi, rsp
0x117707:    mov     qword ptr [rsp], rax
0x11770b:    call    0x104840
0x117710:    mov     rbx, rax
0x117713:    call    0x104920
0x117718:    movsxd  rcx, dword ptr [rbx + 0x10]
0x11771c:    sub     rsp, 8
0x117720:    lea     rdx, [rip + 0x12fd9]
0x117727:    mov     r9d, dword ptr [rbx + 8]
0x11772b:    mov     r8d, dword ptr [rbx + 0xc]
0x11772f:    mov     esi, 1
0x117734:    mov     rcx, qword ptr [rdx + rcx*8]
0x117738:    push    rbp
0x117739:    lea     rdx, [rip + 0xa8d0]
0x117740:    push    rax
0x117741:    mov     rdi, qword ptr [rip + 0x13138]
0x117748:    push    qword ptr [rip + 0x1319a]
0x11774e:    mov     eax, dword ptr [rbx]
0x117750:    push    rax
0x117751:    mov     eax, dword ptr [rbx + 4]
0x117754:    push    rax
0x117755:    xor     eax, eax
0x117757:    call    0x104e60
0x11775c:    add     rsp, 0x30
0x117760:    mov     rax, qword ptr [rsp + 8]
0x117765:    sub     rax, qword ptr fs:[0x28]
0x11776e:    jne     0x117777

```

```

0x117770:      add     rsp, 0x18
0x117774:      pop     rbx
0x117775:      pop     rbp
0x117776:      ret
0x117777:      call    0x1049d0
0x11777c:      nop     dword ptr [rax]
0x117780:      endbr64
0x117784:      push    rbp
0x117785:      xor     eax, eax
0x117787:      mov     ebp, edi
0x117789:      call    0x108ad0
0x11778e:      cmp     ebp, 0xb
0x117791:      je      0x1177d3
0x1177d3:      lea     rdi, [rip + 0xa766]
0x1177da:      call    0x1176e0
0x117793:      jg      0x1177c0
0x1177df:      mov     rcx, qword ptr [rip + 0x13102]
0x1177e6:      mov     rdi, qword ptr [rip + 0x13093]
0x1177ed:      lea     rdx, [rip + 0xa88c]
0x1177f4:      xor     eax, eax
0x1177f6:      mov     esi, 1
0x1177fb:      call    0x104e60
0x117795:      lea     eax, [rbp - 1]
0x117798:      cmp     eax, 1
0x11779b:      ja      0x117802
0x1177c0:      cmp     ebp, 0xf
0x1177c3:      jne     0x117802
0x117800:      jmp     0x1177b7
0x11779d:      cmp     ebp, 1
0x1177a0:      lea     rdi, [rip + 0xa791]
0x1177a7:      lea     rax, [rip + 0xa78e]
0x1177ae:      cmovne  rdi, rax
0x1177b2:      call    0x1176e0
0x117802:      mov     rcx, qword ptr [rip + 0x130df]
0x117809:      mov     rdi, qword ptr [rip + 0x13070]
0x117810:      mov     r8d, ebp
0x117813:      xor     eax, eax
0x117815:      lea     rdx, [rip + 0xa82c]
0x11781c:      mov     esi, 1
0x117821:      call    0x104e60
0x1177c5:      lea     rdi, [rip + 0x8dbe]
0x1177cc:      call    0x1176e0
0x1177b7:      mov     edi, ebp
0x1177b9:      call    0x104e20
0x117826:      jmp     0x1177df
0x1177d1:      jmp     0x1177b7
0x1177be:      nop
0x117828:      nop     dword ptr [rax + rax]

```

```

0x117830:      endbr64
0x117834:      push     rbx
0x117835:      mov      rcx, qword ptr [rip + 0x130ac]
0x11783c:      mov      rbx, rdi
0x11783f:      mov      esi, 1
0x117844:      mov      rdi, qword ptr [rip + 0x13035]
0x11784b:      lea      rdx, [rip + 0xa6f3]
0x117852:      xor      eax, eax
0x117854:      call     0x104e60
0x117859:      mov      r8, rbx
0x11785c:      mov      rcx, qword ptr [rip + 0x13085]
0x117863:      pop      rbx
0x117864:      mov      rdi, qword ptr [rip + 0x13015]
0x11786b:      lea      rdx, [rip + 0xa836]
0x117872:      mov      esi, 1
0x117877:      xor      eax, eax
0x117879:      jmp      0x104e60
0x11787e:      nop
0x117880:      endbr64
0x117884:      sub      rsp, 8
0x117888:      mov      r8, rdi
0x11788b:      mov      rcx, qword ptr [rip + 0x13056]
0x117892:      xor      eax, eax
0x117894:      mov      rdi, qword ptr [rip + 0x12fe5]
0x11789b:      lea      rdx, [rip + 0xa836]
0x1178a2:      mov      esi, 1
0x1178a7:      call     0x104e60
0x1178ac:      mov      rcx, qword ptr [rip + 0x13035]
0x1178b3:      mov      esi, 1
0x1178b8:      xor      eax, eax
0x1178ba:      mov      rdi, qword ptr [rip + 0x12fbf]
0x1178c1:      lea      rdx, [rip + 0xa840]
0x1178c8:      call     0x104e60
0x1178cd:      mov      rcx, qword ptr [rip + 0x13014]
0x1178d4:      mov      esi, 1
0x1178d9:      xor      eax, eax
0x1178db:      mov      rdi, qword ptr [rip + 0x12f9e]
0x1178e2:      lea      rdx, [rip + 0xa84f]
0x1178e9:      call     0x104e60
0x1178ee:      mov      rcx, qword ptr [rip + 0x12ff3]
0x1178f5:      mov      esi, 1
0x1178fa:      xor      eax, eax
0x1178fc:      mov      rdi, qword ptr [rip + 0x12f7d]
0x117903:      lea      rdx, [rip + 0xa86e]
0x11790a:      call     0x104e60
0x11790f:      mov      rcx, qword ptr [rip + 0x12fd2]
0x117916:      mov      esi, 1
0x11791b:      xor      eax, eax

```

```

0x11791d:    mov     rdi, qword ptr [rip + 0x12f5c]
0x117924:    lea     rdx, [rip + 0xa885]
0x11792b:    call    0x104e60
0x117930:    mov     rcx, qword ptr [rip + 0x12fb1]
0x117937:    mov     esi, 1
0x11793c:    xor     eax, eax
0x11793e:    mov     rdi, qword ptr [rip + 0x12f3b]
0x117945:    lea     rdx, [rip + 0xa8a4]
0x11794c:    call    0x104e60
0x117951:    mov     rcx, qword ptr [rip + 0x12f90]
0x117958:    mov     esi, 1
0x11795d:    xor     eax, eax
0x11795f:    mov     rdi, qword ptr [rip + 0x12f1a]
0x117966:    lea     rdx, [rip + 0xa8c3]
0x11796d:    call    0x104e60
0x117972:    mov     rcx, qword ptr [rip + 0x12f6f]
0x117979:    mov     esi, 1
0x11797e:    xor     eax, eax
0x117980:    mov     rdi, qword ptr [rip + 0x12ef9]
0x117987:    lea     rdx, [rip + 0xa8ea]
0x11798e:    add     rsp, 8
0x117992:    jmp     0x104e60
0x117997:    nop     word ptr [rax + rax]
0x1179a0:    endbr64
0x1179a4:    push    r12
0x1179a6:    lea     r12, [rip + 0x12d33]
0x1179ad:    push    rbp
0x1179ae:    sub     rsp, 8
0x1179b2:    mov     rdi, qword ptr [rip + 0x140ef]
0x1179b9:    call    0x1049b0
0x1179be:    mov     rdi, r12
0x1179c1:    mov     rbp, rax
0x1179c4:    call    0x1049b0
0x1179c9:    lea     rdi, [rbp + rax + 9]
0x1179ce:    call    0x11d670
0x1179d3:    mov     rsi, qword ptr [rip + 0x140ce]
0x1179da:    mov     rdi, rax
0x1179dd:    mov     rbp, rax
0x1179e0:    call    0x104980
0x1179e5:    mov     rsi, r12
0x1179e8:    mov     rdi, rax
0x1179eb:    call    0x104980
0x1179f0:    mov     edx, 0x6e6f
0x1179f5:    mov     rdi, rbp
0x1179f8:    mov     dword ptr [rax], 0x6d6d6f63
0x1179fe:    mov     word ptr [rax + 4], dx
0x117a02:    mov     byte ptr [rax + 6], 0
0x117a06:    call    0x117230

```

```

0x117a0b:    test    eax, eax
0x117a0d:    je      0x117a17
0x117a17:    mov     rdi, rbp
0x117a1a:    call    0x117830
0x117a0f:    add     rsp, 8
0x117a13:    pop     rbp
0x117a14:    pop     r12
0x117a16:    ret
0x117a1f:    mov     edi, 1
0x117a24:    call    0x104e20
0x117a29:    nop     dword ptr [rax]
0x117a30:    endbr64
0x117a34:    push    r13
0x117a36:    push    r12
0x117a38:    lea     r12, [rip + 0x12c91]
0x117a3f:    push    rbp
0x117a40:    push    rbx
0x117a41:    sub     rsp, 8
0x117a45:    mov     rdi, qword ptr [rip + 0x12ea4]
0x117a4c:    call    0x1049b0
0x117a51:    mov     rdi, r12
0x117a54:    mov     rbx, rax
0x117a57:    call    0x1049b0
0x117a5c:    mov     rdi, qword ptr [rip + 0x12eed]
0x117a63:    mov     rbp, rax
0x117a66:    call    0x1049b0
0x117a6b:    lea     rdi, [rbx + rbp + 2]
0x117a70:    add     rdi, rax
0x117a73:    call    0x11d670
0x117a78:    mov     rsi, qword ptr [rip + 0x12e71]
0x117a7f:    mov     rdi, rax
0x117a82:    mov     rbp, rax
0x117a85:    call    0x104980
0x117a8a:    mov     rsi, r12
0x117a8d:    mov     rdi, rax
0x117a90:    call    0x104980
0x117a95:    mov     rsi, qword ptr [rip + 0x12eb4]
0x117a9c:    mov     rdi, rax
0x117a9f:    call    0x1048b0
0x117aa4:    mov     rdi, rbp
0x117aa7:    call    0x117230
0x117aac:    test    eax, eax
0x117aae:    jne     0x117b2a
0x117ab0:    mov     rdi, rbp
0x117ab3:    lea     r12, [rip + 0x12c06]
0x117aba:    call    0x118d10
0x117b2a:    mov     rdi, rbp
0x117b2d:    mov     r12d, 1

```



```

0x117b33:    call    0x118d10
0x117abf:    mov     rdi, qword ptr [rip + 0x12e2a]
0x117ac6:    call    0x1049b0
0x117b38:    add     rsp, 8
0x117b3c:    mov     eax, r12d
0x117b3f:    pop     rbx
0x117b40:    pop     rbp
0x117b41:    pop     r12
0x117b43:    pop     r13
0x117b45:    ret
0x117acb:    mov     rdi, r12
0x117ace:    mov     rbp, rax
0x117ad1:    call    0x1049b0
0x117ad6:    mov     rdi, qword ptr [rip + 0x12e73]
0x117add:    mov     rbx, rax
0x117ae0:    call    0x1049b0
0x117ae5:    lea     rdi, [rbp + rbx + 2]
0x117aea:    add     rdi, rax
0x117aed:    call    0x11d670
0x117af2:    mov     rsi, qword ptr [rip + 0x12df7]
0x117af9:    mov     rdi, rax
0x117afc:    mov     rbp, rax
0x117aff:    call    0x104980
0x117b04:    mov     rsi, r12
0x117b07:    mov     rdi, rax
0x117b0a:    call    0x104980
0x117b0f:    mov     rsi, qword ptr [rip + 0x12e3a]
0x117b16:    mov     rdi, rax
0x117b19:    call    0x1048b0
0x117b1e:    mov     rdi, rbp
0x117b21:    call    0x117230
0x117b26:    test    eax, eax
0x117b28:    je      0x117b50
0x117b50:    mov     rdi, rbp
0x117b53:    lea     r13, [rip + 0x12b86]
0x117b5a:    call    0x118d10
0x117b5f:    mov     rdi, qword ptr [rip + 0x13f42]
0x117b66:    call    0x1049b0
0x117b6b:    mov     rdi, r13
0x117b6e:    mov     rbp, rax
0x117b71:    call    0x1049b0
0x117b76:    mov     rdi, qword ptr [rip + 0x12dd3]
0x117b7d:    mov     rbx, rax
0x117b80:    call    0x1049b0
0x117b85:    lea     rdi, [rbp + rbx + 2]
0x117b8a:    add     rdi, rax
0x117b8d:    call    0x11d670
0x117b92:    mov     rsi, qword ptr [rip + 0x13f0f]

```

```

0x117b99:    mov     rdi, rax
0x117b9c:    mov     rbp, rax
0x117b9f:    call    0x104980
0x117ba4:    mov     rsi, r13
0x117ba7:    mov     rdi, rax
0x117baa:    call    0x104980
0x117baf:    mov     rsi, qword ptr [rip + 0x12d9a]
0x117bb6:    mov     rdi, rax
0x117bb9:    call    0x1048b0
0x117bbe:    mov     rdi, rbp
0x117bc1:    call    0x117230
0x117bc6:    mov     r12d, eax
0x117bc9:    test    eax, eax
0x117bcb:    jne     0x117b2a
0x117bd1:    xor     eax, eax
0x117bd3:    mov     rdi, rbp
0x117bd6:    call    0x117880
0x117bdb:    mov     rdi, rbp
0x117bde:    call    0x118d10
0x117be3:    mov     rdi, qword ptr [rip + 0x13ebe]
0x117bea:    call    0x1049b0
0x117bef:    mov     rdi, r13
0x117bf2:    mov     rbx, rax
0x117bf5:    call    0x1049b0
0x117bfa:    lea     rdi, [rbx + rax + 0xa]
0x117bff:    call    0x11d670
0x117c04:    mov     rsi, qword ptr [rip + 0x13e9d]
0x117c0b:    mov     rdi, rax
0x117c0e:    mov     rbp, rax
0x117c11:    call    0x104980
0x117c16:    mov     rsi, r13
0x117c19:    mov     rdi, rax
0x117c1c:    call    0x104980
0x117c21:    mov     rdi, rbp
0x117c24:    movabs  rdx, 0x636972656e6567
0x117c2e:    mov     qword ptr [rax], rdx
0x117c31:    call    0x117230
0x117c36:    test    eax, eax
0x117c38:    jne     0x117b38
0x117c3e:    mov     rdi, rbp
0x117c41:    call    0x117830
0x117c46:    mov     edi, 1
0x117c4b:    call    0x104e20
0x117b46:    nop     word ptr cs:[rax + rax]
0x117c50:    endbr64
0x117c54:    push    rbp
0x117c55:    push    rbx
0x117c56:    mov     rbx, rdi

```

```

0x117c59:    sub     rsp, 8
0x117c5d:    call    0x1172a0
0x117c62:    cmp     eax, -1
0x117c65:    je      0x117c70
0x117c70:    mov     edx, 7
0x117c75:    lea     rsi, [rip + 0xa2e5]
0x117c7c:    mov     rdi, rbx
0x117c7f:    call    0x104890
0x117c67:    add     rsp, 8
0x117c6b:    pop     rbx
0x117c6c:    pop     rbp
0x117c6d:    ret
0x117c84:    test    eax, eax
0x117c86:    jne     0x117cd4
0x117c88:    lea     rdi, [rbx + 5]
0x117c8c:    call    0x1049b0
0x117cd4:    mov     rdi, qword ptr [rip + 0x12ba5]
0x117cdb:    mov     rcx, qword ptr [rip + 0x12c06]
0x117ce2:    mov     r8, rbx
0x117ce5:    xor     eax, eax
0x117ce7:    lea     rdx, [rip + 0xa5ba]
0x117cee:    mov     esi, 1
0x117cf3:    call    0x104e60
0x117c91:    lea     rdi, [rax + 6]
0x117c95:    call    0x11d670
0x117cf8:    mov     edi, 1
0x117cfd:    call    0x104e20
0x117c9a:    lea     rsi, [rbx + 7]
0x117c9e:    mov     dword ptr [rax], 0x5449475b
0x117ca4:    mov     rbp, rax
0x117ca7:    lea     rdi, [rax + 5]
0x117cab:    mov     byte ptr [rax + 4], 0x2d
0x117caf:    call    0x1048b0
0x117cb4:    mov     rdi, rbp
0x117cb7:    call    0x1172a0
0x117cbc:    mov     rdi, rbp
0x117cbf:    cmp     eax, -1
0x117cc2:    je      0x117ccf
0x117ccf:    call    0x118d10
0x117cc4:    add     rsp, 8
0x117cc8:    pop     rbx
0x117cc9:    pop     rbp
0x117cca:    jmp     0x118d10
0x117c6e:    nop
0x117d02:    nop     word ptr cs:[rax + rax]
0x117d0d:    nop     dword ptr [rax]
0x117d10:    endbr64
0x117d14:    push    rbx

```

```

0x117d15:    mov     ecx, 1
0x117d1a:    mov     ebx, esi
0x117d1c:    mov     edx, 1
0x117d21:    sub     rsp, 0x10
0x117d25:    mov     rax, qword ptr fs:[0x28]
0x117d2e:    mov     qword ptr [rsp + 8], rax
0x117d33:    xor     eax, eax
0x117d35:    mov     rsi, rsp
0x117d38:    call    0x117320
0x117d3d:    mov     rdi, qword ptr [rsp]
0x117d41:    mov     eax, ebx
0x117d43:    test    rdi, rdi
0x117d46:    je      0x117d54
0x117d54:    mov     rdx, qword ptr [rsp + 8]
0x117d59:    sub     rdx, qword ptr fs:[0x28]
0x117d62:    jne     0x117d6a
0x117d48:    mov     edx, 0xa
0x117d4d:    xor     esi, esi
0x117d4f:    call    0x104ba0
0x117d64:    add     rsp, 0x10
0x117d68:    pop     rbx
0x117d69:    ret
0x117d6a:    call    0x1049d0
0x117d6f:    nop
0x117d70:    endbr64
0x117d74:    push    r15
0x117d76:    push    r14
0x117d78:    mov     r14, rdi
0x117d7b:    push    r13
0x117d7d:    mov     r13d, ecx
0x117d80:    mov     ecx, 1
0x117d85:    push    r12
0x117d87:    mov     r12d, r13d
0x117d8a:    push    rbp
0x117d8b:    mov     ebp, edx
0x117d8d:    mov     edx, 1
0x117d92:    push    rbx
0x117d93:    mov     rbx, rsi
0x117d96:    sub     rsp, 0x18
0x117d9a:    mov     rax, qword ptr fs:[0x28]
0x117da3:    mov     qword ptr [rsp + 8], rax
0x117da8:    xor     eax, eax
0x117daa:    mov     rsi, rsp
0x117dad:    call    0x117320
0x117db2:    mov     r15, qword ptr [rsp]
0x117db6:    test    r15, r15
0x117db9:    je      0x117de4
0x117de4:    mov     rax, qword ptr [rsp + 8]

```

```

0x117de9:    sub     rax, qword ptr fs:[0x28]
0x117df2:    jne     0x117e43
0x117dbb:    test    ebp, ebp
0x117dbd:    jle     0x117e3c
0x117df4:    add     rsp, 0x18
0x117df8:    mov     eax, r12d
0x117dfb:    pop     rbx
0x117dfc:    pop     rbp
0x117dfd:    pop     r12
0x117dff:    pop     r13
0x117e01:    pop     r14
0x117e03:    pop     r15
0x117e05:    ret
0x117e43:    call    0x1049d0
0x117e3c:    je      0x117e10
0x117dbf:    xor     r12d, r12d
0x117dc2:    jmp     0x117dd5
0x117e10:    mov     rcx, qword ptr [rip + 0x12ad1]
0x117e17:    mov     r9, r15
0x117e1a:    mov     r8, r14
0x117e1d:    xor     eax, eax
0x117e1f:    mov     rdi, qword ptr [rip + 0x12a5a]
0x117e26:    lea     rdx, [rip + 0xa13c]
0x117e2d:    mov     esi, 1
0x117e32:    mov     r12d, r13d
0x117e35:    call    0x104e60
0x117e3e:    xor     r12d, r12d
0x117e41:    jmp     0x117de4
0x117dd5:    mov     rsi, qword ptr [rbx]
0x117dd8:    mov     rdi, r15
0x117ddb:    call    0x104b30
0x117e3a:    jmp     0x117de4
0x117de0:    test    eax, eax
0x117de2:    jne     0x117dc8
0x117dc8:    add     r12d, 1
0x117dcc:    add     rbx, 8
0x117dd0:    cmp     ebp, r12d
0x117dd3:    je      0x117e10
0x117dc4:    nop     dword ptr [rax]
0x117e06:    nop     word ptr cs:[rax + rax]
0x117e48:    nop     dword ptr [rax + rax]
0x117e50:    endbr64
0x117e54:    push    r12
0x117e56:    mov     ecx, 1
0x117e5b:    mov     edx, 1
0x117e60:    mov     r12, rdi
0x117e63:    push    rbp
0x117e64:    push    rbx

```

```

0x117e65:    mov     ebx, esi
0x117e67:    sub     rsp, 0x10
0x117e6b:    mov     rax, qword ptr fs:[0x28]
0x117e74:    mov     qword ptr [rsp + 8], rax
0x117e79:    xor     eax, eax
0x117e7b:    mov     rsi, rsp
0x117e7e:    call    0x117320
0x117e83:    mov     rbp, qword ptr [rsp]
0x117e87:    mov     eax, ebx
0x117e89:    test    rbp, rbp
0x117e8c:    je      0x117ea7
0x117ea7:    mov     rdx, qword ptr [rsp + 8]
0x117eac:    sub     rdx, qword ptr fs:[0x28]
0x117eb5:    jne     0x117f03
0x117e8e:    cmp     byte ptr [rbp], 0x4f
0x117e92:    je      0x117ec0
0x117eb7:    add     rsp, 0x10
0x117ebb:    pop     rbx
0x117ebc:    pop     rbp
0x117ebd:    pop     r12
0x117ebf:    ret
0x117f03:    call    0x1049d0
0x117ec0:    cmp     byte ptr [rbp + 1], 0x4e
0x117ec4:    jne     0x117e94
0x117e94:    lea     rsi, [rip + 0x8e4b]
0x117e9b:    mov     rdi, rbp
0x117e9e:    call    0x104b30
0x117ec6:    cmp     byte ptr [rbp + 2], 0
0x117eca:    mov     eax, 1
0x117ecf:    je      0x117ea7
0x117ea3:    test    eax, eax
0x117ea5:    jne     0x117ed8
0x117ed1:    jmp     0x117e94
0x117ed8:    mov     rcx, qword ptr [rip + 0x12a09]
0x117edf:    mov     r9, rbp
0x117ee2:    mov     r8, r12
0x117ee5:    xor     eax, eax
0x117ee7:    mov     rdi, qword ptr [rip + 0x12992]
0x117eee:    lea     rdx, [rip + 0xa074]
0x117ef5:    mov     esi, 1
0x117efa:    call    0x104e60
0x117eff:    mov     eax, ebx
0x117f01:    jmp     0x117ea7
0x117ed3:    nop     dword ptr [rax + rax]
0x117f08:    nop     dword ptr [rax + rax]
0x117f10:    endbr64
0x117f14:    push    r12
0x117f16:    mov     ecx, 1

```

```

0x117f1b:    mov     r12, rsi
0x117f1e:    mov     edx, 1
0x117f23:    sub     rsp, 0x10
0x117f27:    mov     rax, qword ptr fs:[0x28]
0x117f30:    mov     qword ptr [rsp + 8], rax
0x117f35:    xor     eax, eax
0x117f37:    mov     rsi, rsp
0x117f3a:    call    0x117320
0x117f3f:    mov     rdi, qword ptr [rsp]
0x117f43:    test    rdi, rdi
0x117f46:    je      0x117f50
0x117f50:    mov     rax, qword ptr [rsp + 8]
0x117f55:    sub     rax, qword ptr fs:[0x28]
0x117f5e:    jne     0x117f6a
0x117f48:    call    0x11d7a0
0x117f60:    add     rsp, 0x10
0x117f64:    mov     rax, r12
0x117f67:    pop     r12
0x117f69:    ret
0x117f6a:    call    0x1049d0
0x117f4d:    mov     r12, rax
0x117f50:    mov     rax, qword ptr [rsp + 8]
0x117f55:    sub     rax, qword ptr fs:[0x28]
0x117f5e:    jne     0x117f6a
0x117f6f:    nop
0x117f70:    endbr64
0x117f74:    push    r14
0x117f76:    push    r13
0x117f78:    push    r12
0x117f7a:    push    rbp
0x117f7b:    push    rbx
0x117f7c:    sub     rsp, 0x10
0x117f80:    mov     rax, qword ptr fs:[0x28]
0x117f89:    mov     qword ptr [rsp + 8], rax
0x117f8e:    xor     eax, eax
0x117f90:    test    edx, edx
0x117f92:    jle     0x117fe8
0x117fe8:    mov     rax, qword ptr [rsp + 8]
0x117fed:    sub     rax, qword ptr fs:[0x28]
0x117ff6:    jne     0x11802e
0x117f94:    lea     eax, [rdx - 1]
0x117f97:    mov     rbx, rsi
0x117f9a:    mov     rbp, rdi
0x117f9d:    mov     r12, rsp
0x117fa0:    lea     r13, [rsi + rax*8 + 8]
0x117fa5:    lea     r14, [rip + 0x9fbd]
0x117fac:    nop     dword ptr [rax]
0x117fb0:    mov     rdi, qword ptr [rbx]

```

```

0x117fb3:    mov     ecx, 1
0x117fb8:    mov     edx, 1
0x117fbd:    mov     rsi, r12
0x117fc0:    call    0x117320
0x117ff8:    add     rsp, 0x10
0x117ffc:    pop     rbx
0x117ffd:    pop     rbp
0x117ffe:    pop     r12
0x118000:    pop     r13
0x118002:    pop     r14
0x118004:    ret
0x11802e:    call    0x1049d0
0x117fc5:    mov     rdi, qword ptr [rsp]
0x117fc9:    test    rdi, rdi
0x117fcc:    je      0x117fdb
0x117fdb:    add     rbx, 8
0x117fdf:    add     rbp, 4
0x117fe3:    cmp     rbx, r13
0x117fe6:    jne     0x117fb0
0x117fce:    call    0x10bd00
0x117fb0:    mov     rdi, qword ptr [rbx]
0x117fb3:    mov     ecx, 1
0x117fb8:    mov     edx, 1
0x117fbd:    mov     rsi, r12
0x117fc0:    call    0x117320
0x117fd3:    cmp     eax, -1
0x117fd6:    je      0x118008
0x118008:    mov     r9, qword ptr [rsp]
0x11800c:    mov     r8, qword ptr [rbx]
0x11800f:    mov     rdx, r14
0x118012:    mov     esi, 1
0x118017:    mov     rcx, qword ptr [rip + 0x128ca]
0x11801e:    mov     rdi, qword ptr [rip + 0x1285b]
0x118025:    xor     eax, eax
0x118027:    call    0x104e60
0x117fd8:    mov     dword ptr [rbp], eax
0x117fdb:    add     rbx, 8
0x117fdf:    add     rbp, 4
0x117fe3:    cmp     rbx, r13
0x117fe6:    jne     0x117fb0
0x11802c:    jmp     0x117fdb
0x118005:    nop     dword ptr [rax]
0x118033:    nop     word ptr cs:[rax + rax]
0x11803e:    nop
0x118040:    endbr64
0x118044:    push    r12
0x118046:    cmp     byte ptr [rdi], 0x2f
0x118049:    mov     r12, rdi

```



```

0x11804c:    lea     rax, [rdi + 1]
0x118050:    je      0x118061
0x118061:    movzx   edx, byte ptr [rax]
0x118064:    mov     rsi, rax
0x118067:    test    dl, dl
0x118069:    jne     0x11808c
0x118052:    lea     rdi, [rip + 0x9f26]
0x118059:    call    0x108b00
0x11806b:    jmp     0x1180f0
0x11808c:    cmp     dl, 0x2f
0x11808f:    jne     0x118070
0x1180f0:    cmp     byte ptr [rsi - 1], 0x2f
0x1180f4:    jne     0x118109
0x118091:    movzx   ecx, byte ptr [rax + 1]
0x118095:    jmp     0x118086
0x118070:    cmp     dl, 0x2e
0x118073:    jne     0x118118
0x1180f6:    mov     rax, rsi
0x1180f9:    sub     rax, r12
0x1180fc:    cmp     rax, 1
0x118100:    setg    al
0x118103:    movzx   eax, al
0x118106:    sub     rsi, rax
0x118109:    mov     rax, r12
0x11810c:    mov     byte ptr [rsi], 0
0x11810f:    pop     r12
0x118111:    ret
0x118109:    mov     rax, r12
0x11810c:    mov     byte ptr [rsi], 0
0x11810f:    pop     r12
0x118111:    ret
0x118086:    add     rax, 1
0x11808a:    mov     edx, ecx
0x11808c:    cmp     dl, 0x2f
0x11808f:    jne     0x118070
0x118079:    movzx   ecx, byte ptr [rax + 1]
0x11807d:    cmp     cl, 0x2f
0x118080:    je      0x118086
0x118118:    test    dl, dl
0x11811a:    jne     0x1180c0
0x118082:    test    cl, cl
0x118084:    jne     0x1180a0
0x11811c:    jmp     0x1180f0
0x1180c0:    mov     rdi, rax
0x1180c3:    mov     rcx, rsi
0x1180c6:    add     rax, 1
0x1180ca:    add     rsi, 1
0x1180ce:    mov     byte ptr [rsi - 1], dl

```

```

0x1180d1:    movzx    edx, byte ptr [rax]
0x1180d4:    test     dl, dl
0x1180d6:    je       0x1180f0
0x1180a0:    cmp      cl, 0x2e
0x1180a3:    jne      0x1180c0
0x1180d8:    cmp      dl, 0x2f
0x1180db:    jne      0x1180c0
0x1180a5:    movzx    edx, byte ptr [rax + 2]
0x1180a9:    cmp      dl, 0x2f
0x1180ac:    je       0x118120
0x1180dd:    test     dl, dl
0x1180df:    je       0x1180f0
0x118120:    lea      rdi, [rsi - 1]
0x118124:    cmp      rdi, r12
0x118127:    jne      0x11812f
0x1180ae:    test     dl, dl
0x1180b0:    je       0x118120
0x1180e1:    mov      byte ptr [rsi], 0x2f
0x1180e4:    movzx    edx, byte ptr [rdi + 1]
0x1180e8:    lea      rsi, [rcx + 2]
0x1180ec:    test     dl, dl
0x1180ee:    jne      0x11808c
0x118129:    add      rax, 2
0x11812d:    jmp      0x1180ec
0x11812f:    cmp      byte ptr [rsi - 2], 0x2f
0x118133:    lea      rcx, [rsi - 2]
0x118137:    je       0x11814e
0x1180b2:    mov      edx, 0x2e
0x1180b7:    nop      word ptr [rax + rax]
0x1180c0:    mov      rdi, rax
0x1180c3:    mov      rcx, rsi
0x1180c6:    add      rax, 1
0x1180ca:    add      rsi, 1
0x1180ce:    mov      byte ptr [rsi - 1], dl
0x1180d1:    movzx    edx, byte ptr [rax]
0x1180d4:    test     dl, dl
0x1180d6:    je       0x1180f0
0x1180ec:    test     dl, dl
0x1180ee:    jne      0x11808c
0x11814e:    mov      rsi, rdi
0x118151:    jmp      0x118129
0x118139:    nop      dword ptr [rax]
0x118140:    mov      rsi, rcx
0x118143:    sub      rcx, 1
0x118147:    cmp      byte ptr [rcx], 0x2f
0x11814a:    jne      0x118140
0x11814c:    jmp      0x118129
0x11805e:    mov      rax, r12

```

```

0x118061:    movzx    edx, byte ptr [rax]
0x118064:    mov      rsi, rax
0x118067:    test     dl, dl
0x118069:    jne      0x11808c
0x118097:    nop      word ptr [rax + rax]
0x118112:    nop      word ptr [rax + rax]
0x11811e:    nop
0x118153:    nop      word ptr cs:[rax + rax]
0x11815e:    nop
0x118160:    endbr64
0x118164:    push     rbx
0x118165:    call     0x104a60
0x11816a:    mov      edi, eax
0x11816c:    mov      ebx, eax
0x11816e:    call     0x104970
0x118173:    test     rax, rax
0x118176:    je       0x118189
0x118189:    mov      rcx, qword ptr [rip + 0x12758]
0x118190:    mov      r8d, ebx
0x118193:    mov      rdi, qword ptr [rip + 0x126e6]
0x11819a:    lea      rdx, [rip + 0xa13f]
0x1181a1:    mov      esi, 1
0x1181a6:    call     0x104e60
0x118178:    mov      rdi, qword ptr [rax]
0x11817b:    call     0x11d7a0
0x1181ab:    mov      rcx, qword ptr [rip + 0x12736]
0x1181b2:    xor      eax, eax
0x1181b4:    mov      rdi, qword ptr [rip + 0x126c5]
0x1181bb:    lea      r8, [rip + 0x9dd7]
0x1181c2:    lea      rdx, [rip + 0xa14f]
0x1181c9:    mov      esi, 1
0x1181ce:    call     0x104e60
0x118180:    pop      rbx
0x118181:    mov      qword ptr [rip + 0x13910], rax
0x118188:    ret
0x1181d3:    mov      rdi, qword ptr [rip + 0x126a6]
0x1181da:    mov      esi, 1
0x1181df:    xor      eax, eax
0x1181e1:    mov      rcx, qword ptr [rip + 0x12700]
0x1181e8:    lea      rdx, [rip + 0xa161]
0x1181ef:    call     0x104e60
0x1181f4:    mov      edi, 1
0x1181f9:    call     0x104e20
0x1181fe:    nop
0x118200:    endbr64
0x118204:    push     r12
0x118206:    push     rbp
0x118207:    mov      rbp, rsi

```

```

0x11820a:    push    rbx
0x11820b:    movsxd  rbx, edx
0x11820e:    cmp     ebx, 2
0x118211:    je      0x118250
0x118250:    mov     ecx, 0x2020
0x118255:    mov     word ptr [rsi], cx
0x118258:    pop     rbx
0x118259:    pop     rbp
0x11825a:    pop     r12
0x11825c:    ret
0x118213:    mov     r12, rdi
0x118216:    jg      0x118230
0x118218:    test    ebx, ebx
0x11821a:    je      0x118224
0x118230:    cmp     ebx, 3
0x118233:    jne     0x118260
0x118224:    pop     rbx
0x118225:    pop     rbp
0x118226:    pop     r12
0x118228:    ret
0x11821c:    cmp     ebx, 1
0x11821f:    jne     0x118260
0x118235:    mov     edx, 0x2020
0x11823a:    mov     byte ptr [rsi + 2], 0x20
0x11823e:    mov     word ptr [rsi], dx
0x118241:    pop     rbx
0x118242:    pop     rbp
0x118243:    pop     r12
0x118245:    ret
0x118260:    mov     rdi, r12
0x118263:    call    0x1049b0
0x118221:    mov     byte ptr [rsi], 0x20
0x118224:    pop     rbx
0x118225:    pop     rbp
0x118226:    pop     r12
0x118228:    ret
0x118268:    movsxd  rdx, eax
0x11826b:    cmp     ebx, eax
0x11826d:    jge     0x1182a0
0x11826f:    mov     eax, 0x2e2e
0x118274:    lea     r8d, [rbx - 3]
0x118278:    mov     byte ptr [rbp + 2], 0x2e
0x11827c:    sub     rdx, rbx
0x11827f:    mov     word ptr [rbp], ax
0x118283:    movsxd  r8, r8d
0x118286:    lea     rsi, [r12 + rdx + 3]
0x11828b:    pop     rbx
0x11828c:    lea     rdi, [rbp + 3]

```

```

0x118290:    mov     rdx, r8
0x118293:    pop     rbp
0x118294:    pop     r12
0x118296:    jmp     0x104bb0
0x1182a0:    pop     rbx
0x1182a1:    mov     rsi, r12
0x1182a4:    mov     rdi, rbp
0x1182a7:    pop     rbp
0x1182a8:    pop     r12
0x1182aa:    jmp     0x104bb0
0x118229:    nop     dword ptr [rax]
0x118246:    nop     word ptr cs:[rax + rax]
0x11825d:    nop     dword ptr [rax]
0x11829b:    nop     dword ptr [rax + rax]
0x1182af:    nop
0x1182b0:    endbr64
0x1182b4:    push    r12
0x1182b6:    mov     r12, rsi
0x1182b9:    push    rbp
0x1182ba:    mov     rbp, rdi
0x1182bd:    push    rbx
0x1182be:    lea     ebx, [rdx - 1]
0x1182c1:    mov     edx, ebx
0x1182c3:    call    0x118200
0x1182c8:    mov     rdi, rbp
0x1182cb:    call    0x1049b0
0x1182d0:    movsxd  rcx, ebx
0x1182d3:    cmp     ebx, eax
0x1182d5:    movsxd  rdx, eax
0x1182d8:    mov     rax, rcx
0x1182db:    pop     rbx
0x1182dc:    cmovg   rax, rdx
0x1182e0:    pop     rbp
0x1182e1:    mov     byte ptr [r12 + rax], 0
0x1182e6:    mov     rax, r12
0x1182e9:    pop     r12
0x1182eb:    ret
0x1182ec:    nop     dword ptr [rax]
0x1182f0:    endbr64
0x1182f4:    push    r13
0x1182f6:    mov     edx, 1
0x1182fb:    xor     esi, esi
0x1182fd:    push    r12
0x1182ff:    push    rbp
0x118300:    mov     ebp, edi
0x118302:    call    0x104e50
0x118307:    mov     edi, ebp
0x118309:    mov     edx, 2

```

```

0x11830e:    xor     esi, esi
0x118310:    mov     r13, rax
0x118313:    call    0x104e50
0x118318:    mov     rsi, r13
0x11831b:    mov     edi, ebp
0x11831d:    xor     edx, edx
0x11831f:    mov     r12, rax
0x118322:    call    0x104e50
0x118327:    mov     rax, r12
0x11832a:    pop     rbp
0x11832b:    pop     r12
0x11832d:    pop     r13
0x11832f:    ret
0x118330:    endbr64
0x118334:    sub     rsp, 0x18
0x118338:    xor     edi, edi
0x11833a:    mov     rax, qword ptr fs:[0x28]
0x118343:    mov     qword ptr [rsp + 8], rax
0x118348:    xor     eax, eax
0x11834a:    call    0x104be0
0x11834f:    mov     rdi, rsp
0x118352:    mov     qword ptr [rsp], rax
0x118356:    call    0x104840
0x11835b:    mov     rdx, qword ptr [rsp + 8]
0x118360:    sub     rdx, qword ptr fs:[0x28]
0x118369:    jne     0x118370
0x11836b:    add     rsp, 0x18
0x11836f:    ret
0x118370:    call    0x1049d0
0x118375:    nop     word ptr cs:[rax + rax]
0x118380:    endbr64
0x118384:    push    r12
0x118386:    push    rbp
0x118387:    mov     rbp, rsi
0x11838a:    push    rbx
0x11838b:    mov     rbx, rdi
0x11838e:    call    0x1049b0
0x118393:    mov     rdi, rbp
0x118396:    mov     r12, rax
0x118399:    call    0x1049b0
0x11839e:    lea     rdi, [r12 + rax + 2]
0x1183a3:    call    0x11d670
0x1183a8:    mov     r9, rbp
0x1183ab:    mov     r8, rbx
0x1183ae:    mov     esi, 1
0x1183b3:    mov     r12, rax
0x1183b6:    mov     rdi, rax
0x1183b9:    xor     eax, eax

```

```

0x1183bb:    mov     rdx, -1
0x1183c2:    lea     rcx, [rip + 0x9be5]
0x1183c9:    call    0x104f10
0x1183ce:    mov     rdi, r12
0x1183d1:    call    0x104b20
0x1183d6:    mov     r12d, eax
0x1183d9:    cmp     eax, -1
0x1183dc:    je      0x1183f0
0x1183f0:    mov     rcx, qword ptr [rip + 0x124f1]
0x1183f7:    mov     rdi, qword ptr [rip + 0x12482]
0x1183fe:    mov     r8, rbx
0x118401:    xor     eax, eax
0x118403:    lea     rdx, [rip + 0x9f6e]
0x11840a:    mov     esi, 1
0x11840f:    call    0x104e60
0x1183de:    mov     eax, r12d
0x1183e1:    pop     rbx
0x1183e2:    pop     rbp
0x1183e3:    pop     r12
0x1183e5:    ret
0x118414:    mov     eax, r12d
0x118417:    pop     rbx
0x118418:    pop     rbp
0x118419:    pop     r12
0x11841b:    ret
0x1183e6:    nop     word ptr cs:[rax + rax]
0x11841c:    nop     dword ptr [rax]
0x118420:    endbr64
0x118424:    push    r13
0x118426:    push    r12
0x118428:    lea     r12, [rip + 0x9b85]
0x11842f:    push    rbp
0x118430:    mov     rdi, r12
0x118433:    push    rbx
0x118434:    sub     rsp, 8
0x118438:    call    0x1047e0
0x11843d:    mov     r13, qword ptr [rip + 0x1365c]
0x118444:    test    rax, rax
0x118447:    je      0x1184b0
0x1184b0:    add     rsp, 8
0x1184b4:    mov     rsi, r13
0x1184b7:    mov     rdi, r12
0x1184ba:    pop     rbx
0x1184bb:    pop     rbp
0x1184bc:    pop     r12
0x1184be:    pop     r13
0x1184c0:    jmp     0x118380
0x118449:    mov     rdi, rax

```

```

0x11844c:    mov     rbx, rax
0x11844f:    call   0x1049b0
0x118454:    mov     rdi, r13
0x118457:    mov     rbp, rax
0x11845a:    call   0x1049b0
0x11845f:    lea     rdi, [rbp + rax + 2]
0x118464:    call   0x11d670
0x118469:    mov     r9, qword ptr [rip + 0x13630]
0x118470:    mov     r8, rbx
0x118473:    lea     rcx, [rip + 0x9b3f]
0x11847a:    mov     rbp, rax
0x11847d:    mov     rdi, rax
0x118480:    mov     esi, 1
0x118485:    xor     eax, eax
0x118487:    mov     rdx, -1
0x11848e:    call   0x104f10
0x118493:    mov     rdi, r12
0x118496:    mov     rsi, rbp
0x118499:    call   0x118380
0x11849e:    add     rsp, 8
0x1184a2:    mov     rdi, rbp
0x1184a5:    pop     rbx
0x1184a6:    pop     rbp
0x1184a7:    pop     r12
0x1184a9:    pop     r13
0x1184ab:    jmp     0x118d10
0x1184c5:    nop     word ptr cs:[rax + rax]
0x1184d0:    endbr64
0x1184d4:    test    rsi, rsi
0x1184d7:    je      0x118520
0x118520:    ret
0x1184d9:    push    rbp
0x1184da:    mov     rbp, rsi
0x1184dd:    push    rbx
0x1184de:    mov     rbx, rdi
0x1184e1:    sub     rsp, 8
0x1184e5:    call   0x104f00
0x1184ea:    mov     rdi, rbx
0x1184ed:    lea     rsi, [rbx + rbp]
0x1184f1:    nop     dword ptr [rax]
0x1184f8:    movsx   rcx, byte ptr [rdi]
0x1184fc:    mov     rdx, qword ptr [rax]
0x1184ff:    test    byte ptr [rdx + rcx*2 + 1], 0x40
0x118504:    jne     0x118509
0x118506:    mov     byte ptr [rdi], 0x3f
0x118509:    add     rdi, 1
0x11850d:    cmp     rsi, rdi
0x118510:    jne     0x1184f8

```



```

0x118509:      add     rdi, 1
0x11850d:      cmp     rsi, rdi
0x118510:      jne     0x1184f8
0x118512:      add     rsp, 8
0x118516:      pop     rbx
0x118517:      pop     rbp
0x118518:      ret
0x1184f8:      movsx   rcx, byte ptr [rdi]
0x1184fc:      mov     rdx, qword ptr [rax]
0x1184ff:      test    byte ptr [rdx + rcx*2 + 1], 0x40
0x118504:      jne     0x118509
0x118519:      nop     dword ptr [rax]
0x118521:      nop     word ptr cs:[rax + rax]
0x11852c:      nop     dword ptr [rax]
0x118530:      endbr64
0x118534:      test    rsi, rsi
0x118537:      je      0x1185c2
0x1185c2:      xor     eax, eax
0x1185c4:      ret
0x11853d:      push    rbp
0x11853e:      mov     rbp, rsi
0x118541:      push    rbx
0x118542:      mov     rbx, rdi
0x118545:      sub     rsp, 8
0x118549:      call    0x104f00
0x11854e:      mov     rdi, rbx
0x118551:      lea     rsi, [rbx + rbp]
0x118555:      movabs  r9, 0x68000000c0000001
0x11855f:      movabs  r8, 0x2400f82000000000
0x118569:      mov     rcx, qword ptr [rax]
0x11856c:      nop     dword ptr [rax]
0x118570:      movsx   rdx, byte ptr [rdi]
0x118574:      mov     rax, rdx
0x118577:      test    byte ptr [rcx + rdx*2], 8
0x11857b:      jne     0x1185b0
0x11857d:      cmp     dl, 0x3d
0x118580:      jg      0x1185a0
0x1185b0:      add     rdi, 1
0x1185b4:      cmp     rsi, rdi
0x1185b7:      jne     0x118570
0x118582:      cmp     dl, 0x24
0x118585:      jle     0x11858d
0x1185a0:      sub     eax, 0x40
0x1185a3:      cmp     al, 0x3e
0x1185a5:      ja      0x11858d
0x1185b9:      add     rsp, 8
0x1185bd:      xor     eax, eax
0x1185bf:      pop     rbx

```

```

0x1185c0:    pop     rbp
0x1185c1:    ret
0x118570:    movsx   rdx, byte ptr [rdi]
0x118574:    mov     rax, rdx
0x118577:    test    byte ptr [rcx + rdx*2], 8
0x11857b:    jne     0x1185b0
0x11858d:    add     rsp, 8
0x118591:    mov     eax, 1
0x118596:    pop     rbx
0x118597:    pop     rbp
0x118598:    ret
0x118587:    bt      r8, rdx
0x11858b:    jb      0x1185b0
0x1185a7:    bt      r9, rax
0x1185ab:    jae     0x11858d
0x1185ad:    nop     dword ptr [rax]
0x118599:    nop     dword ptr [rax]
0x1185c5:    nop     word ptr cs:[rax + rax]
0x1185d0:    endbr64
0x1185d4:    push    rbx
0x1185d5:    mov     rbx, rdi
0x1185d8:    call    0x1049b0
0x1185dd:    sub     eax, 1
0x1185e0:    js      0x118618
0x118618:    xor     eax, eax
0x11861a:    pop     rbx
0x11861b:    ret
0x1185e2:    cdqe
0x1185e4:    jmp     0x118602
0x118602:    movzx   edx, byte ptr [rbx + rax]
0x118606:    cmp     dl, 0x26
0x118609:    jne     0x1185f0
0x11860b:    mov     eax, 1
0x118610:    pop     rbx
0x118611:    ret
0x1185f0:    cmp     dl, 0x20
0x1185f3:    je      0x1185fa
0x1185fa:    sub     rax, 1
0x1185fe:    test    eax, eax
0x118600:    js      0x118618
0x1185f5:    cmp     dl, 9
0x1185f8:    jne     0x118618
0x1185e6:    nop     word ptr cs:[rax + rax]
0x118612:    nop     word ptr [rax + rax]
0x11861c:    nop     dword ptr [rax]
0x118620:    endbr64
0x118624:    jmp     0x11863c
0x11863c:    movzx   eax, byte ptr [rdi]

```

```

0x11863f:    test    al, al
0x118641:    jne     0x118630
0x118643:    mov     eax, 1
0x118648:    ret
0x118630:    cmp     al, 0x20
0x118632:    je      0x118638
0x118638:    add     rdi, 1
0x11863c:    movzx   eax, byte ptr [rdi]
0x11863f:    test    al, al
0x118641:    jne     0x118630
0x118634:    cmp     al, 0x3b
0x118636:    jne     0x118650
0x118650:    xor     eax, eax
0x118652:    ret
0x118626:    nop     word ptr cs:[rax + rax]
0x118649:    nop     dword ptr [rax]
0x118653:    nop     word ptr cs:[rax + rax]
0x11865e:    nop
0x118660:    endbr64
0x118664:    push    r15
0x118666:    push    r14
0x118668:    push    r13
0x11866a:    push    r12
0x11866c:    push    rbp
0x11866d:    push    rbx
0x11866e:    sub     rsp, 0x98
0x118675:    mov     rax, qword ptr fs:[0x28]
0x11867e:    mov     qword ptr [rsp + 0x88], rax
0x118686:    xor     eax, eax
0x118688:    lea     r15, [rsp + 0x10]
0x11868d:    lea     r13, [rsp + 0x30]
0x118692:    mov     qword ptr [rsp + 8], 0
0x11869b:    mov     qword ptr [rsp], 0
0x1186a3:    jmp     0x1186d3
0x1186d3:    xor     ecx, ecx
0x1186d5:    mov     edx, 3
0x1186da:    mov     rsi, r15
0x1186dd:    mov     rdi, r13
0x1186e0:    call    0x117320
0x1186e5:    cmp     byte ptr [rsp + 0x30], 0
0x1186ea:    je      0x118758
0x118758:    mov     rdx, qword ptr [rsp + 8]
0x11875d:    test    rdx, rdx
0x118760:    je      0x11877d
0x1186ec:    mov     rdi, qword ptr [rsp + 0x10]
0x1186f1:    mov     r12d, 0xffffffff
0x1186f7:    test    rdi, rdi
0x1186fa:    je      0x118704

```

```

0x11877d:    mov     rax, qword ptr [rsp + 0x88]
0x118785:    sub     rax, qword ptr fs:[0x28]
0x11878e:    jne     0x1187b6
0x118762:    mov     rax, qword ptr [rip + 0x13327]
0x118769:    test    rax, rax
0x11876c:    je      0x1187a8
0x118704:    mov     rdi, qword ptr [rsp + 0x18]
0x118709:    mov     ebp, 0xffffffff
0x11870e:    test    rdi, rdi
0x118711:    je      0x11871a
0x1186fc:    call    0x10bd00
0x118790:    add     rsp, 0x98
0x118797:    pop     rbx
0x118798:    pop     rbp
0x118799:    pop     r12
0x11879b:    pop     r13
0x11879d:    pop     r14
0x11879f:    pop     r15
0x1187a1:    ret
0x1187b6:    call    0x1049d0
0x1187a8:    mov     rax, qword ptr [rsp + 8]
0x1187ad:    mov     qword ptr [rip + 0x132dc], rax
0x1187b4:    jmp     0x11877d
0x11876e:    mov     rcx, qword ptr [rsp]
0x118772:    mov     qword ptr [rip + 0x13317], rdx
0x118779:    mov     qword ptr [rcx + 0x18], rax
0x11877d:    mov     rax, qword ptr [rsp + 0x88]
0x118785:    sub     rax, qword ptr fs:[0x28]
0x11878e:    jne     0x1187b6
0x11871a:    mov     rdi, qword ptr [rsp + 0x20]
0x11871f:    mov     ebx, 0xffffffff
0x118724:    test    rdi, rdi
0x118727:    je      0x118730
0x118713:    call    0x10bd00
0x118701:    mov     r12d, eax
0x118704:    mov     rdi, qword ptr [rsp + 0x18]
0x118709:    mov     ebp, 0xffffffff
0x11870e:    test    rdi, rdi
0x118711:    je      0x11871a
0x118730:    mov     edi, 0x20
0x118735:    call    0x11d670
0x118729:    call    0x10bd00
0x118718:    mov     ebp, eax
0x11871a:    mov     rdi, qword ptr [rsp + 0x20]
0x11871f:    mov     ebx, 0xffffffff
0x118724:    test    rdi, rdi
0x118727:    je      0x118730
0x11873a:    cmp     qword ptr [rsp + 8], 0

```

```

0x118740:      mov     r14, rax
0x118743:      jne     0x1186a8
0x11872e:      mov     ebx, eax
0x118730:      mov     edi, 0x20
0x118735:      call    0x11d670
0x118749:      mov     qword ptr [rsp + 8], rax
0x11874e:      jmp     0x1186b0
0x1186a8:      mov     rax, qword ptr [rsp]
0x1186ac:      mov     qword ptr [rax + 0x18], r14
0x1186b0:      mov     rdi, r13
0x1186b3:      call    0x11d7a0
0x1186b0:      mov     rdi, r13
0x1186b3:      call    0x11d7a0
0x1186b8:      mov     dword ptr [r14 + 8], r12d
0x1186bc:      mov     qword ptr [r14], rax
0x1186bf:      mov     dword ptr [r14 + 0xc], ebp
0x1186c3:      mov     dword ptr [r14 + 0x10], ebx
0x1186c7:      mov     qword ptr [r14 + 0x18], 0
0x1186cf:      mov     qword ptr [rsp], r14
0x1186d3:      xor     ecx, ecx
0x1186d5:      mov     edx, 3
0x1186da:      mov     rsi, r15
0x1186dd:      mov     rdi, r13
0x1186e0:      call    0x117320
0x1186a5:      nop     dword ptr [rax]
0x118753:      nop     dword ptr [rax + rax]
0x1187a2:      nop     word ptr [rax + rax]
0x1187bb:      nop     dword ptr [rax + rax]
0x1187c0:      endbr64
0x1187c4:      sub     rsp, 8
0x1187c8:      lea     rsi, [rip - 0xf]
0x1187cf:      call    0x104b50
0x1187d4:      mov     dword ptr [rip + 0x132d2], 1
0x1187de:      add     rsp, 8
0x1187e2:      ret
0x1187e3:      nop     word ptr cs:[rax + rax]
0x1187ee:      nop
0x1187f0:      cmp     edi, 0x14
0x1187f3:      je      0x118850
0x118850:      lea     rsi, [rip + 0x89]
0x118857:      mov     edi, 0x14
0x11885c:      jmp     0x104b50
0x1187f5:      jg      0x118818
0x1187f7:      cmp     edi, 0xe
0x1187fa:      je      0x118838
0x118818:      cmp     edi, 0x1c
0x11881b:      jne     0x118830
0x118838:      lea     rsi, [rip + 0x31]

```

```

0x11883f:    mov     edi, 0xe
0x118844:    jmp     0x104b50
0x1187fc:    cmp     edi, 0x12
0x1187ff:    jne     0x118861
0x11881d:    lea     rsi, [rip + 0x1cc]
0x118824:    mov     edi, 0x1c
0x118829:    jmp     0x104b50
0x118830:    ret
0x118801:    lea     rsi, [rip + 0x148]
0x118808:    mov     edi, 0x12
0x11880d:    jmp     0x104b50
0x118861:    ret
0x118812:    nop     word ptr [rax + rax]
0x11882e:    nop
0x118831:    nop     dword ptr [rax]
0x118849:    nop     dword ptr [rax]
0x118862:    nop     word ptr cs:[rax + rax]
0x11886d:    nop     dword ptr [rax]
0x118870:    endbr64
0x118874:    mov     edx, dword ptr [rip + 0x1323a]
0x11887a:    push    rbp
0x11887b:    mov     eax, 1
0x118880:    mov     ebp, edi
0x118882:    test    edx, edx
0x118884:    jne     0x1188b0
0x118886:    mov     edi, ebp
0x118888:    mov     dword ptr [rip + 0x1322e], eax
0x11888e:    call    0x1187f0
0x1188b0:    xor     eax, eax
0x1188b2:    call    0x118330
0x118893:    xor     eax, eax
0x118895:    call    0x118330
0x1188b7:    mov     eax, dword ptr [rax]
0x1188b9:    test    eax, eax
0x1188bb:    je      0x1188d0
0x11889a:    mov     edi, 0x3c
0x11889f:    pop     rbp
0x1188a0:    sub     edi, dword ptr [rax]
0x1188a2:    jmp     0x104aa0
0x1188d0:    mov     edi, ebp
0x1188d2:    call    0x116fd0
0x1188bd:    xor     eax, eax
0x1188bf:    call    0x10b4e0
0x1188d7:    jmp     0x1188bd
0x1188c4:    xor     eax, eax
0x1188c6:    jmp     0x118886
0x1188a7:    nop     word ptr [rax + rax]
0x1188c8:    nop     dword ptr [rax + rax]

```

```

0x1188d9:      nop      dword ptr [rax]
0x1188e0:      endbr64
0x1188e4:      mov      eax, dword ptr [rip + 0x131ca]
0x1188ea:      push     rbp
0x1188eb:      mov      ebp, edi
0x1188ed:      test     eax, eax
0x1188ef:      jne      0x118908
0x1188f1:      mov      eax, 1
0x1188f6:      mov      edi, ebp
0x1188f8:      pop      rbp
0x1188f9:      mov      dword ptr [rip + 0x131c5], eax
0x1188ff:      jmp      0x1187f0
0x118908:      xor      eax, eax
0x11890a:      call     0x109c90
0x11890f:      cmp      eax, 1
0x118912:      sete     al
0x118915:      movzx    eax, al
0x118918:      mov      dword ptr [rip + 0x1319a], eax
0x11891e:      xor      eax, eax
0x118920:      call     0x109610
0x118925:      call     0x104920
0x11892a:      mov      esi, 0x13
0x11892f:      mov      edi, eax
0x118931:      call     0x104bd0
0x118936:      xor      eax, eax
0x118938:      mov      edi, ebp
0x11893a:      pop      rbp
0x11893b:      mov      dword ptr [rip + 0x13183], eax
0x118941:      jmp      0x1187f0
0x118904:      nop      dword ptr [rax]
0x118946:      nop      word ptr cs:[rax + rax]
0x118950:      endbr64
0x118954:      mov      eax, dword ptr [rip + 0x1315e]
0x11895a:      push     rbx
0x11895b:      mov      ebx, edi
0x11895d:      test     eax, eax
0x11895f:      jne      0x1189a0
0x118961:      cmp      ebx, 0x14
0x118964:      je       0x1189b4
0x1189a0:      call     0x107bd0
0x1189b4:      lea      rsi, [rip - 0xdb]
0x1189bb:      mov      edi, 0x14
0x1189c0:      pop      rbx
0x1189c1:      jmp      0x104b50
0x118966:      jg       0x118988
0x1189a5:      mov      dword ptr [rip + 0x13111], 0
0x1189af:      cmp      ebx, 0x14
0x1189b2:      jne      0x118966

```

```

0x118968:      cmp     ebx, 0xe
0x11896b:      je      0x1189d8
0x118988:      cmp     ebx, 0x1c
0x11898b:      jne     0x1189d0
0x1189d8:      lea     rsi, [rip - 0x16f]
0x1189df:      mov     edi, 0xe
0x1189e4:      pop     rbx
0x1189e5:      jmp     0x104b50
0x11896d:      cmp     ebx, 0x12
0x118970:      jne     0x1189d0
0x11898d:      lea     rsi, [rip + 0x5c]
0x118994:      mov     edi, 0x1c
0x118999:      pop     rbx
0x11899a:      jmp     0x104b50
0x1189d0:      pop     rbx
0x1189d1:      ret
0x118972:      lea     rsi, [rip - 0x29]
0x118979:      mov     edi, 0x12
0x11897e:      pop     rbx
0x11897f:      jmp     0x104b50
0x118984:      nop     dword ptr [rax]
0x11899f:      nop
0x1189c6:      nop     word ptr cs:[rax + rax]
0x1189d2:      nop     word ptr [rax + rax]
0x1189ea:      nop     word ptr [rax + rax]
0x1189f0:      endbr64
0x1189f4:      mov     eax, dword ptr [rip + 0x130ba]
0x1189fa:      push    rbx
0x1189fb:      mov     ebx, edi
0x1189fd:      test    eax, eax
0x1189ff:      jne     0x118a50
0x118a01:      mov     eax, 1
0x118a06:      mov     dword ptr [rip + 0x130b4], eax
0x118a0c:      cmp     ebx, 0x14
0x118a0f:      je      0x118a72
0x118a50:      xor     eax, eax
0x118a52:      call    0x10a6c0
0x118a72:      lea     rsi, [rip - 0x199]
0x118a79:      mov     edi, 0x14
0x118a7e:      pop     rbx
0x118a7f:      jmp     0x104b50
0x118a11:      jg      0x118a30
0x118a57:      xor     eax, eax
0x118a59:      call    0x109d80
0x118a13:      cmp     ebx, 0xe
0x118a16:      je      0x118a90
0x118a30:      cmp     ebx, 0x1c
0x118a33:      jne     0x118a88

```



```

0x118a5e:    mov     edi, ebx
0x118a60:    call    0x107bd0
0x118a90:    lea     rsi, [rip - 0x227]
0x118a97:    mov     edi, 0xe
0x118a9c:    pop     rbx
0x118a9d:    jmp     0x104b50
0x118a18:    cmp     ebx, 0x12
0x118a1b:    jne     0x118a88
0x118a35:    lea     rsi, [rip - 0x4c]
0x118a3c:    mov     edi, 0x1c
0x118a41:    pop     rbx
0x118a42:    jmp     0x104b50
0x118a88:    pop     rbx
0x118a89:    ret
0x118a65:    xor     eax, eax
0x118a67:    mov     dword ptr [rip + 0x13053], eax
0x118a6d:    cmp     ebx, 0x14
0x118a70:    jne     0x118a11
0x118a1d:    lea     rsi, [rip - 0xd4]
0x118a24:    mov     edi, 0x12
0x118a29:    pop     rbx
0x118a2a:    jmp     0x104b50
0x118a2f:    nop
0x118a47:    nop     word ptr [rax + rax]
0x118a84:    nop     dword ptr [rax]
0x118a8a:    nop     word ptr [rax + rax]
0x118aa2:    nop     word ptr cs:[rax + rax]
0x118aad:    nop     dword ptr [rax]
0x118ab0:    endbr64
0x118ab4:    sub     rsp, 8
0x118ab8:    mov     ecx, dword ptr [rip + 0x13006]
0x118abe:    test    ecx, ecx
0x118ac0:    jne     0x118af0
0x118ac2:    mov     edx, dword ptr [rip + 0x12ff8]
0x118ac8:    test    edx, edx
0x118aca:    jne     0x118b30
0x118af0:    xor     eax, eax
0x118af2:    call    0x109610
0x118acc:    mov     eax, dword ptr [rip + 0x12fea]
0x118ad2:    test    eax, eax
0x118ad4:    jne     0x118ae0
0x118b30:    xor     eax, eax
0x118b32:    call    0x10a6c0
0x118af7:    call    0x104920
0x118ad6:    add     rsp, 8
0x118ada:    ret
0x118ae0:    mov     dword ptr [rip + 0x12fd2], 0
0x118aea:    add     rsp, 8

```

```

0x118aee:    ret
0x118b37:    xor     eax, eax
0x118b39:    call   0x109d80
0x118afc:    mov     esi, 0x13
0x118b01:    mov     edi, eax
0x118b03:    call   0x104bd0
0x118b3e:    mov     edi, 0x1c
0x118b43:    call   0x107bd0
0x118b08:    mov     dword ptr [rip + 0x12fb2], 0
0x118b12:    mov     dword ptr [rip + 0x12fa4], 0
0x118b1c:    mov     dword ptr [rip + 0x12f96], 0
0x118b26:    add     rsp, 8
0x118b2a:    ret
0x118b48:    mov     dword ptr [rip + 0x12f6e], 0
0x118b52:    mov     dword ptr [rip + 0x12f60], 0
0x118b5c:    add     rsp, 8
0x118b60:    ret
0x118adb:    nop     dword ptr [rax + rax]
0x118aef:    nop
0x118b2b:    nop     dword ptr [rax + rax]
0x118b61:    nop     word ptr cs:[rax + rax]
0x118b6c:    nop     dword ptr [rax]
0x118b70:    endbr64
0x118b74:    mov     dword ptr [rip + 0x12f3a], edi
0x118b7a:    test    edi, edi
0x118b7c:    jne     0x118b80
0x118b7e:    ret
0x118b80:    xor     eax, eax
0x118b82:    jmp     0x118ab0
0x118b7f:    nop
0x118b87:    nop     word ptr [rax + rax]
0x118b90:    endbr64
0x118b94:    sub     rsp, 8
0x118b98:    mov     eax, dword ptr [rip + 0x11bfa]
0x118b9e:    cmp     edi, 1
0x118ba1:    je      0x118bf0
0x118bf0:    test    eax, eax
0x118bf2:    jne     0x118c20
0x118ba3:    test    eax, eax
0x118ba5:    jne     0x118bd0
0x118bf4:    lea     rsi, [rip - 0x20b]
0x118bfb:    mov     edi, 0x1c
0x118c00:    call   0x104b50
0x118c20:    lea     rsi, [rip - 0x347]
0x118c27:    mov     edi, 0x14
0x118c2c:    call   0x104b50
0x118ba7:    mov     esi, 1
0x118bac:    mov     edi, 0x1c

```

```

0x118bb1:    call    0x104b50
0x118bd0:    mov     esi, 1
0x118bd5:    mov     edi, 0x14
0x118bda:    call    0x104b50
0x118c05:    lea     rsi, [rip - 0x39c]
0x118c0c:    mov     edi, 0xe
0x118c11:    add     rsp, 8
0x118c15:    jmp     0x104b50
0x118c31:    lea     rsi, [rip - 0x2e8]
0x118c38:    mov     edi, 0x12
0x118c3d:    call    0x104b50
0x118bb6:    mov     esi, 1
0x118bbb:    mov     edi, 0xe
0x118bc0:    add     rsp, 8
0x118bc4:    jmp     0x104b50
0x118bdf:    mov     esi, 1
0x118be4:    mov     edi, 0x12
0x118be9:    call    0x104b50
0x118c42:    jmp     0x118bf4
0x118bee:    jmp     0x118ba7
0x118bc9:    nop     dword ptr [rax]
0x118c1a:    nop     word ptr [rax + rax]
0x118c44:    nop     word ptr cs:[rax + rax]
0x118c4f:    nop
0x118c50:    endbr64
0x118c54:    push    rbp
0x118c55:    mov     esi, 1
0x118c5a:    mov     edi, 0x14
0x118c5f:    call    0x104b50
0x118c64:    mov     esi, 1
0x118c69:    mov     edi, 0x12
0x118c6e:    cmp     rax, 1
0x118c72:    setne   al
0x118c75:    movzx   eax, al
0x118c78:    mov     dword ptr [rip + 0x11b1a], eax
0x118c7e:    call    0x104b50
0x118c83:    mov     esi, 1
0x118c88:    mov     edi, 0x1c
0x118c8d:    call    0x104b50
0x118c92:    lea     rbp, [rip - 0x1519]
0x118c99:    mov     edi, 0xb
0x118c9e:    mov     rsi, rbp
0x118ca1:    call    0x104b50
0x118ca6:    mov     rsi, rbp
0x118ca9:    mov     edi, 1
0x118cae:    call    0x104b50
0x118cb3:    mov     rsi, rbp
0x118cb6:    mov     edi, 0xf

```

```

0x118cbb:    call    0x104b50
0x118cc0:    mov     esi, 1
0x118cc5:    mov     edi, 3
0x118cca:    call    0x104b50
0x118ccf:    lea     rsi, [rip - 0x516]
0x118cd6:    mov     edi, 2
0x118cdb:    pop     rbp
0x118cdc:    jmp     0x104b50
0x118ce1:    nop     word ptr cs:[rax + rax]
0x118ceb:    nop     dword ptr [rax + rax]
0x118cf0:    endbr64
0x118cf4:    push    rax
0x118cf5:    pop     rax
0x118cf6:    lea     rdi, [rip + 0x96a8]
0x118cfd:    sub     rsp, 8
0x118d01:    call    0x108b00
0x118d06:    call    0x104850
0x118d0b:    nop     dword ptr [rax + rax]
0x118d10:    endbr64
0x118d14:    test    rdi, rdi
0x118d17:    je      0x118d20
0x118d20:    lea     rdi, [rip + 0x9697]
0x118d27:    jmp     0x108b00
0x118d19:    jmp     0x104800
0x118d1e:    nop
0x118d2c:    nop     dword ptr [rax]
0x118d30:    endbr64
0x118d34:    xor     r8d, r8d
0x118d37:    test    rdx, rdx
0x118d3a:    je      0x118d90
0x118d90:    mov     eax, r8d
0x118d93:    ret
0x118d3c:    push    r14
0x118d3e:    push    r13
0x118d40:    push    r12
0x118d42:    mov     r12, rsi
0x118d45:    push    rbp
0x118d46:    mov     ebp, edi
0x118d48:    push    rbx
0x118d49:    mov     rbx, rdx
0x118d4c:    call    0x104860
0x118d51:    mov     r13d, dword ptr [rax]
0x118d54:    mov     r14, rax
0x118d57:    jmp     0x118d66
0x118d66:    mov     dword ptr [r14], r13d
0x118d69:    mov     rdx, rbx
0x118d6c:    mov     rsi, r12
0x118d6f:    mov     edi, ebp

```

```

0x118d71:    call    0x1047a0
0x118d76:    mov     r8d, eax
0x118d79:    test    eax, eax
0x118d7b:    js      0x118d60
0x118d60:    cmp     dword ptr [r14], 4
0x118d64:    jne     0x118d7d
0x118d7d:    pop     rbx
0x118d7e:    mov     eax, r8d
0x118d81:    pop     rbp
0x118d82:    pop     r12
0x118d84:    pop     r13
0x118d86:    pop     r14
0x118d88:    ret
0x118d59:    nop     dword ptr [rax]
0x118d89:    nop     dword ptr [rax]
0x118d94:    nop     word ptr cs:[rax + rax]
0x118d9f:    nop
0x118da0:    endbr64
0x118da4:    xor     r8d, r8d
0x118da7:    test    rdx, rdx
0x118daa:    je      0x118e00
0x118e00:    mov     eax, r8d
0x118e03:    ret
0x118dac:    push    r14
0x118dae:    push    r13
0x118db0:    push    r12
0x118db2:    mov     r12, rsi
0x118db5:    push    rbp
0x118db6:    mov     ebp, edi
0x118db8:    push    rbx
0x118db9:    mov     rbx, rdx
0x118dbc:    call    0x104860
0x118dc1:    mov     r13d, dword ptr [rax]
0x118dc4:    mov     r14, rax
0x118dc7:    jmp     0x118dd6
0x118dd6:    mov     dword ptr [r14], r13d
0x118dd9:    mov     rdx, rbx
0x118ddc:    mov     rsi, r12
0x118ddf:    mov     edi, ebp
0x118de1:    call    0x104790
0x118de6:    mov     r8d, eax
0x118de9:    test    eax, eax
0x118deb:    js      0x118dd0
0x118dd0:    cmp     dword ptr [r14], 4
0x118dd4:    jne     0x118ded
0x118ded:    pop     rbx
0x118dee:    mov     eax, r8d
0x118df1:    pop     rbp

```

```

0x118df2:    pop     r12
0x118df4:    pop     r13
0x118df6:    pop     r14
0x118df8:    ret
0x118dc9:    nop     dword ptr [rax]
0x118df9:    nop     dword ptr [rax]
0x118e04:    nop     word ptr cs:[rax + rax]
0x118e0f:    nop
0x118e10:    endbr64
0x118e14:    xor     r8d, r8d
0x118e17:    test    rdx, rdx
0x118e1a:    je      0x118e70
0x118e70:    mov     eax, r8d
0x118e73:    ret
0x118e1c:    push    r14
0x118e1e:    push    r13
0x118e20:    push    r12
0x118e22:    mov     r12, rsi
0x118e25:    push    rbp
0x118e26:    mov     rbp, rdi
0x118e29:    push    rbx
0x118e2a:    mov     rbx, rdx
0x118e2d:    call    0x104860
0x118e32:    mov     r13d, dword ptr [rax]
0x118e35:    mov     r14, rax
0x118e38:    jmp     0x118e46
0x118e46:    mov     dword ptr [r14], r13d
0x118e49:    mov     rdx, rbx
0x118e4c:    mov     rsi, r12
0x118e4f:    mov     rdi, rbp
0x118e52:    call    0x104910
0x118e57:    mov     r8d, eax
0x118e5a:    test    eax, eax
0x118e5c:    js      0x118e40
0x118e40:    cmp     dword ptr [r14], 4
0x118e44:    jne     0x118e5e
0x118e5e:    pop     rbx
0x118e5f:    mov     eax, r8d
0x118e62:    pop     rbp
0x118e63:    pop     r12
0x118e65:    pop     r13
0x118e67:    pop     r14
0x118e69:    ret
0x118e3a:    nop     word ptr [rax + rax]
0x118e6a:    nop     word ptr [rax + rax]
0x118e74:    nop     word ptr cs:[rax + rax]
0x118e7f:    nop
0x118e80:    endbr64

```

```

0x118e84:    push    r13
0x118e86:    mov     r13, rdi
0x118e89:    push    r12
0x118e8b:    push    rbp
0x118e8c:    push    rbx
0x118e8d:    mov     ebx, 0x64
0x118e92:    sub     rsp, 0x18
0x118e96:    jmp     0x118eaa
0x118eaa:    movsxd  r12, ebx
0x118ead:    mov     rdi, r12
0x118eb0:    call    0x11d670
0x118eb5:    mov     rdx, r12
0x118eb8:    mov     rdi, r13
0x118ebb:    mov     rbp, rax
0x118ebe:    mov     rsi, rax
0x118ec1:    xor     eax, eax
0x118ec3:    call    0x118e10
0x118ec8:    cmp     ebx, eax
0x118eca:    jle     0x118ea0
0x118ea0:    mov     rdi, rbp
0x118ea3:    add     ebx, ebx
0x118ea5:    call    0x118d10
0x118ecc:    mov     rdi, rbp
0x118ecf:    mov     dword ptr [rsp + 0xc], eax
0x118ed3:    call    0x118d10
0x118ed8:    mov     eax, dword ptr [rsp + 0xc]
0x118edc:    add     rsp, 0x18
0x118ee0:    pop     rbx
0x118ee1:    pop     rbp
0x118ee2:    pop     r12
0x118ee4:    pop     r13
0x118ee6:    ret
0x118e98:    nop     dword ptr [rax + rax]
0x118ee7:    nop     word ptr [rax + rax]
0x118ef0:    endbr64
0x118ef4:    push    r13
0x118ef6:    push    r12
0x118ef8:    mov     r12d, edi
0x118efb:    push    rbp
0x118efc:    mov     rbp, rsi
0x118eff:    push    rbx
0x118f00:    sub     rsp, 8
0x118f04:    call    0x104860
0x118f09:    mov     r13d, dword ptr [rax]
0x118f0c:    mov     rbx, rax
0x118f0f:    jmp     0x118f1d
0x118f1d:    mov     dword ptr [rbx], r13d
0x118f20:    mov     rsi, rbp

```

```

0x118f23:    mov     edi, r12d
0x118f26:    call    0x104eb0
0x118f2b:    test    eax, eax
0x118f2d:    js       0x118f18
0x118f18:    cmp     dword ptr [rbx], 4
0x118f1b:    jne     0x118f2f
0x118f2f:    add     rsp, 8
0x118f33:    pop     rbx
0x118f34:    pop     rbp
0x118f35:    pop     r12
0x118f37:    pop     r13
0x118f39:    ret
0x118f11:    nop     dword ptr [rax]
0x118f3a:    nop     word ptr [rax + rax]
0x118f40:    endbr64
0x118f44:    push    r13
0x118f46:    push    r12
0x118f48:    mov     r12, rdi
0x118f4b:    push    rbp
0x118f4c:    mov     rbp, rsi
0x118f4f:    push    rbx
0x118f50:    sub     rsp, 8
0x118f54:    call    0x104860
0x118f59:    mov     r13d, dword ptr [rax]
0x118f5c:    mov     rbx, rax
0x118f5f:    jmp     0x118f6d
0x118f6d:    mov     dword ptr [rbx], r13d
0x118f70:    mov     rsi, rbp
0x118f73:    mov     rdi, r12
0x118f76:    call    0x104b90
0x118f7b:    test    eax, eax
0x118f7d:    js       0x118f68
0x118f68:    cmp     dword ptr [rbx], 4
0x118f6b:    jne     0x118f7f
0x118f7f:    add     rsp, 8
0x118f83:    pop     rbx
0x118f84:    pop     rbp
0x118f85:    pop     r12
0x118f87:    pop     r13
0x118f89:    ret
0x118f61:    nop     dword ptr [rax]
0x118f8a:    nop     word ptr [rax + rax]
0x118f90:    endbr64
0x118f94:    push    r13
0x118f96:    push    r12
0x118f98:    mov     r12, rdi
0x118f9b:    push    rbp
0x118f9c:    mov     rbp, rsi

```



```

0x118f9f:    push    rbx
0x118fa0:    sub     rsp, 8
0x118fa4:    call    0x104860
0x118fa9:    mov     r13d, dword ptr [rax]
0x118fac:    mov     rbx, rax
0x118faf:    jmp     0x118fbd
0x118fbd:    mov     dword ptr [rbx], r13d
0x118fc0:    mov     rsi, rbp
0x118fc3:    mov     rdi, r12
0x118fc6:    call    0x104ad0
0x118fcb:    test    eax, eax
0x118fcd:    js      0x118fb8
0x118fb8:    cmp     dword ptr [rbx], 4
0x118fbb:    jne     0x118fcf
0x118fcf:    add     rsp, 8
0x118fd3:    pop     rbx
0x118fd4:    pop     rbp
0x118fd5:    pop     r12
0x118fd7:    pop     r13
0x118fd9:    ret
0x118fb1:    nop     dword ptr [rax]
0x118fda:    nop     word ptr [rax + rax]
0x118fe0:    endbr64
0x118fe4:    push    r12
0x118fe6:    push    rbp
0x118fe7:    push    rbx
0x118fe8:    mov     ebx, 0x40
0x118fed:    call    0x104860
0x118ff2:    mov     edi, 0x40
0x118ff7:    mov     dword ptr [rax], 0
0x118ffd:    mov     r12, rax
0x119000:    call    0x11d670
0x119005:    mov     rbp, rax
0x119008:    jmp     0x119031
0x119031:    mov     rsi, rbx
0x119034:    mov     rdi, rbp
0x119037:    call    0x104a80
0x11903c:    test    rax, rax
0x11903f:    je      0x119010
0x119010:    cmp     dword ptr [r12], 0x22
0x119015:    jne     0x119050
0x119041:    mov     rax, rbp
0x119044:    pop     rbx
0x119045:    pop     rbp
0x119046:    pop     r12
0x119048:    ret
0x119017:    add     rbx, 0x40
0x11901b:    mov     rdi, rbp

```

```

0x11901e:    mov     rsi, rbx
0x119021:    call   0x11d690
0x119050:    mov     rdi, rbp
0x119053:    xor     ebp, ebp
0x119055:    call   0x118d10
0x119026:    mov     dword ptr [r12], 0
0x11902e:    mov     rbp, rax
0x119031:    mov     rsi, rbx
0x119034:    mov     rdi, rbp
0x119037:    call   0x104a80
0x11905a:    mov     rax, rbp
0x11905d:    pop     rbx
0x11905e:    pop     rbp
0x11905f:    pop     r12
0x119061:    ret
0x11900a:    nop     word ptr [rax + rax]
0x119049:    nop     dword ptr [rax]
0x119062:    nop     word ptr cs:[rax + rax]
0x11906d:    nop     dword ptr [rax]
0x119070:    endbr64
0x119074:    push    rbp
0x119075:    mov     esi, 0x2f
0x11907a:    mov     rbp, rdi
0x11907d:    push    rbx
0x11907e:    sub     rsp, 8
0x119082:    call   0x104a20
0x119087:    test    rax, rax
0x11908a:    je      0x1190a8
0x1190a8:    add     rsp, 8
0x1190ac:    lea     rdi, [rip + 0x862b]
0x1190b3:    pop     rbx
0x1190b4:    pop     rbp
0x1190b5:    jmp     0x11d7a0
0x11908c:    mov     rbx, rax
0x11908f:    mov     rdi, rbp
0x119092:    call   0x11d7a0
0x119097:    sub     rbx, rbp
0x11909a:    mov     byte ptr [rax + rbx], 0x30
0x11909e:    add     rsp, 8
0x1190a2:    pop     rbx
0x1190a3:    pop     rbp
0x1190a4:    ret
0x1190a5:    nop     dword ptr [rax]
0x1190ba:    nop     word ptr [rax + rax]
0x1190c0:    endbr64
0x1190c4:    push    rbx
0x1190c5:    mov     rbx, rdi
0x1190c8:    call   0x1049b0

```

```

0x1190cd:    lea     rax, [rbx + rax - 1]
0x1190d2:    cmp     byte ptr [rax], 0x2f
0x1190d5:    jne     0x1190da
0x1190d7:    mov     byte ptr [rax], 0
0x1190da:    mov     rdi, rbx
0x1190dd:    mov     esi, 0x2f
0x1190e2:    call    0x104a20
0x1190da:    mov     rdi, rbx
0x1190dd:    mov     esi, 0x2f
0x1190e2:    call    0x104a20
0x1190e7:    lea     rdx, [rax + 1]
0x1190eb:    test    rax, rax
0x1190ee:    mov     rax, rbx
0x1190f1:    pop     rbx
0x1190f2:    cmovne  rax, rdx
0x1190f6:    ret
0x1190f7:    nop     word ptr [rax + rax]
0x119100:    endbr64
0x119104:    push    rbx
0x119105:    mov     esi, 0x10
0x11910a:    mov     ebx, edi
0x11910c:    mov     edi, 1
0x119111:    call    0x11d740
0x119116:    mov     dword ptr [rax + 8], ebx
0x119119:    pop     rbx
0x11911a:    ret
0x11911b:    nop     dword ptr [rax + rax]
0x119120:    endbr64
0x119124:    test    rdi, rdi
0x119127:    je      0x119148
0x119148:    ret
0x119129:    push    rbp
0x11912a:    mov     rbp, rdi
0x11912d:    mov     rdi, qword ptr [rdi]
0x119130:    test    rdi, rdi
0x119133:    je      0x11913a
0x11913a:    mov     rdi, rbp
0x11913d:    pop     rbp
0x11913e:    jmp     0x118d10
0x119135:    call    0x118d10
0x119143:    nop     dword ptr [rax + rax]
0x119149:    nop     dword ptr [rax]
0x119150:    endbr64
0x119154:    push    rbp
0x119155:    mov     rbp, rsi
0x119158:    push    rbx
0x119159:    mov     rbx, rdi
0x11915c:    sub     rsp, 8

```

```

0x119160:    mov     eax, dword ptr [rdi + 0xc]
0x119163:    lea     esi, [rax + 1]
0x119166:    mov     dword ptr [rdi + 0xc], esi
0x119169:    imul    esi, dword ptr [rdi + 8]
0x11916d:    mov     rdi, qword ptr [rdi]
0x119170:    movsxd  rsi, esi
0x119173:    call    0x11d690
0x119178:    mov     ecx, dword ptr [rbx + 0xc]
0x11917b:    movsxd  rdx, dword ptr [rbx + 8]
0x11917f:    mov     rsi, rbp
0x119182:    mov     qword ptr [rbx], rax
0x119185:    add     rsp, 8
0x119189:    sub     ecx, 1
0x11918c:    pop     rbx
0x11918d:    pop     rbp
0x11918e:    imul    ecx, edx
0x119191:    movsxd  rcx, ecx
0x119194:    lea     rdi, [rax + rcx]
0x119198:    jmp     0x104bb0
0x11919d:    nop     dword ptr [rax]
0x1191a0:    endbr64
0x1191a4:    mov     eax, dword ptr [rdi + 0xc]
0x1191a7:    test    eax, eax
0x1191a9:    je      0x1191f8
0x1191f8:    xor     eax, eax
0x1191fa:    ret
0x1191ab:    push    rbp
0x1191ac:    sub     eax, 1
0x1191af:    mov     rbp, rsi
0x1191b2:    push    rbx
0x1191b3:    mov     rbx, rdi
0x1191b6:    sub     rsp, 8
0x1191ba:    movsxd  rdx, dword ptr [rdi + 8]
0x1191be:    mov     dword ptr [rdi + 0xc], eax
0x1191c1:    mov     rsi, rdx
0x1191c4:    imul    esi, eax
0x1191c7:    movsxd  rsi, esi
0x1191ca:    add     rsi, qword ptr [rdi]
0x1191cd:    mov     rdi, rbp
0x1191d0:    call    0x104bb0
0x1191d5:    mov     esi, dword ptr [rbx + 0xc]
0x1191d8:    imul    esi, dword ptr [rbx + 8]
0x1191dc:    mov     rdi, qword ptr [rbx]
0x1191df:    movsxd  rsi, esi
0x1191e2:    call    0x11d690
0x1191e7:    mov     qword ptr [rbx], rax
0x1191ea:    add     rsp, 8
0x1191ee:    mov     rax, rbp

```

```

0x1191f1:    pop     rbx
0x1191f2:    pop     rbp
0x1191f3:    ret
0x1191f4:    nop     dword ptr [rax]
0x1191fb:    nop     dword ptr [rax + rax]
0x119200:    endbr64
0x119204:    mov     ecx, dword ptr [rdi + 0xc]
0x119207:    cmp     ecx, edx
0x119209:    jl      0x119240
0x119240:    xor     eax, eax
0x119242:    ret
0x11920b:    test    ecx, ecx
0x11920d:    je      0x119240
0x11920f:    sub     rsp, 8
0x119213:    movsxd  r9, dword ptr [rdi + 8]
0x119217:    sub     ecx, edx
0x119219:    mov     r8, rsi
0x11921c:    imul    ecx, r9d
0x119220:    mov     rdx, r9
0x119223:    movsxd  rsi, ecx
0x119226:    add     rsi, qword ptr [rdi]
0x119229:    mov     rdi, r8
0x11922c:    call    0x104bb0
0x119231:    add     rsp, 8
0x119235:    ret
0x119236:    nop     word ptr cs:[rax + rax]
0x119243:    nop     word ptr cs:[rax + rax]
0x11924e:    nop
0x119250:    endbr64
0x119254:    push    rbx
0x119255:    mov     rbx, rdi
0x119258:    mov     dword ptr [rdi + 0xc], esi
0x11925b:    imul    esi, dword ptr [rdi + 8]
0x11925f:    mov     rdi, qword ptr [rdi]
0x119262:    movsxd  rsi, esi
0x119265:    call    0x11d690
0x11926a:    mov     qword ptr [rbx], rax
0x11926d:    pop     rbx
0x11926e:    ret
0x11926f:    nop
0x119270:    endbr64
0x119274:    mov     eax, dword ptr [rdi + 0xc]
0x119277:    ret
0x119278:    nop     dword ptr [rax + rax]
0x119280:    push    r12
0x119282:    mov     r12, rdi
0x119285:    mov     edi, 2
0x11928a:    push    rbp

```

```

0x11928b:    push    rbx
0x11928c:    mov     ebx, esi
0x11928e:    xor     esi, esi
0x119290:    call    0x104b50
0x119295:    xor     esi, esi
0x119297:    mov     edi, 3
0x11929c:    call    0x104b50
0x1192a1:    test    ebx, ebx
0x1192a3:    jne     0x1192e0
0x1192a5:    mov     r9, qword ptr [rip + 0x115b4]
0x1192ac:    xor     r8d, r8d
0x1192af:    mov     rcx, r12
0x1192b2:    xor     eax, eax
0x1192b4:    lea     rdi, [rip + 0x7363]
0x1192bb:    lea     rdx, [rip + 0x911e]
0x1192c2:    lea     rsi, [rip + 0x735a]
0x1192c9:    call    0x104e30
0x1192e0:    mov     edi, 1
0x1192e5:    lea     rbp, [rip + 0x90ed]
0x1192ec:    call    0x104ab0
0x1192ce:    mov     edi, 0x7f
0x1192d3:    call    0x1048a0
0x1192f1:    mov     edi, 2
0x1192f6:    call    0x104ab0
0x1192fb:    mov     rdi, qword ptr [rip + 0x127ce]
0x119302:    mov     rsi, rbp
0x119305:    call    0x104da0
0x11930a:    mov     rdi, qword ptr [rip + 0x127b7]
0x119311:    mov     rsi, rbp
0x119314:    call    0x104da0
0x119319:    mov     edx, 4
0x11931e:    lea     rsi, [rip + 0x90b6]
0x119325:    mov     rdi, r12
0x119328:    call    0x104890
0x11932d:    test    eax, eax
0x11932f:    je      0x1192a5
0x119335:    xor     edi, edi
0x119337:    call    0x104ab0
0x11933c:    jmp     0x1192a5
0x1192d8:    nop     dword ptr [rax + rax]
0x119341:    nop     word ptr cs:[rax + rax]
0x11934c:    nop     dword ptr [rax]
0x119350:    endbr64
0x119354:    push    r12
0x119356:    push    rbp
0x119357:    push    rbx
0x119358:    sub     rsp, 0x10
0x11935c:    mov     rax, qword ptr fs:[0x28]

```

```

0x119365:    mov     qword ptr [rsp + 8], rax
0x11936a:    xor     eax, eax
0x11936c:    test    rdi, rdi
0x11936f:    je      0x1193d0
0x1193d0:    mov     eax, 1
0x1193d5:    jmp     0x1193a0
0x119371:    mov     rbp, rdi
0x119374:    mov     r12d, esi
0x119377:    call    0x104ed0
0x1193a0:    mov     rdx, qword ptr [rsp + 8]
0x1193a5:    sub     rdx, qword ptr fs:[0x28]
0x1193ae:    jne     0x1193d7
0x11937c:    mov     ebx, eax
0x11937e:    test    eax, eax
0x119380:    js      0x1193c0
0x1193b0:    add     rsp, 0x10
0x1193b4:    pop     rbx
0x1193b5:    pop     rbp
0x1193b6:    pop     r12
0x1193b8:    ret
0x1193d7:    call    0x1049d0
0x1193c0:    mov     eax, 0xffffffff
0x1193c5:    jmp     0x1193a0
0x119382:    je      0x1193dc
0x1193dc:    mov     esi, r12d
0x1193df:    mov     rdi, rbp
0x1193e2:    xor     eax, eax
0x1193e4:    call    0x119280
0x119384:    lea     rbp, [rsp + 4]
0x119389:    nop     dword ptr [rax]
0x119390:    mov     rdi, rbp
0x119393:    call    0x104ec0
0x119398:    cmp     eax, ebx
0x11939a:    jne     0x119390
0x1193e9:    nop     dword ptr [rax]
0x11939c:    mov     eax, dword ptr [rsp + 4]
0x1193a0:    mov     rdx, qword ptr [rsp + 8]
0x1193a5:    sub     rdx, qword ptr fs:[0x28]
0x1193ae:    jne     0x1193d7
0x119390:    mov     rdi, rbp
0x119393:    call    0x104ec0
0x1193b9:    nop     dword ptr [rax]
0x1193c7:    nop     word ptr [rax + rax]
0x1193f0:    endbr64
0x1193f4:    push    r13
0x1193f6:    mov     r13, rdi
0x1193f9:    push    r12
0x1193fb:    push    rbp

```

```

0x1193fc:    push    rbx
0x1193fd:    sub     rsp, 0x18
0x119401:    mov     rax, qword ptr fs:[0x28]
0x11940a:    mov     qword ptr [rsp + 8], rax
0x11940f:    xor     eax, eax
0x119411:    test    esi, esi
0x119413:    je      0x119488
0x119488:    xor     edi, edi
0x11948a:    call    0x109b00
0x119415:    mov     edi, 1
0x11941a:    mov     ebp, esi
0x11941c:    call    0x118b70
0x11948f:    xor     eax, eax
0x119491:    call    0x10a6c0
0x119421:    test    r13, r13
0x119424:    je      0x119520
0x119496:    mov     rdi, qword ptr [rip + 0x1142b]
0x11949d:    call    0x10bb40
0x119520:    mov     r12d, 1
0x119526:    jmp     0x11945a
0x11942a:    call    0x104ed0
0x1194a2:    xor     edi, edi
0x1194a4:    call    0x118b90
0x11945a:    xor     edi, edi
0x11945c:    call    0x118b70
0x11942f:    mov     r12d, eax
0x119432:    test    eax, eax
0x119434:    js      0x119550
0x1194a9:    test    r13, r13
0x1194ac:    je      0x119530
0x119461:    mov     rax, qword ptr [rsp + 8]
0x119466:    sub     rax, qword ptr fs:[0x28]
0x11946f:    jne     0x11955b
0x119550:    mov     r12d, 0xffffffff
0x119556:    jmp     0x11945a
0x11943a:    je      0x119560
0x119530:    mov     r12d, 1
0x119536:    jmp     0x1194e1
0x1194b2:    call    0x104ed0
0x119475:    add     rsp, 0x18
0x119479:    mov     eax, r12d
0x11947c:    pop     rbx
0x11947d:    pop     rbp
0x11947e:    pop     r12
0x119480:    pop     r13
0x119482:    ret
0x11955b:    call    0x1049d0
0x119560:    mov     esi, ebp

```



```

0x119562:    mov     rdi, r13
0x119565:    xor     eax, eax
0x119567:    call    0x119280
0x119440:    lea     rbx, [rsp + 4]
0x119445:    nop     dword ptr [rax]
0x119448:    mov     rdi, rbx
0x11944b:    call    0x104ec0
0x1194e1:    mov     edi, 1
0x1194e6:    call    0x118b90
0x1194b7:    mov     ebp, eax
0x1194b9:    test    eax, eax
0x1194bb:    js      0x119540
0x119450:    cmp     r12d, eax
0x119453:    jne     0x119448
0x1194eb:    mov     edx, 2
0x1194f0:    mov     edi, 1
0x1194f5:    lea     rsi, [rip + 0x8ee7]
0x1194fc:    call    0x118da0
0x119540:    mov     r12d, 0xffffffff
0x119546:    jmp     0x1194e1
0x1194c1:    je      0x11956c
0x11956c:    xor     esi, esi
0x11956e:    mov     rdi, r13
0x119571:    xor     eax, eax
0x119573:    call    0x119280
0x119455:    mov     r12d, dword ptr [rsp + 4]
0x11945a:    xor     edi, edi
0x11945c:    call    0x118b70
0x119448:    mov     rdi, rbx
0x11944b:    call    0x104ec0
0x119501:    mov     edi, 1
0x119506:    call    0x109b00
0x1194c7:    lea     rbx, [rsp + 4]
0x1194cc:    nop     dword ptr [rax]
0x1194d0:    mov     rdi, rbx
0x1194d3:    call    0x104ec0
0x119578:    nop     dword ptr [rax + rax]
0x11950b:    xor     eax, eax
0x11950d:    call    0x10a6c0
0x1194d8:    cmp     ebp, eax
0x1194da:    jne     0x1194d0
0x119512:    xor     edi, edi
0x119514:    call    0x107890
0x1194dc:    mov     r12d, dword ptr [rsp + 4]
0x1194e1:    mov     edi, 1
0x1194e6:    call    0x118b90
0x1194d0:    mov     rdi, rbx
0x1194d3:    call    0x104ec0

```

```

0x119519:      jmp      0x119461
0x119483:      nop      dword ptr [rax + rax]
0x11951e:      nop
0x11952b:      nop      dword ptr [rax + rax]
0x119538:      nop      dword ptr [rax + rax]
0x119548:      nop      dword ptr [rax + rax]
0x119580:      endbr64
0x119584:      mov      rdi, qword ptr [rip + 0x12545]
0x11958b:      test     rdi, rdi
0x11958e:      je       0x1195b8
0x1195b8:      mov      rdi, qword ptr [rip + 0x12509]
0x1195bf:      test     rdi, rdi
0x1195c2:      je       0x1195d0
0x119590:      sub      rsp, 8
0x119594:      call     0x104870
0x1195d0:      ret
0x1195c4:      jmp      0x104870
0x119599:      mov      rdi, qword ptr [rip + 0x12528]
0x1195a0:      test     rdi, rdi
0x1195a3:      je       0x1195b0
0x1195b0:      add      rsp, 8
0x1195b4:      ret
0x1195a5:      add      rsp, 8
0x1195a9:      jmp      0x104870
0x1195ae:      nop
0x1195b5:      nop      dword ptr [rax]
0x1195c9:      nop      dword ptr [rax]
0x1195d1:      nop      word ptr cs:[rax + rax]
0x1195dc:      nop      dword ptr [rax]
0x1195e0:      endbr64
0x1195e4:      push     r12
0x1195e6:      lea      rsi, [rip + 0x89dc]
0x1195ed:      push     rbp
0x1195ee:      mov      rbp, rdi
0x1195f1:      sub      rsp, 8
0x1195f5:      mov      rdi, qword ptr [rip + 0x124cc]
0x1195fc:      call     0x104da0
0x119601:      test     rax, rax
0x119604:      je       0x119676
0x119676:      mov      rdi, rbp
0x119679:      call     0x1049b0
0x119606:      mov      edi, 0x801
0x11960b:      mov      r12, rax
0x11960e:      call     0x11d670
0x11967e:      mov      rdi, qword ptr [rip + 0x12443]
0x119685:      mov      r12, rax
0x119688:      call     0x1049b0
0x119613:      mov      rbp, rax

```

```

0x119616:      jmp      0x11964a
0x11968d:      lea      rdi, [r12 + rax + 0x20]
0x119692:      call     0x11d670
0x11964a:      mov      rdx, r12
0x11964d:      mov      esi, 0x801
0x119652:      mov      rdi, rbp
0x119655:      call     0x104b00
0x119697:      mov      r9, qword ptr [rip + 0x1242a]
0x11969e:      mov      r8, rbp
0x1196a1:      or       rdx, 0xffffffffffffffff
0x1196a5:      mov      r12, rax
0x1196a8:      mov      rdi, rax
0x1196ab:      mov      esi, 1
0x1196b0:      xor      eax, eax
0x1196b2:      lea      rcx, [rip + 0x8d2d]
0x1196b9:      call     0x104f10
0x11965a:      test     rax, rax
0x11965d:      jne      0x119620
0x1196be:      mov      rdi, r12
0x1196c1:      mov      edx, 0x1e
0x1196c6:      xor      esi, esi
0x1196c8:      call     0x108070
0x11965f:      mov      rdi, rbp
0x119662:      call     0x118d10
0x119620:      mov      rdi, rbp
0x119623:      call     0x1049b0
0x1196cd:      pop      rax
0x1196ce:      mov      rdi, r12
0x1196d1:      pop      rbp
0x1196d2:      pop      r12
0x1196d4:      jmp      0x118d10
0x119667:      add      rsp, 8
0x11966b:      mov      rdi, r12
0x11966e:      pop      rbp
0x11966f:      pop      r12
0x119671:      jmp      0x104940
0x119628:      cdqe
0x11962a:      lea      rax, [rbp + rax - 1]
0x11962f:      cmp      byte ptr [rax], 0xa
0x119632:      jne      0x119637
0x119634:      mov      byte ptr [rax], 0
0x119637:      xor      esi, esi
0x119639:      mov      edx, 0xa
0x11963e:      mov      rdi, rbp
0x119641:      call     0x108070
0x119637:      xor      esi, esi
0x119639:      mov      edx, 0xa
0x11963e:      mov      rdi, rbp

```

```

0x119641:      call    0x108070
0x119646:      test    al, al
0x119648:      je      0x11965f
0x119618:      nop     dword ptr [rax + rax]
0x1196d9:      nop     dword ptr [rax]
0x1196e0:      endbr64
0x1196e4:      push    r15
0x1196e6:      push    r14
0x1196e8:      push    r13
0x1196ea:      push    r12
0x1196ec:      push    rbp
0x1196ed:      push    rbx
0x1196ee:      sub     rsp, 8
0x1196f2:      test    rdi, rdi
0x1196f5:      je      0x119878
0x119878:      xor     r12d, r12d
0x11987b:      jmp     0x119800
0x1196fb:      mov     rbp, rdi
0x1196fe:      call    0x1049b0
0x119800:      add     rsp, 8
0x119804:      mov     rax, r12
0x119807:      pop     rbx
0x119808:      pop     rbp
0x119809:      pop     r12
0x11980b:      pop     r13
0x11980d:      pop     r14
0x11980f:      pop     r15
0x119811:      ret
0x119703:      lea     rdi, [rax + 1]
0x119707:      call    0x11d670
0x11970c:      mov     rsi, rbp
0x11970f:      mov     rdi, rax
0x119712:      call    0x1048b0
0x119717:      cmp     byte ptr [rax], 0x7e
0x11971a:      mov     rbp, rax
0x11971d:      mov     r12, rax
0x119720:      jne     0x119800
0x119726:      movzx   eax, byte ptr [rax + 1]
0x11972a:      lea     r13, [rbp + 1]
0x11972e:      test    al, al
0x119730:      je      0x119818
0x119818:      lea     rdi, [rip + 0x6d41]
0x11981f:      call    0x1047e0
0x119736:      cmp     al, 0x2f
0x119738:      je      0x119818
0x119824:      mov     r14, rax
0x119827:      test    rax, rax
0x11982a:      je      0x1198f8

```

```

0x11973e:    mov     rdi, rbp
0x119741:    call   0x1049b0
0x1198f8:    call   0x1049e0
0x119830:    mov     rdi, r14
0x119833:    call   0x1049b0
0x119746:    mov     rdi, rax
0x119749:    call   0x11d670
0x1198fd:    mov     edi, eax
0x1198ff:    call   0x104970
0x119838:    mov     rdi, r13
0x11983b:    mov     rbx, rax
0x11983e:    call   0x1049b0
0x11974e:    movzx   ecx, byte ptr [rbp + 1]
0x119752:    mov     r14, rax
0x119755:    cmp     cl, 0x2f
0x119758:    je      0x119938
0x119904:    test    rax, rax
0x119907:    je      0x119916
0x119843:    movsxd  rbx, ebx
0x119846:    lea     rdi, [rbx + rax + 1]
0x11984b:    call   0x11d670
0x119938:    mov     rax, r14
0x11993b:    jmp     0x119794
0x11975e:    test    cl, cl
0x119760:    je      0x119938
0x119916:    mov     rdi, r13
0x119919:    call   0x1049b0
0x119909:    mov     r14, qword ptr [rax + 0x20]
0x11990d:    test    r14, r14
0x119910:    jne     0x119830
0x119850:    mov     rsi, r14
0x119853:    mov     r12, rax
0x119856:    mov     rdi, rax
0x119859:    call   0x1048b0
0x119794:    mov     byte ptr [rax], 0
0x119797:    mov     rdi, r14
0x11979a:    call   0x104b60
0x119766:    mov     edx, 2
0x11976b:    nop     dword ptr [rax + rax]
0x119770:    mov     byte ptr [r14 + rdx - 2], cl
0x119775:    movzx   ecx, byte ptr [rbp + rdx]
0x11977a:    lea     r13, [rbp + rdx]
0x11977f:    mov     rax, rdx
0x119782:    add     rdx, 1
0x119786:    test    cl, cl
0x119788:    je      0x11978f
0x11991e:    lea     rdi, [rax + 1]
0x119922:    call   0x11d670

```

```

0x11985e:    lea     rdi, [r12 + rbx]
0x119862:    mov     rsi, r13
0x119865:    call    0x1048b0
0x11979f:    mov     rbx, rax
0x1197a2:    test    rax, rax
0x1197a5:    je      0x119880
0x11978f:    lea     rax, [r14 + rax - 1]
0x119794:    mov     byte ptr [rax], 0
0x119797:    mov     rdi, r14
0x11979a:    call    0x104b60
0x11978a:    cmp     cl, 0x2f
0x11978d:    jne     0x119770
0x119927:    mov     r12, rax
0x11992a:    mov     rdi, rax
0x11992d:    jmp     0x119862
0x11986a:    mov     rdi, rbp
0x11986d:    call    0x118d10
0x119880:    mov     rdx, qword ptr [rip + 0x12251]
0x119887:    test    rdx, rdx
0x11988a:    je      0x1197f3
0x1197ab:    mov     rdi, qword ptr [rax + 0x20]
0x1197af:    call    0x1049b0
0x119770:    mov     byte ptr [r14 + rdx - 2], cl
0x119775:    movzx   ecx, byte ptr [rbp + rdx]
0x11977a:    lea     r13, [rbp + rdx]
0x11977f:    mov     rax, rdx
0x119782:    add     rdx, 1
0x119786:    test    cl, cl
0x119788:    je      0x11978f
0x119862:    mov     rsi, r13
0x119865:    call    0x1048b0
0x119872:    jmp     0x119800
0x1197f3:    call    0x104c70
0x119890:    mov     rdi, r14
0x119893:    xor     eax, eax
0x119895:    call    rdx
0x1197b4:    mov     rdi, r13
0x1197b7:    mov     r15, rax
0x1197ba:    call    0x1049b0
0x1197f8:    mov     rdi, r14
0x1197fb:    call    0x118d10
0x119897:    mov     r15, rax
0x11989a:    test    rax, rax
0x11989d:    je      0x1197f3
0x1197bf:    lea     edi, [r15 + 1]
0x1197c3:    movsxd  rdi, edi
0x1197c6:    add     rdi, rax
0x1197c9:    call    0x11d670

```

```

0x1198a3:      mov     rdi, rax
0x1198a6:      call   0x1049b0
0x1197ce:      mov     rsi, qword ptr [rbx + 0x20]
0x1197d2:      mov     rdi, rax
0x1197d5:      mov     r12, rax
0x1197d8:      call   0x1048b0
0x1198ab:      mov     rdi, r13
0x1198ae:      mov     rbx, rax
0x1198b1:      call   0x1049b0
0x1197dd:      movsxd  rdi, r15d
0x1197e0:      mov     rsi, r13
0x1197e3:      add     rdi, r12
0x1197e6:      call   0x1048b0
0x1198b6:      lea     edi, [rbx + 1]
0x1198b9:      movsxd  rdi, edi
0x1198bc:      add     rdi, rax
0x1198bf:      call   0x11d670
0x1197eb:      mov     rdi, rbp
0x1197ee:      call   0x118d10
0x1198c4:      mov     rsi, r15
0x1198c7:      mov     rdi, rax
0x1198ca:      mov     r12, rax
0x1198cd:      call   0x1048b0
0x1198d2:      movsxd  rdi, ebx
0x1198d5:      mov     rsi, r13
0x1198d8:      add     rdi, r12
0x1198db:      call   0x1048b0
0x1198e0:      mov     rdi, r15
0x1198e3:      call   0x118d10
0x1198e8:      mov     rdi, rbp
0x1198eb:      call   0x118d10
0x1198f0:      jmp     0x1197f3
0x119812:      nop     word ptr [rax + rax]
0x119874:      nop     dword ptr [rax]
0x11987d:      nop     dword ptr [rax]
0x1198f5:      nop     dword ptr [rax]
0x119932:      nop     word ptr [rax + rax]
0x119940:      endbr64
0x119944:      push    r15
0x119946:      push    r14
0x119948:      mov     r14, rdi
0x11994b:      push    r13
0x11994d:      push    r12
0x11994f:      push    rbp
0x119950:      push    rbx
0x119951:      xor     ebx, ebx
0x119953:      sub     rsp, 0x48
0x119957:      mov     dword ptr [rsp + 0x1c], 0

```

```

0x11995f:    mov     qword ptr [rsp + 0x10], 0
0x119968:    nop     dword ptr [rax + rax]
0x119970:    mov     r15, qword ptr [rip + 0x10e31]
0x119977:    mov     rdi, r14
0x11997a:    mov     qword ptr [rsp + 8], r15
0x11997f:    call    0x1049b0
0x119984:    mov     qword ptr [rsp + 0x20], rax
0x119989:    mov     rcx, rax
0x11998c:    movzx   eax, byte ptr [r14]
0x119990:    test    al, al
0x119992:    je      0x119bd0
0x119bd0:    mov     dword ptr [rsp + 8], ebx
0x119bd4:    mov     r15, r14
0x119bd7:    mov     ecx, dword ptr [rsp + 8]
0x119bdb:    xor     r13d, r13d
0x119bde:    mov     dword ptr [rsp + 0x18], 0
0x119be6:    cmp     dword ptr [rsp + 0x1c], ecx
0x119bea:    jg      0x119a62
0x119998:    cmp     al, 0x7e
0x11999a:    je      0x119bd0
0x119bf0:    mov     eax, dword ptr [rsp + 0x1c]
0x119bf4:    mov     ecx, dword ptr [rsp + 0x18]
0x119bf8:    mov     rdi, qword ptr [rsp + 0x10]
0x119bfd:    lea     eax, [rax + rcx + 0x14]
0x119c01:    lea     esi, [rax + 1]
0x119c04:    mov     dword ptr [rsp + 0x1c], eax
0x119c08:    movsxd  rsi, esi
0x119c0b:    call    0x11d690
0x119a62:    movsxd  rbx, ebx
0x119a65:    add     rbx, qword ptr [rsp + 0x10]
0x119a6a:    mov     rdx, r13
0x119a6d:    mov     rsi, r14
0x119a70:    mov     rdi, rbx
0x119a73:    mov     qword ptr [rsp + 0x28], rbx
0x119a78:    call    0x104880
0x1199a0:    mov     dword ptr [rsp + 0x18], ecx
0x1199a4:    mov     rax, rcx
0x1199a7:    test    r15, r15
0x1199aa:    je      0x119c50
0x119c10:    mov     qword ptr [rsp + 0x10], rax
0x119c15:    jmp     0x119a62
0x119a7d:    mov     rdi, r15
0x119a80:    mov     r13, qword ptr [rip + 0x10d19]
0x119a87:    call    0x1049b0
0x119c50:    mov     ecx, eax
0x119c52:    movsxd  r13, eax
0x119c55:    add     ecx, ebx
0x119c57:    lea     r15, [r14 + r13]

```



```

0x119c5b:    mov     dword ptr [rsp + 8], ecx
0x119c5f:    jmp     0x119a54
0x1199b0:    mov     edx, dword ptr [rsp + 0x20]
0x1199b4:    test    edx, edx
0x1199b6:    jle     0x119c4a
0x119a8c:    mov     dword ptr [rsp + 0x20], eax
0x119a90:    test    eax, eax
0x119a92:    jle     0x119c64
0x119a54:    mov     ecx, dword ptr [rsp + 8]
0x119a58:    cmp     dword ptr [rsp + 0x1c], ecx
0x119a5c:    jle     0x119bf0
0x119c4a:    mov     rax, qword ptr [rsp + 0x20]
0x119c4f:    nop
0x119c50:    mov     ecx, eax
0x119c52:    movsxd  r13, eax
0x119c55:    add     ecx, ebx
0x119c57:    lea     r15, [r14 + r13]
0x119c5b:    mov     dword ptr [rsp + 8], ecx
0x119c5f:    jmp     0x119a54
0x1199bc:    mov     rax, qword ptr [rsp + 8]
0x1199c1:    mov     dword ptr [rsp + 0x3c], ebx
0x1199c5:    xor     r12d, r12d
0x1199c8:    mov     rax, qword ptr [rax]
0x1199cb:    mov     qword ptr [rsp + 0x28], rax
0x1199d0:    mov     rax, qword ptr [rsp + 0x20]
0x1199d5:    sub     eax, 1
0x1199d8:    mov     qword ptr [rsp + 0x30], rax
0x1199dd:    mov     rax, qword ptr [rsp + 8]
0x1199e2:    mov     dword ptr [rsp + 0x38], r12d
0x1199e7:    lea     rbp, [r14 + r12]
0x1199eb:    lea     rbx, [rax + 8]
0x1199ef:    mov     rax, qword ptr [rsp + 0x28]
0x1199f4:    mov     r13, rax
0x1199f7:    test    rax, rax
0x1199fa:    jne     0x119a18
0x119c64:    xor     ebp, ebp
0x119c66:    jmp     0x119b84
0x119a98:    mov     rbx, r15
0x119a9b:    xor     ebp, ebp
0x119a9d:    movzx   eax, byte ptr [rbx]
0x119aa0:    cmp     al, 0x2f
0x119aa2:    je      0x119b80
0x1199fc:    jmp     0x119bb0
0x119a18:    mov     rdi, r13
0x119a1b:    call    0x1049b0
0x119b84:    mov     eax, dword ptr [rsp + 0x18]
0x119b88:    test    eax, eax
0x119b8a:    jne     0x119b00

```

```

0x119b80:    or     dword ptr [rsp + 0x18], ebp
0x119b84:    mov     eax, dword ptr [rsp + 0x18]
0x119b88:    test    eax, eax
0x119b8a:    jne     0x119b00
0x119aa8:    test    al, al
0x119aaa:    je      0x119b80
0x119bb0:    lea     rax, [r12 + 1]
0x119bb5:    cmp     qword ptr [rsp + 0x30], r12
0x119bba:    je      0x119c46
0x119a20:    mov     rsi, r13
0x119a23:    mov     rdi, rbp
0x119a26:    mov     rdx, rax
0x119a29:    mov     r15, rax
0x119a2c:    call    0x104890
0x119b90:    mov     rax, qword ptr [rsp + 0x28]
0x119b95:    mov     byte ptr [rax], 0
0x119b98:    mov     rax, qword ptr [rsp + 0x10]
0x119b9d:    add     rsp, 0x48
0x119ba1:    pop     rbx
0x119ba2:    pop     rbp
0x119ba3:    pop     r12
0x119ba5:    pop     r13
0x119ba7:    pop     r14
0x119ba9:    pop     r15
0x119bab:    ret
0x119b00:    lea     edi, [rbp + 1]
0x119b03:    movsxd  rbp, ebp
0x119b06:    movsxd  rdi, edi
0x119b09:    lea     r14, [r15 + rbp]
0x119b0d:    call    0x11d670
0x119ab0:    test    r13, r13
0x119ab3:    je      0x119aea
0x119c46:    mov     ebx, dword ptr [rsp + 0x3c]
0x119c4a:    mov     rax, qword ptr [rsp + 0x20]
0x119c4f:    nop
0x119c50:    mov     ecx, eax
0x119c52:    movsxd  r13, eax
0x119c55:    add     ecx, ebx
0x119c57:    lea     r15, [r14 + r13]
0x119c5b:    mov     dword ptr [rsp + 8], ecx
0x119c5f:    jmp     0x119a54
0x119bc0:    mov     r12, rax
0x119bc3:    jmp     0x1199dd
0x119a31:    test    eax, eax
0x119a33:    jne     0x119a08
0x119b12:    mov     rdx, rbp
0x119b15:    mov     rsi, r15
0x119b18:    mov     r13, rax

```

```

0x119b1b:    mov     rdi, rax
0x119b1e:    call   0x104880
0x119aea:    add     ebp, 1
0x119aed:    add     rbx, 1
0x119af1:    cmp     dword ptr [rsp + 0x20], ebp
0x119af5:    jne     0x119a9d
0x119ab5:    mov     r14, r13
0x119ab8:    jmp     0x119ae2
0x1199dd:    mov     rax, qword ptr [rsp + 8]
0x1199e2:    mov     dword ptr [rsp + 0x38], r12d
0x1199e7:    lea     rbp, [r14 + r12]
0x1199eb:    lea     rbx, [rax + 8]
0x1199ef:    mov     rax, qword ptr [rsp + 0x28]
0x1199f4:    mov     r13, rax
0x1199f7:    test    rax, rax
0x1199fa:    jne     0x119a18
0x119a35:    mov     eax, dword ptr [rsp + 0x38]
0x119a39:    mov     ebx, dword ptr [rsp + 0x3c]
0x119a3d:    lea     eax, [rax + r15 - 1]
0x119a42:    lea     ecx, [rbx + rax]
0x119a45:    mov     dword ptr [rsp + 0x18], eax
0x119a49:    movsxd  r13, eax
0x119a4c:    mov     dword ptr [rsp + 8], ecx
0x119a50:    lea     r15, [r14 + r13]
0x119a54:    mov     ecx, dword ptr [rsp + 8]
0x119a58:    cmp     dword ptr [rsp + 0x1c], ecx
0x119a5c:    jle     0x119bf0
0x119a08:    mov     r13, qword ptr [rbx]
0x119a0b:    add     rbx, 8
0x119a0f:    test    r13, r13
0x119a12:    je      0x119bb0
0x119b23:    mov     byte ptr [r13 + rbp], 0
0x119b29:    mov     rdi, r13
0x119b2c:    xor     eax, eax
0x119b2e:    call   0x1196e0
0x119a9d:    movzx   eax, byte ptr [rbx]
0x119aa0:    cmp     al, 0x2f
0x119aa2:    je      0x119b80
0x119af7:    nop     word ptr [rax + rax]
0x119ae2:    mov     r12, qword ptr [r14]
0x119ae5:    test    r12, r12
0x119ae8:    jne     0x119ac0
0x119b33:    mov     rdi, r13
0x119b36:    mov     rbp, rax
0x119b39:    call   0x118d10
0x119ac0:    mov     rdi, r12
0x119ac3:    add     r14, 8
0x119ac7:    call   0x1049b0

```

```

0x119b3e:    mov     rdi, rbp
0x119b41:    call   0x1049b0
0x119acc:    mov     rsi, r12
0x119acf:    mov     rdi, rbx
0x119ad2:    mov     rdx, rax
0x119ad5:    call   0x104890
0x119b46:    mov     ecx, dword ptr [rsp + 8]
0x119b4a:    lea     ebx, [rcx + rax]
0x119b4d:    cmp     ebx, dword ptr [rsp + 0x1c]
0x119b51:    jge     0x119c20
0x119ada:    test    eax, eax
0x119adc:    je      0x119b80
0x119b57:    movsxd  rdi, dword ptr [rsp + 8]
0x119b5c:    mov     rsi, rbp
0x119b5f:    add     rdi, qword ptr [rsp + 0x10]
0x119b64:    call   0x1048b0
0x119c20:    mov     ecx, dword ptr [rsp + 0x1c]
0x119c24:    mov     rdi, qword ptr [rsp + 0x10]
0x119c29:    lea     eax, [rcx + rax + 0x14]
0x119c2d:    lea     esi, [rax + 1]
0x119c30:    mov     dword ptr [rsp + 0x1c], eax
0x119c34:    movsxd  rsi, esi
0x119c37:    call   0x11d690
0x119b69:    mov     rdi, rbp
0x119b6c:    call   0x118d10
0x119c3c:    mov     qword ptr [rsp + 0x10], rax
0x119c41:    jmp     0x119b57
0x119b71:    jmp     0x119970
0x119970:    mov     r15, qword ptr [rip + 0x10e31]
0x119977:    mov     rdi, r14
0x11997a:    mov     qword ptr [rsp + 8], r15
0x11997f:    call   0x1049b0
0x119a01:    nop     dword ptr [rax]
0x119aba:    nop     word ptr [rax + rax]
0x119b76:    nop     word ptr cs:[rax + rax]
0x119bac:    nop     dword ptr [rax]
0x119bc8:    nop     dword ptr [rax + rax]
0x119c1a:    nop     word ptr [rax + rax]
0x119c6b:    nop     dword ptr [rax + rax]
0x119c70:    push    r15
0x119c72:    mov     r15, rsi
0x119c75:    push    r14
0x119c77:    mov     r14d, edx
0x119c7a:    push    r13
0x119c7c:    push    r12
0x119c7e:    mov     r12, rdi
0x119c81:    push    rbp
0x119c82:    push    rbx

```

```

0x119c83:    sub     rsp, 0x18
0x119c87:    movsxd  rsi, dword ptr [rsi]
0x119c8a:    lea     r13, [rdi + rsi]
0x119c8e:    mov     edi, ecx
0x119c90:    mov     ebx, esi
0x119c92:    mov     rbp, rsi
0x119c95:    movzx   eax, byte ptr [r13]
0x119c9a:    and     edi, 1
0x119c9d:    mov     rdx, rsi
0x119ca0:    test    al, al
0x119ca2:    jne     0x119cdc
0x119ca4:    jmp     0x119db9
0x119cdc:    movsx   ecx, al
0x119cdf:    cmp     ecx, r14d
0x119ce2:    jne     0x119cb0
0x119db9:    test    ecx, ecx
0x119dbb:    je      0x119db1
0x119ce4:    cmp     ebp, ebx
0x119ce6:    jl      0x119d10
0x119cb0:    cmp     al, 0x5c
0x119cb2:    jne     0x119cc5
0x119db1:    xor     r9d, r9d
0x119db4:    jmp     0x119cfa
0x119dbd:    mov     edi, 1
0x119dc2:    mov     qword ptr [rsp], r8
0x119dc6:    call    0x11d670
0x119d10:    mov     edi, ebx
0x119d12:    mov     qword ptr [rsp], r8
0x119d16:    sub     edi, ebp
0x119d18:    mov     qword ptr [rsp + 8], rsi
0x119d1d:    add     edi, 1
0x119d20:    movsxd  rdi, edi
0x119d23:    call    0x11d670
0x119ce8:    test    dil, dil
0x119ceb:    jne     0x119d10
0x119cb4:    movsx   eax, byte ptr [r12 + rdx + 1]
0x119cba:    cmp     eax, r14d
0x119cbd:    sete    al
0x119cc0:    movzx   eax, al
0x119cc3:    add     ebx, eax
0x119cc5:    add     ebx, 1
0x119cc8:    movsxd  rdx, ebx
0x119ccb:    lea     r13, [r12 + rdx]
0x119ccf:    movzx   eax, byte ptr [r13]
0x119cd4:    test    al, al
0x119cd6:    je      0x119da0
0x119cc5:    add     ebx, 1
0x119cc8:    movsxd  rdx, ebx

```

```

0x119ccb:    lea     r13, [r12 + rdx]
0x119ccf:    movzx  eax, byte ptr [r13]
0x119cd4:    test   al, al
0x119cd6:    je     0x119da0
0x119cfa:    mov     dword ptr [r15], ebx
0x119cfd:    add     rsp, 0x18
0x119d01:    mov     rax, r9
0x119d04:    pop     rbx
0x119d05:    pop     rbp
0x119d06:    pop     r12
0x119d08:    pop     r13
0x119d0a:    pop     r14
0x119d0c:    pop     r15
0x119d0e:    ret
0x119dcb:    mov     r8, qword ptr [rsp]
0x119dcf:    xor     edx, edx
0x119dd1:    mov     r9, rax
0x119dd4:    jmp     0x119d86
0x119d28:    cmp     ebp, ebx
0x119d2a:    mov     r8, qword ptr [rsp]
0x119d2e:    mov     r9, rax
0x119d31:    jge     0x119dd6
0x119ced:    movzx  eax, byte ptr [r13]
0x119cf2:    xor     r9d, r9d
0x119cf5:    cmp     al, 1
0x119cf7:    sbb     ebx, -1
0x119cfa:    mov     dword ptr [r15], ebx
0x119cfd:    add     rsp, 0x18
0x119d01:    mov     rax, r9
0x119d04:    pop     rbx
0x119d05:    pop     rbp
0x119d06:    pop     r12
0x119d08:    pop     r13
0x119d0a:    pop     r14
0x119d0c:    pop     r15
0x119d0e:    ret
0x119da0:    cmp     ebp, ebx
0x119da2:    jl      0x119d10
0x119d86:    mov     byte ptr [rax], 0
0x119d89:    mov     dword ptr [r8], edx
0x119d8c:    movzx  eax, byte ptr [r13]
0x119d91:    jmp     0x119cf5
0x119d37:    mov     rsi, qword ptr [rsp + 8]
0x119d3c:    mov     eax, 1
0x119d41:    jmp     0x119d5d
0x119dd6:    xor     edx, edx
0x119dd8:    jmp     0x119d86
0x119da8:    test    dil, dil

```

```

0x119dab:    jne     0x119d10
0x119cf5:    cmp     al, 1
0x119cf7:    sbb     ebx, -1
0x119cfa:    mov     dword ptr [r15], ebx
0x119cfd:    add     rsp, 0x18
0x119d01:    mov     rax, r9
0x119d04:    pop     rbx
0x119d05:    pop     rbp
0x119d06:    pop     r12
0x119d08:    pop     r13
0x119d0a:    pop     r14
0x119d0c:    pop     r15
0x119d0e:    ret
0x119d5d:    movzx   edx, byte ptr [r12 + rsi]
0x119d62:    cmp     dl, 0x5c
0x119d65:    jne     0x119d48
0x119d67:    movsx   esi, byte ptr [r12 + rsi + 1]
0x119d6d:    cmp     esi, r14d
0x119d70:    jne     0x119d48
0x119d48:    add     ebp, 1
0x119d4b:    mov     byte ptr [r9 + rax - 1], dl
0x119d50:    mov     edx, eax
0x119d52:    add     rax, 1
0x119d56:    cmp     ebp, ebx
0x119d58:    jge     0x119d80
0x119d72:    add     ebp, 1
0x119d75:    mov     edx, esi
0x119d77:    jmp     0x119d48
0x119d5a:    movsxd  rsi, ebp
0x119d5d:    movzx   edx, byte ptr [r12 + rsi]
0x119d62:    cmp     dl, 0x5c
0x119d65:    jne     0x119d48
0x119d80:    movsxd  rax, edx
0x119d83:    add     rax, r9
0x119d86:    mov     byte ptr [rax], 0
0x119d89:    mov     dword ptr [r8], edx
0x119d8c:    movzx   eax, byte ptr [r13]
0x119d91:    jmp     0x119cf5
0x119ca9:    nop     dword ptr [rax]
0x119d0f:    nop
0x119d43:    nop     dword ptr [rax + rax]
0x119d79:    nop     dword ptr [rax]
0x119d96:    nop     word ptr cs:[rax + rax]
0x119dda:    nop     word ptr [rax + rax]
0x119de0:    push    r15
0x119de2:    push    r14
0x119de4:    push    r13
0x119de6:    push    r12

```

```

0x119de8:    push    rbp
0x119de9:    push    rbx
0x119dea:    sub     rsp, 0x28
0x119dee:    mov     eax, dword ptr [rip + 0x11cf0]
0x119df4:    mov     ebx, dword ptr [rip + 0x11d02]
0x119dfa:    mov     dword ptr [rsp + 8], esi
0x119dfe:    mov     dword ptr [rsp + 0x18], edx
0x119e02:    mov     dword ptr [rsp + 0x1c], eax
0x119e06:    test    eax, eax
0x119e08:    je      0x119f9d
0x119f9d:    mov     r15d, 0xffffffff
0x119fa3:    jmp     0x119ecd
0x119e0e:    mov     rbp, rdi
0x119e11:    cmp     eax, ebx
0x119e13:    je      0x119fb0
0x119ecd:    add     rsp, 0x28
0x119ed1:    mov     eax, r15d
0x119ed4:    pop     rbx
0x119ed5:    pop     rbp
0x119ed6:    pop     r12
0x119ed8:    pop     r13
0x119eda:    pop     r14
0x119edc:    pop     r15
0x119ede:    ret
0x119fb0:    test    esi, esi
0x119fb2:    jns     0x119f9d
0x119e19:    call    0x1049b0
0x119fb4:    mov     eax, dword ptr [rsp + 0x1c]
0x119fb8:    lea     ebx, [rax - 1]
0x119fbb:    mov     rdi, rbp
0x119fbe:    call    0x1049b0
0x119e1e:    mov     ecx, dword ptr [rsp + 8]
0x119e22:    mov     dword ptr [rsp + 0xc], eax
0x119e26:    test    ecx, ecx
0x119e28:    js      0x119fbb
0x119fc3:    mov     dword ptr [rsp + 0xc], eax
0x119fc7:    test    dword ptr [rsp + 8], ebx
0x119fcb:    jns     0x119e2e
0x119fbb:    mov     rdi, rbp
0x119fbe:    call    0x1049b0
0x119e2e:    mov     rax, qword ptr [rip + 0x11ccb]
0x119e35:    movsxd  r14, dword ptr [rsp + 0xc]
0x119e3a:    mov     qword ptr [rsp + 0x10], rax
0x119e3f:    mov     edx, dword ptr [rsp + 8]
0x119e43:    test    edx, edx
0x119e45:    js      0x119f16
0x119fd1:    jmp     0x119f9d
0x119f16:    mov     rcx, qword ptr [rsp + 0x10]

```



```

0x119f1b:    movsxd    rax, ebx
0x119f1e:    mov      rax, qword ptr [rcx + rax*8]
0x119f22:    mov      r13, qword ptr [rax]
0x119f25:    mov      rdi, r13
0x119f28:    call     0x1049b0
0x119e4b:    cmp      dword ptr [rsp + 0x1c], ebx
0x119e4f:    je       0x119f9d
0x119f2d:    mov      r8, rax
0x119f30:    cmp      dword ptr [rsp + 0xc], eax
0x119f34:    jg       0x119f90
0x119e55:    mov      rcx, qword ptr [rsp + 0x10]
0x119e5a:    movsxd    rax, ebx
0x119e5d:    mov      rax, qword ptr [rcx + rax*8]
0x119e61:    mov      r13, qword ptr [rax]
0x119e64:    mov      rdi, r13
0x119e67:    call     0x1049b0
0x119f36:    cmp      dword ptr [rsp + 0x18], 1
0x119f3b:    movzx    r12d, byte ptr [rbp]
0x119f40:    je       0x119ee0
0x119f90:    sub      ebx, 1
0x119f93:    test     dword ptr [rsp + 8], ebx
0x119f97:    jns      0x119e3f
0x119e6c:    mov      r10, rax
0x119e6f:    cmp      dword ptr [rsp + 0xc], eax
0x119e73:    jg       0x119f07
0x119ee0:    cmp      byte ptr [r13], r12b
0x119ee4:    jne      0x119efb
0x119f42:    mov      r15d, r8d
0x119f45:    sub      r15d, dword ptr [rsp + 0xc]
0x119f4a:    movsxd    rcx, r15d
0x119f4d:    add      r13, rcx
0x119f50:    jmp      0x119f66
0x119e3f:    mov      edx, dword ptr [rsp + 8]
0x119e43:    test     edx, edx
0x119e45:    js       0x119f16
0x119e79:    cmp      dword ptr [rsp + 0x18], 1
0x119e7e:    movzx    r12d, byte ptr [rbp]
0x119e83:    je       0x119ee0
0x119f07:    mov      edx, dword ptr [rsp + 8]
0x119f0b:    add      ebx, 1
0x119f0e:    test     edx, edx
0x119f10:    jns      0x119e4b
0x119ee6:    mov      rdx, r14
0x119ee9:    mov      rsi, r13
0x119eec:    mov      rdi, rbp
0x119eef:    call     0x104890
0x119efb:    mov      eax, dword ptr [rsp + 8]
0x119eff:    test     eax, eax

```

```

0x119f01:      js      0x119f90
0x119f66:      cmp     byte ptr [r13], r12b
0x119f6a:      jne     0x119f58
0x119e85:      mov     r15d, dword ptr [rsp + 0x18]
0x119e8a:      sub     r10d, dword ptr [rsp + 0xc]
0x119e8f:      jmp     0x119ea5
0x119ef4:      mov     r15d, eax
0x119ef7:      test    eax, eax
0x119ef9:      je      0x119ec7
0x119f6c:      mov     rdx, r14
0x119f6f:      mov     rsi, r13
0x119f72:      mov     rdi, rbp
0x119f75:      call    0x104890
0x119f58:      sub     r15d, 1
0x119f5c:      sub     r13, 1
0x119f60:      cmp     r15d, -1
0x119f64:      je      0x119f90
0x119ea5:      cmp     byte ptr [r13], r12b
0x119ea9:      jne     0x119e98
0x119ec7:      mov     dword ptr [rip + 0x11c2f], ebx
0x119ecd:      add     rsp, 0x28
0x119ed1:      mov     eax, r15d
0x119ed4:      pop     rbx
0x119ed5:      pop     rbp
0x119ed6:      pop     r12
0x119ed8:      pop     r13
0x119eda:      pop     r14
0x119edc:      pop     r15
0x119ede:      ret
0x119f7a:      test    eax, eax
0x119f7c:      je      0x119ec7
0x119eab:      mov     rdx, r14
0x119eae:      mov     rsi, r13
0x119eb1:      mov     rdi, rbp
0x119eb4:      mov     dword ptr [rsp + 4], r10d
0x119eb9:      call    0x104890
0x119e98:      add     r15d, 1
0x119e9c:      add     r13, 1
0x119ea0:      cmp     r15d, r10d
0x119ea3:      jg      0x119f07
0x119f82:      sub     r15d, 1
0x119f86:      sub     r13, 1
0x119f8a:      cmp     r15d, -1
0x119f8e:      jne     0x119f66
0x119ebe:      mov     r10d, dword ptr [rsp + 4]
0x119ec3:      test    eax, eax
0x119ec5:      jne     0x119e98
0x119e91:      nop     dword ptr [rax]

```

```

0x119edf:      nop
0x119f52:      nop      word ptr [rax + rax]
0x119fa8:      nop      dword ptr [rax + rax]
0x119fd3:      nop      word ptr cs:[rax + rax]
0x119fde:      nop
0x119fe0:      push     r14
0x119fe2:      push     r13
0x119fe4:      movsxd   r13, esi
0x119fe7:      sub      edx, r13d
0x119fea:      push     r12
0x119fec:      push     rbp
0x119fed:      mov      rbp, rdi
0x119ff0:      push     rbx
0x119ff1:      movsxd   rbx, edx
0x119ff4:      cmp      ecx, 2
0x119ff7:      je       0x11a070
0x11a070:      mov      eax, 0x13
0x11a075:      lea      r14, [rip + 0x83af]
0x11a07c:      jmp      0x11a028
0x119ff9:      mov      eax, 0x1d
0x119ffe:      lea      r14, [rip + 0x843a]
0x11a005:      cmp      ecx, 3
0x11a008:      je       0x11a028
0x11a028:      lea      edi, [rax + rbx + 3]
0x11a02c:      movsxd   rdi, edi
0x11a02f:      call     0x11d670
0x11a00a:      xor      eax, eax
0x11a00c:      lea      r14, [rip + 0x83f5]
0x11a013:      lea      rdx, [rip + 0x83fe]
0x11a01a:      cmp      ecx, 1
0x11a01d:      sete     al
0x11a020:      cmov     r14, rdx
0x11a024:      lea      eax, [rax + rax*2 + 0xf]
0x11a028:      lea      edi, [rax + rbx + 3]
0x11a02c:      movsxd   rdi, edi
0x11a02f:      call     0x11d670
0x11a034:      lea      rsi, [rbp + r13]
0x11a039:      mov      rdx, rbx
0x11a03c:      mov      r12, rax
0x11a03f:      mov      rdi, rax
0x11a042:      call     0x104880
0x11a047:      mov      byte ptr [r12 + rbx], 0x3a
0x11a04c:      lea      rdi, [r12 + rbx + 2]
0x11a051:      mov      rsi, r14
0x11a054:      mov      byte ptr [r12 + rbx + 1], 0x20
0x11a05a:      call     0x1048b0
0x11a05f:      pop      rbx
0x11a060:      mov      rax, r12

```

```

0x11a063:    pop     rbp
0x11a064:    pop     r12
0x11a066:    pop     r13
0x11a068:    pop     r14
0x11a06a:    ret
0x11a06b:    nop     dword ptr [rax + rax]
0x11a07e:    nop
0x11a080:    push    r12
0x11a082:    push    rbp
0x11a083:    push    rbx
0x11a084:    test    rdi, rdi
0x11a087:    je      0x11a0b0
0x11a0b0:    lea     rdi, [rip + 0x64a9]
0x11a0b7:    call    0x1047e0
0x11a089:    mov     rbp, rdi
0x11a08c:    call    0x1049b0
0x11a0bc:    mov     r12, rax
0x11a0bf:    test    rax, rax
0x11a0c2:    je      0x11a110
0x11a091:    lea     rdi, [rax + 1]
0x11a095:    call    0x11d670
0x11a110:    mov     ebx, 2
0x11a115:    mov     ebp, 1
0x11a11a:    mov     edi, 0xb
0x11a11f:    lea     r12, [rip + 0x75b8]
0x11a126:    jmp     0x11a0d9
0x11a0c4:    mov     rdi, rax
0x11a0c7:    call    0x1049b0
0x11a09a:    pop     rbx
0x11a09b:    mov     rsi, rbp
0x11a09e:    pop     rbp
0x11a09f:    mov     rdi, rax
0x11a0a2:    pop     r12
0x11a0a4:    jmp     0x1048b0
0x11a0d9:    call    0x11d670
0x11a0cc:    lea     edi, [rax + 0xa]
0x11a0cf:    movsxd  rbp, eax
0x11a0d2:    movsxd  rdi, edi
0x11a0d5:    lea     rbx, [rbp + 1]
0x11a0d9:    call    0x11d670
0x11a0de:    mov     rsi, r12
0x11a0e1:    mov     rdi, rax
0x11a0e4:    call    0x1048b0
0x11a0e9:    movabs  rdx, 0x79726f747369682e
0x11a0f3:    mov     byte ptr [rax + rbp], 0x2f
0x11a0f7:    mov     r8, rax
0x11a0fa:    lea     rax, [rax + rbx]
0x11a0fe:    mov     qword ptr [rax], rdx

```

```

0x11a101:    mov     byte ptr [rax + 8], 0
0x11a105:    mov     rax, r8
0x11a108:    pop     rbx
0x11a109:    pop     rbp
0x11a10a:    pop     r12
0x11a10c:    ret
0x11a0a9:    nop     dword ptr [rax]
0x11a10d:    nop     dword ptr [rax]
0x11a128:    nop     dword ptr [rax + rax]
0x11a130:    push    r15
0x11a132:    xor     eax, eax
0x11a134:    push    r14
0x11a136:    mov     r14d, esi
0x11a139:    push    r13
0x11a13b:    push    r12
0x11a13d:    push    rbp
0x11a13e:    push    rbx
0x11a13f:    mov     ebx, edx
0x11a141:    sub     rsp, 0x18
0x11a145:    call    0x11a080
0x11a14a:    cmp     ebx, 1
0x11a14d:    mov     edx, 0x1b6
0x11a152:    sbb     esi, esi
0x11a154:    mov     r13, rax
0x11a157:    mov     rdi, rax
0x11a15a:    xor     eax, eax
0x11a15c:    and     esi, 0x1c0
0x11a162:    add     esi, 0x241
0x11a168:    call    0x104d80
0x11a16d:    cmp     eax, -1
0x11a170:    je      0x11a270
0x11a270:    mov     rdi, r13
0x11a273:    call    0x118d10
0x11a176:    mov     ebp, eax
0x11a178:    mov     eax, dword ptr [rip + 0x11966]
0x11a17e:    cmp     eax, r14d
0x11a181:    mov     edx, eax
0x11a183:    cmovle  r14d, eax
0x11a187:    sub     edx, r14d
0x11a18a:    cmp     eax, edx
0x11a18c:    jle     0x11a290
0x11a278:    call    0x104860
0x11a290:    mov     qword ptr [rsp + 8], 0
0x11a299:    jmp     0x11a1d1
0x11a192:    sub     eax, 1
0x11a195:    mov     rcx, qword ptr [rip + 0x11964]
0x11a19c:    movsxd  rsi, edx
0x11a19f:    xor     r12d, r12d

```

```

0x11a1a2:    sub    eax, edx
0x11a1a4:    add    rax, rsi
0x11a1a7:    lea    rbx, [rcx + rsi*8]
0x11a1ab:    lea    r15, [rcx + rax*8 + 8]
0x11a1b0:    mov    rax, qword ptr [rbx]
0x11a1b3:    add    rbx, 8
0x11a1b7:    mov    rdi, qword ptr [rax]
0x11a1ba:    call   0x1049b0
0x11a27d:    mov    eax, dword ptr [rax]
0x11a27f:    add    rsp, 0x18
0x11a283:    pop    rbx
0x11a284:    pop    rbp
0x11a285:    pop    r12
0x11a287:    pop    r13
0x11a289:    pop    r14
0x11a28b:    pop    r15
0x11a28d:    ret
0x11a1d1:    mov    rdi, qword ptr [rsp + 8]
0x11a1d6:    call   0x11d670
0x11a1bf:    lea    r12d, [r12 + rax + 1]
0x11a1c4:    cmp    r15, rbx
0x11a1c7:    jne    0x11a1b0
0x11a1db:    mov    r15, rax
0x11a1de:    mov    eax, dword ptr [rip + 0x11900]
0x11a1e4:    mov    ebx, eax
0x11a1e6:    sub    ebx, r14d
0x11a1e9:    cmp    eax, ebx
0x11a1eb:    jle    0x11a238
0x11a1c9:    movsxd rax, r12d
0x11a1cc:    mov    qword ptr [rsp + 8], rax
0x11a1d1:    mov    rdi, qword ptr [rsp + 8]
0x11a1d6:    call   0x11d670
0x11a1b0:    mov    rax, qword ptr [rbx]
0x11a1b3:    add    rbx, 8
0x11a1b7:    mov    rdi, qword ptr [rax]
0x11a1ba:    call   0x1049b0
0x11a238:    mov    rdx, qword ptr [rsp + 8]
0x11a23d:    mov    rsi, r15
0x11a240:    mov    edi, ebp
0x11a242:    call   0x104790
0x11a1ed:    mov    r12, qword ptr [rip + 0x1190c]
0x11a1f4:    movsxd rbx, ebx
0x11a1f7:    xor    r14d, r14d
0x11a1fa:    nop    word ptr [rax + rax]
0x11a200:    mov    rax, qword ptr [r12 + rbx*8]
0x11a204:    movsxd rdi, r14d
0x11a207:    add    rdi, r15
0x11a20a:    mov    rsi, qword ptr [rax]

```

```

0x11a20d:    call    0x1048b0
0x11a247:    mov     rdi, r15
0x11a24a:    call    0x118d10
0x11a212:    mov     rax, qword ptr [r12 + rbx*8]
0x11a216:    add     rbx, 1
0x11a21a:    mov     rdi, qword ptr [rax]
0x11a21d:    call    0x1049b0
0x11a24f:    mov     edi, ebp
0x11a251:    call    0x104ab0
0x11a222:    add     eax, r14d
0x11a225:    lea     r14d, [rax + 1]
0x11a229:    cdqe
0x11a22b:    mov     byte ptr [r15 + rax], 0xa
0x11a230:    cmp     dword ptr [rip + 0x118ae], ebx
0x11a236:    jg      0x11a200
0x11a256:    mov     rdi, r13
0x11a259:    call    0x118d10
0x11a200:    mov     rax, qword ptr [r12 + rbx*8]
0x11a204:    movsxd  rdi, r14d
0x11a207:    add     rdi, r15
0x11a20a:    mov     rsi, qword ptr [rax]
0x11a20d:    call    0x1048b0
0x11a25e:    add     rsp, 0x18
0x11a262:    xor     eax, eax
0x11a264:    pop     rbx
0x11a265:    pop     rbp
0x11a266:    pop     r12
0x11a268:    pop     r13
0x11a26a:    pop     r14
0x11a26c:    pop     r15
0x11a26e:    ret
0x11a26f:    nop
0x11a28e:    nop
0x11a29e:    nop
0x11a2a0:    endbr64
0x11a2a4:    sub     rsp, 8
0x11a2a8:    mov     edi, 0x18
0x11a2ad:    call    0x11d670
0x11a2b2:    mov     rdx, qword ptr [rip + 0x11847]
0x11a2b9:    mov     qword ptr [rax], rdx
0x11a2bc:    mov     edx, dword ptr [rip + 0x1183a]
0x11a2c2:    mov     dword ptr [rax + 8], edx
0x11a2c5:    mov     edx, dword ptr [rip + 0x11819]
0x11a2cb:    mov     dword ptr [rax + 0xc], edx
0x11a2ce:    mov     edx, dword ptr [rip + 0x11824]
0x11a2d4:    mov     dword ptr [rax + 0x10], edx
0x11a2d7:    add     rsp, 8
0x11a2db:    ret

```

```

0x11a2dc:    nop     dword ptr [rax]
0x11a2e0:    endbr64
0x11a2e4:    mov     rax, qword ptr [rdi]
0x11a2e7:    mov     qword ptr [rip + 0x11812], rax
0x11a2ee:    mov     eax, dword ptr [rdi + 8]
0x11a2f1:    mov     dword ptr [rip + 0x11805], eax
0x11a2f7:    mov     eax, dword ptr [rdi + 0xc]
0x11a2fa:    mov     dword ptr [rip + 0x117e4], eax
0x11a300:    mov     eax, dword ptr [rdi + 0x10]
0x11a303:    mov     dword ptr [rip + 0x117ef], eax
0x11a309:    ret
0x11a30a:    nop     word ptr [rax + rax]
0x11a310:    endbr64
0x11a314:    mov     eax, dword ptr [rip + 0x117ca]
0x11a31a:    mov     dword ptr [rip + 0x117dc], eax
0x11a320:    ret
0x11a321:    nop     word ptr cs:[rax + rax]
0x11a32c:    nop     dword ptr [rax]
0x11a330:    endbr64
0x11a334:    push    r12
0x11a336:    xor     r12d, r12d
0x11a339:    push    rbx
0x11a33a:    sub     rsp, 8
0x11a33e:    mov     rbx, qword ptr [rip + 0x117bb]
0x11a345:    test    rbx, rbx
0x11a348:    jne     0x11a35b
0x11a34a:    jmp     0x11a367
0x11a35b:    mov     rax, qword ptr [rbx]
0x11a35e:    add     rbx, 8
0x11a362:    test    rax, rax
0x11a365:    jne     0x11a350
0x11a367:    add     rsp, 8
0x11a36b:    mov     eax, r12d
0x11a36e:    pop     rbx
0x11a36f:    pop     r12
0x11a371:    ret
0x11a350:    mov     rdi, qword ptr [rax]
0x11a353:    call    0x1049b0
0x11a358:    add     r12d, eax
0x11a35b:    mov     rax, qword ptr [rbx]
0x11a35e:    add     rbx, 8
0x11a362:    test    rax, rax
0x11a365:    jne     0x11a350
0x11a34c:    nop     dword ptr [rax]
0x11a372:    nop     word ptr cs:[rax + rax]
0x11a37d:    nop     dword ptr [rax]
0x11a380:    endbr64
0x11a384:    push    rbp

```



```

0x11a385:    mov     rbp, rdi
0x11a388:    push   rbx
0x11a389:    sub     rsp, 8
0x11a38d:    mov     eax, dword ptr [rip + 0x11759]
0x11a393:    test    eax, eax
0x11a395:    je      0x11a3aa
0x11a3aa:    mov     eax, dword ptr [rip + 0x11748]
0x11a3b0:    test    eax, eax
0x11a3b2:    je      0x11a428
0x11a397:    movsxd  rdx, dword ptr [rip + 0x11746]
0x11a39e:    cmp     edx, dword ptr [rip + 0x11744]
0x11a3a4:    je      0x11a458
0x11a428:    mov     dword ptr [rip + 0x116c6], 0x32
0x11a432:    mov     edi, 0x190
0x11a437:    call    0x11d670
0x11a3b4:    mov     edx, dword ptr [rip + 0x1172a]
0x11a3ba:    lea     ecx, [rax - 1]
0x11a3bd:    cmp     ecx, edx
0x11a3bf:    je      0x11a4b0
0x11a458:    test    edx, edx
0x11a45a:    je      0x11a41d
0x11a43c:    mov     dword ptr [rip + 0x1169e], 1
0x11a446:    mov     qword ptr [rip + 0x116b3], rax
0x11a44d:    jmp     0x11a3ce
0x11a4b0:    add     eax, 0x32
0x11a4b3:    mov     rdi, qword ptr [rip + 0x11646]
0x11a4ba:    mov     dword ptr [rip + 0x11638], eax
0x11a4c0:    cdq     rax
0x11a4c2:    lea     rsi, [rax*8]
0x11a4ca:    call    0x11d690
0x11a3c5:    add     edx, 1
0x11a3c8:    mov     dword ptr [rip + 0x11716], edx
0x11a3ce:    mov     edi, 0x10
0x11a3d3:    call    0x11d670
0x11a41d:    add     rsp, 8
0x11a421:    pop     rbx
0x11a422:    pop     rbp
0x11a423:    ret
0x11a45c:    mov     rax, qword ptr [rip + 0x1169d]
0x11a463:    mov     rax, qword ptr [rax]
0x11a466:    test    rax, rax
0x11a469:    je      0x11a489
0x11a3ce:    mov     edi, 0x10
0x11a3d3:    call    0x11d670
0x11a4cf:    mov     edx, dword ptr [rip + 0x1160f]
0x11a4d5:    mov     qword ptr [rip + 0x11624], rax
0x11a4dc:    jmp     0x11a3c5
0x11a3d8:    mov     rdi, rbp

```

```

0x11a3db:    mov     rbx, rax
0x11a3de:    call   0x1049b0
0x11a489:    test    edx, edx
0x11a48b:    jle     0x11a4a1
0x11a46b:    mov     rdi, qword ptr [rax]
0x11a46e:    call   0x118d10
0x11a3e3:    lea     rdi, [rax + 1]
0x11a3e7:    call   0x11d670
0x11a4a1:    add     dword ptr [rip + 0x10340], 1
0x11a4a8:    jmp     0x11a3ce
0x11a48d:    mov     rdi, qword ptr [rip + 0x1166c]
0x11a494:    shl     rdx, 3
0x11a498:    lea     rsi, [rdi + 8]
0x11a49c:    call   0x104d40
0x11a473:    mov     rax, qword ptr [rip + 0x11686]
0x11a47a:    mov     rdi, qword ptr [rax]
0x11a47d:    call   0x118d10
0x11a3ec:    mov     rsi, rbp
0x11a3ef:    mov     rdi, rax
0x11a3f2:    call   0x1048b0
0x11a482:    movsxd  rdx, dword ptr [rip + 0x1165b]
0x11a489:    test    edx, edx
0x11a48b:    jle     0x11a4a1
0x11a3f7:    movsxd  rdx, dword ptr [rip + 0x116e6]
0x11a3fe:    mov     qword ptr [rbx + 8], 0
0x11a406:    mov     qword ptr [rbx], rax
0x11a409:    mov     rax, qword ptr [rip + 0x116f0]
0x11a410:    mov     qword ptr [rax + rdx*8], 0
0x11a418:    mov     qword ptr [rax + rdx*8 - 8], rbx
0x11a41d:    add     rsp, 8
0x11a421:    pop     rbx
0x11a422:    pop     rbp
0x11a423:    ret
0x11a424:    nop     dword ptr [rax]
0x11a452:    nop     word ptr [rax + rax]
0x11a4ad:    nop     dword ptr [rax]
0x11a4e1:    nop     word ptr cs:[rax + rax]
0x11a4ec:    nop     dword ptr [rax]
0x11a4f0:    endbr64
0x11a4f4:    push    r14
0x11a4f6:    xor     r14d, r14d
0x11a4f9:    push    r13
0x11a4fb:    mov     r13, rdx
0x11a4fe:    push    r12
0x11a500:    mov     r12, rsi
0x11a503:    push    rbp
0x11a504:    push    rbx
0x11a505:    movsxd  rbx, edi

```

```

0x11a508:    mov     edi, 0x10
0x11a50d:    call    0x11d670
0x11a512:    cmp     dword ptr [rip + 0x115cc], ebx
0x11a518:    jle     0x11a557
0x11a557:    pop     rbx
0x11a558:    mov     rax, r14
0x11a55b:    pop     rbp
0x11a55c:    pop     r12
0x11a55e:    pop     r13
0x11a560:    pop     r14
0x11a562:    ret
0x11a51a:    mov     rbp, rax
0x11a51d:    mov     rax, qword ptr [rip + 0x115dc]
0x11a524:    mov     rdi, r12
0x11a527:    mov     r14, qword ptr [rax + rbx*8]
0x11a52b:    call    0x1049b0
0x11a530:    lea     rdi, [rax + 1]
0x11a534:    call    0x11d670
0x11a539:    mov     rsi, r12
0x11a53c:    mov     rdi, rax
0x11a53f:    call    0x1048b0
0x11a544:    mov     qword ptr [rbp + 8], r13
0x11a548:    mov     qword ptr [rbp], rax
0x11a54c:    mov     rax, qword ptr [rip + 0x115ad]
0x11a553:    mov     qword ptr [rax + rbx*8], rbp
0x11a557:    pop     rbx
0x11a558:    mov     rax, r14
0x11a55b:    pop     rbp
0x11a55c:    pop     r12
0x11a55e:    pop     r13
0x11a560:    pop     r14
0x11a562:    ret
0x11a563:    nop     word ptr cs:[rax + rax]
0x11a56e:    nop
0x11a570:    endbr64
0x11a574:    mov     eax, dword ptr [rip + 0x11582]
0x11a57a:    ret
0x11a57b:    nop     dword ptr [rax + rax]
0x11a580:    endbr64
0x11a584:    xor     edx, edx
0x11a586:    xor     eax, eax
0x11a588:    jmp     0x119de0
0x11a58d:    nop     dword ptr [rax]
0x11a590:    endbr64
0x11a594:    mov     edx, 1
0x11a599:    xor     eax, eax
0x11a59b:    jmp     0x119de0
0x11a5a0:    endbr64

```

```

0x11a5a4:    push    r12
0x11a5a6:    push    rbx
0x11a5a7:    sub     rsp, 8
0x11a5ab:    mov     eax, dword ptr [rip + 0x11533]
0x11a5b1:    cmp     eax, edi
0x11a5b3:    jle     0x11a610
0x11a610:    xor     r12d, r12d
0x11a613:    add     rsp, 8
0x11a617:    mov     rax, r12
0x11a61a:    pop     rbx
0x11a61b:    pop     r12
0x11a61d:    ret
0x11a5b5:    test    eax, eax
0x11a5b7:    je      0x11a610
0x11a5b9:    lea     ebx, [rax - 1]
0x11a5bc:    movsxd  rdx, edi
0x11a5bf:    mov     rcx, qword ptr [rip + 0x1153a]
0x11a5c6:    mov     esi, ebx
0x11a5c8:    shl     rdx, 3
0x11a5cc:    sub     esi, edi
0x11a5ce:    cmp     eax, edi
0x11a5d0:    lea     r9, [rcx + rdx]
0x11a5d4:    mov     eax, 8
0x11a5d9:    lea     r8, [rsi*8 + 8]
0x11a5e1:    mov     r12, qword ptr [r9]
0x11a5e4:    mov     rdi, r9
0x11a5e7:    lea     rsi, [rcx + rdx + 8]
0x11a5ec:    cmovle  r8, rax
0x11a5f0:    mov     rdx, r8
0x11a5f3:    call    0x104d40
0x11a5f8:    mov     dword ptr [rip + 0x114e6], ebx
0x11a5fe:    add     rsp, 8
0x11a602:    mov     rax, r12
0x11a605:    pop     rbx
0x11a606:    pop     r12
0x11a608:    ret
0x11a609:    nop     dword ptr [rax]
0x11a61e:    nop
0x11a620:    endbr64
0x11a624:    push    r12
0x11a626:    mov     eax, dword ptr [rip + 0x114b8]
0x11a62c:    push    rbp
0x11a62d:    xor     ebp, ebp
0x11a62f:    test    edi, edi
0x11a631:    cmovns  ebp, edi
0x11a634:    push    rbx
0x11a635:    cmp     eax, ebp
0x11a637:    jle     0x11a6ce

```

```

0x11a6ce:    pop     rbx
0x11a6cf:    mov     dword ptr [rip + 0x11413], ebp
0x11a6d5:    mov     dword ptr [rip + 0x1140d], 1
0x11a6df:    pop     rbp
0x11a6e0:    pop     r12
0x11a6e2:    ret
0x11a63d:    sub     eax, ebp
0x11a63f:    mov     r12d, edi
0x11a642:    test    eax, eax
0x11a644:    jle     0x11a6e8
0x11a6e8:    xor     edx, edx
0x11a6ea:    jmp     0x11a686
0x11a64a:    xor     ebx, ebx
0x11a64c:    nop     dword ptr [rax]
0x11a650:    mov     rax, qword ptr [rip + 0x114a9]
0x11a657:    mov     rax, qword ptr [rax + rbx*8]
0x11a65b:    mov     rdi, qword ptr [rax]
0x11a65e:    call    0x118d10
0x11a686:    mov     dword ptr [rip + 0x1015c], edx
0x11a68c:    mov     rcx, qword ptr [rip + 0x1146d]
0x11a693:    test    r12d, r12d
0x11a696:    jle     0x11a6f0
0x11a663:    mov     rax, qword ptr [rip + 0x11496]
0x11a66a:    mov     rdi, qword ptr [rax + rbx*8]
0x11a66e:    call    0x118d10
0x11a6f0:    xor     eax, eax
0x11a6f2:    jmp     0x11a6c1
0x11a698:    cdqe
0x11a69a:    lea     rsi, [rcx + rax*8]
0x11a69e:    xor     eax, eax
0x11a6a0:    mov     rdx, qword ptr [rsi + rax*8]
0x11a6a4:    mov     qword ptr [rcx + rax*8], rdx
0x11a6a8:    add     rax, 1
0x11a6ac:    cmp     ebp, eax
0x11a6ae:    jg      0x11a6a0
0x11a673:    mov     eax, dword ptr [rip + 0x1146b]
0x11a679:    lea     edx, [rbx + 1]
0x11a67c:    add     rbx, 1
0x11a680:    sub     eax, ebp
0x11a682:    cmp     eax, ebx
0x11a684:    jg      0x11a650
0x11a6c1:    mov     qword ptr [rcx], 0
0x11a6c8:    mov     dword ptr [rip + 0x11416], eax
0x11a6ce:    pop     rbx
0x11a6cf:    mov     dword ptr [rip + 0x11413], ebp
0x11a6d5:    mov     dword ptr [rip + 0x1140d], 1
0x11a6df:    pop     rbp
0x11a6e0:    pop     r12

```

```

0x11a6e2:    ret
0x11a6b0:    test     ebp, ebp
0x11a6b2:    mov      eax, 1
0x11a6b7:    cmovg    eax, ebp
0x11a6ba:    movsxd   rdx, eax
0x11a6bd:    lea      rcx, [rcx + rdx*8]
0x11a6c1:    mov      qword ptr [rcx], 0
0x11a6c8:    mov      dword ptr [rip + 0x11416], eax
0x11a6ce:    pop      rbx
0x11a6cf:    mov      dword ptr [rip + 0x11413], ebp
0x11a6d5:    mov      dword ptr [rip + 0x1140d], 1
0x11a6df:    pop      rbp
0x11a6e0:    pop      r12
0x11a6e2:    ret
0x11a6a0:    mov      rdx, qword ptr [rsi + rax*8]
0x11a6a4:    mov      qword ptr [rcx + rax*8], rdx
0x11a6a8:    add      rax, 1
0x11a6ac:    cmp      ebp, eax
0x11a6ae:    jg        0x11a6a0
0x11a650:    mov      rax, qword ptr [rip + 0x114a9]
0x11a657:    mov      rax, qword ptr [rax + rbx*8]
0x11a65b:    mov      rdi, qword ptr [rax]
0x11a65e:    call     0x118d10
0x11a6e3:    nop      dword ptr [rax + rax]
0x11a6ec:    nop      dword ptr [rax]
0x11a6f4:    nop      word ptr cs:[rax + rax]
0x11a6ff:    nop
0x11a700:    endbr64
0x11a704:    mov      edx, dword ptr [rip + 0x113e2]
0x11a70a:    mov      eax, dword ptr [rip + 0x113d8]
0x11a710:    test     edx, edx
0x11a712:    je       0x11a720
0x11a720:    ret
0x11a714:    mov      dword ptr [rip + 0x113ce], 0
0x11a71e:    neg      eax
0x11a720:    ret
0x11a721:    nop      word ptr cs:[rax + rax]
0x11a72c:    nop      dword ptr [rax]
0x11a730:    endbr64
0x11a734:    push     r15
0x11a736:    push     r14
0x11a738:    mov      r14d, esi
0x11a73b:    push     r13
0x11a73d:    push     r12
0x11a73f:    mov      r12d, edx
0x11a742:    push     rbp
0x11a743:    push     rbx
0x11a744:    sub      rsp, 0xa8

```

```

0x11a74b:    mov     rax, qword ptr fs:[0x28]
0x11a754:    mov     qword ptr [rsp + 0x98], rax
0x11a75c:    xor     eax, eax
0x11a75e:    call    0x11a080
0x11a763:    xor     esi, esi
0x11a765:    mov     edx, 0x1b6
0x11a76a:    mov     rbp, rax
0x11a76d:    mov     rdi, rax
0x11a770:    xor     eax, eax
0x11a772:    call    0x104d80
0x11a777:    test    eax, eax
0x11a779:    js      0x11a80f
0x11a80f:    mov     rdi, rbp
0x11a812:    call    0x118d10
0x11a77f:    mov     rsi, rsp
0x11a782:    mov     edi, eax
0x11a784:    mov     r15d, eax
0x11a787:    call    0x118ef0
0x11a817:    jmp     0x11a7d7
0x11a78c:    cmp     eax, -1
0x11a78f:    je      0x11a807
0x11a7d7:    call    0x104860
0x11a807:    mov     edi, r15d
0x11a80a:    call    0x104ab0
0x11a791:    mov     eax, dword ptr [rsp + 0x30]
0x11a795:    lea     edi, [rax + 1]
0x11a798:    movsxd  rdi, edi
0x11a79b:    call    0x11d670
0x11a7dc:    mov     eax, dword ptr [rax]
0x11a7de:    mov     rdx, qword ptr [rsp + 0x98]
0x11a7e6:    sub     rdx, qword ptr fs:[0x28]
0x11a7ef:    jne     0x11a8f0
0x11a7a0:    mov     rdx, qword ptr [rsp + 0x30]
0x11a7a5:    mov     edi, r15d
0x11a7a8:    mov     rsi, rax
0x11a7ab:    mov     r13, rax
0x11a7ae:    call    0x1047a0
0x11a7f5:    add     rsp, 0xa8
0x11a7fc:    pop     rbx
0x11a7fd:    pop     rbp
0x11a7fe:    pop     r12
0x11a800:    pop     r13
0x11a802:    pop     r14
0x11a804:    pop     r15
0x11a806:    ret
0x11a8f0:    call    0x1049d0
0x11a7b3:    cmp     qword ptr [rsp + 0x30], rax
0x11a7b8:    je      0x11a819

```

```

0x11a819:    mov     edi, r15d
0x11a81c:    call    0x104ab0
0x11a7ba:    mov     edi, r15d
0x11a7bd:    call    0x104ab0
0x11a821:    mov     rcx, qword ptr [rsp + 0x30]
0x11a826:    test    r12d, r12d
0x11a829:    jns     0x11a82e
0x11a7c2:    mov     rdi, rbp
0x11a7c5:    call    0x118d10
0x11a82b:    mov     r12d, ecx
0x11a82e:    xor     ebx, ebx
0x11a830:    xor     esi, esi
0x11a832:    movsxd  rdx, esi
0x11a835:    cmp     ebx, r14d
0x11a838:    jge     0x11a885
0x11a82e:    xor     ebx, ebx
0x11a830:    xor     esi, esi
0x11a832:    movsxd  rdx, esi
0x11a835:    cmp     ebx, r14d
0x11a838:    jge     0x11a885
0x11a7ca:    test    r13, r13
0x11a7cd:    je      0x11a7d7
0x11a83a:    cmp     rdx, rcx
0x11a83d:    jge     0x11a885
0x11a885:    lea     r14, [r13 + rdx]
0x11a88a:    lea     r15d, [rsi + 1]
0x11a88e:    cmp     rcx, rdx
0x11a891:    jg      0x11a8aa
0x11a7cf:    mov     rdi, r13
0x11a7d2:    call    0x118d10
0x11a83f:    lea     eax, [rsi + 1]
0x11a842:    cdqe
0x11a844:    cmp     rdx, rcx
0x11a847:    jl      0x11a85c
0x11a893:    jmp     0x11a8d4
0x11a8aa:    cmp     byte ptr [r14], 0xa
0x11a8ae:    jne     0x11a898
0x11a85c:    cmp     byte ptr [r13 + rax - 1], 0xa
0x11a862:    jne     0x11a850
0x11a849:    jmp     0x11a87b
0x11a8d4:    mov     rdi, rbp
0x11a8d7:    call    0x118d10
0x11a8b0:    lea     rdi, [r13 + rsi]
0x11a8b5:    mov     byte ptr [r14], 0
0x11a8b9:    cmp     byte ptr [rdi], 0
0x11a8bc:    jne     0x11a8cb
0x11a898:    movsxd  rax, r15d
0x11a89b:    add     r14, 1

```



```

0x11a89f:      add     r15d, 1
0x11a8a3:      cmp     rax, qword ptr [rsp + 0x30]
0x11a8a8:      jge     0x11a8d4
0x11a864:      add     ebx, 1
0x11a867:      movsxd   rsi, eax
0x11a86a:      cmp     r14d, ebx
0x11a86d:      je      0x11a832
0x11a850:      lea     rdx, [rax + 1]
0x11a854:      cmp     rax, rcx
0x11a857:      je      0x11a878
0x11a87b:      cmp     rcx, rdx
0x11a87e:      jle     0x11a885
0x11a8dc:      test    r13, r13
0x11a8df:      je      0x11a8e9
0x11a8be:      add     ebx, 1
0x11a8c1:      cmp     r12d, ebx
0x11a8c4:      jle     0x11a8d4
0x11a8cb:      xor     eax, eax
0x11a8cd:      call    0x11a380
0x11a832:      movsxd   rdx, esi
0x11a835:      cmp     ebx, r14d
0x11a838:      jge     0x11a885
0x11a86f:      lea     rdx, [rax + 1]
0x11a873:      cmp     rax, rcx
0x11a876:      jne     0x11a859
0x11a878:      movsxd   rdx, esi
0x11a87b:      cmp     rcx, rdx
0x11a87e:      jle     0x11a885
0x11a859:      mov     rax, rdx
0x11a85c:      cmp     byte ptr [r13 + rax - 1], 0xa
0x11a862:      jne     0x11a850
0x11a880:      cmp     r14d, ebx
0x11a883:      jg      0x11a83f
0x11a8e9:      xor     eax, eax
0x11a8eb:      jmp     0x11a7de
0x11a8e1:      mov     rdi, r13
0x11a8e4:      call    0x118d10
0x11a8c6:      movsxd   rsi, r15d
0x11a8c9:      jmp     0x11a898
0x11a8d2:      jmp     0x11a8be
0x11a7de:      mov     rdx, qword ptr [rsp + 0x98]
0x11a7e6:      sub     rdx, qword ptr fs:[0x28]
0x11a7ef:      jne     0x11a8f0
0x11a84b:      nop     dword ptr [rax + rax]
0x11a895:      nop     dword ptr [rax]
0x11a8f5:      nop     word ptr cs:[rax + rax]
0x11a900:      endbr64
0x11a904:      mov     edx, 0xffffffff

```

```

0x11a909:    xor     esi, esi
0x11a90b:    xor     eax, eax
0x11a90d:    jmp     0x11a730
0x11a912:    nop     word ptr cs:[rax + rax]
0x11a91d:    nop     dword ptr [rax]
0x11a920:    endbr64
0x11a924:    push    r14
0x11a926:    push    r13
0x11a928:    push    r12
0x11a92a:    mov     r12d, esi
0x11a92d:    push    rbp
0x11a92e:    push    rbx
0x11a92f:    sub     rsp, 0xa0
0x11a936:    mov     rax, qword ptr fs:[0x28]
0x11a93f:    mov     qword ptr [rsp + 0x98], rax
0x11a947:    xor     eax, eax
0x11a949:    call    0x11a080
0x11a94e:    xor     esi, esi
0x11a950:    mov     edx, 0x1b6
0x11a955:    mov     rbp, rax
0x11a958:    mov     rdi, rax
0x11a95b:    xor     eax, eax
0x11a95d:    call    0x104d80
0x11a962:    cmp     eax, -1
0x11a965:    je      0x11aa58
0x11aa58:    mov     rdi, rbp
0x11aa5b:    call    0x118d10
0x11a96b:    mov     rsi, rsp
0x11a96e:    mov     edi, eax
0x11a970:    mov     r14d, eax
0x11a973:    call    0x118ef0
0x11aa60:    mov     rax, qword ptr [rsp + 0x98]
0x11aa68:    sub     rax, qword ptr fs:[0x28]
0x11aa71:    jne     0x11aa8f
0x11a978:    cmp     eax, -1
0x11a97b:    je      0x11aa58
0x11aa73:    add     rsp, 0xa0
0x11aa7a:    xor     eax, eax
0x11aa7c:    pop     rbx
0x11aa7d:    pop     rbp
0x11aa7e:    pop     r12
0x11aa80:    pop     r13
0x11aa82:    pop     r14
0x11aa84:    ret
0x11aa8f:    call    0x1049d0
0x11a981:    mov     eax, dword ptr [rsp + 0x30]
0x11a985:    lea     edi, [rax + 1]
0x11a988:    movsxd  rdi, edi

```

```

0x11a98b:    call    0x11d670
0x11a990:    mov     rdx, qword ptr [rsp + 0x30]
0x11a995:    mov     edi, r14d
0x11a998:    mov     rsi, rax
0x11a99b:    mov     r13, rax
0x11a99e:    call    0x1047a0
0x11a9a3:    mov     edi, r14d
0x11a9a6:    mov     rbx, rax
0x11a9a9:    call    0x104ab0
0x11a9ae:    test    ebx, ebx
0x11a9b0:    jle     0x11aa88
0x11aa88:    test    r13, r13
0x11aa8b:    je      0x11aa58
0x11a9b6:    lea     eax, [rbx - 1]
0x11a9b9:    cmp     ebx, 1
0x11a9bc:    je      0x11a9f3
0x11aa8d:    jmp     0x11aa50
0x11a9f3:    test    eax, eax
0x11a9f5:    je      0x11aa88
0x11a9be:    test    r12d, r12d
0x11a9c1:    je      0x11a9f3
0x11aa50:    mov     rdi, r13
0x11aa53:    call    0x118d10
0x11a9fb:    cdqe
0x11a9fd:    jmp     0x11aa08
0x11a9c3:    lea     edx, [rbx - 2]
0x11a9c6:    movsxd  rdx, edx
0x11a9c9:    nop     dword ptr [rax]
0x11a9d0:    cmp     byte ptr [r13 + rdx + 1], 0xa
0x11a9d6:    jne     0x11a9dc
0x11aa08:    cmp     byte ptr [r13 + rax], 0xa
0x11aa0e:    mov     ebx, eax
0x11aa10:    jne     0x11aa00
0x11a9d8:    sub     r12d, 1
0x11a9dc:    test    r12d, r12d
0x11a9df:    mov     eax, edx
0x11a9e1:    setne   sil
0x11a9e5:    test    edx, edx
0x11a9e7:    setne   cl
0x11a9ea:    sub     rdx, 1
0x11a9ee:    test    sil, cl
0x11a9f1:    jne     0x11a9d0
0x11a9dc:    test    r12d, r12d
0x11a9df:    mov     eax, edx
0x11a9e1:    setne   sil
0x11a9e5:    test    edx, edx
0x11a9e7:    setne   cl
0x11a9ea:    sub     rdx, 1

```

```

0x11a9ee:    test    sil, cl
0x11a9f1:    jne     0x11a9d0
0x11aa12:    mov     edx, 0x1b6
0x11aa17:    mov     esi, 0x201
0x11aa1c:    mov     rdi, rbp
0x11aa1f:    xor     eax, eax
0x11aa21:    call    0x104d80
0x11aa00:    sub     rax, 1
0x11aa04:    test    eax, eax
0x11aa06:    je      0x11aa50
0x11a9d0:    cmp     byte ptr [r13 + rdx + 1], 0xa
0x11a9d6:    jne     0x11a9dc
0x11aa26:    mov     r12d, eax
0x11aa29:    cmp     eax, -1
0x11aa2c:    je      0x11aa50
0x11aa2e:    mov     rdx, qword ptr [rsp + 0x30]
0x11aa33:    lea     esi, [rbx + 1]
0x11aa36:    mov     edi, eax
0x11aa38:    movsxd  rsi, esi
0x11aa3b:    sub     rdx, rsi
0x11aa3e:    add     rsi, r13
0x11aa41:    call    0x104790
0x11aa46:    mov     edi, r12d
0x11aa49:    call    0x104ab0
0x11aa4e:    nop
0x11a9ff:    nop
0x11aa85:    nop     dword ptr [rax]
0x11aa94:    nop     word ptr cs:[rax + rax]
0x11aa9f:    nop
0x11aaa0:    endbr64
0x11aaa4:    mov     r8d, edi
0x11aaa7:    xor     edx, edx
0x11aaa9:    mov     rdi, rsi
0x11aaac:    xor     eax, eax
0x11aaae:    mov     esi, r8d
0x11aab1:    jmp     0x11a130
0x11aab6:    nop     word ptr cs:[rax + rax]
0x11aac0:    endbr64
0x11aac4:    mov     esi, dword ptr [rip + 0x1101a]
0x11aaca:    mov     edx, 1
0x11aacf:    xor     eax, eax
0x11aad1:    jmp     0x11a130
0x11aad6:    nop     word ptr cs:[rax + rax]
0x11aae0:    endbr64
0x11aae4:    movsxd  rdx, dword ptr [rip + 0x11011]
0x11aaeb:    cmp     edx, dword ptr [rip + 0x10ff3]
0x11aaf1:    je      0x11ab08
0x11ab08:    xor     eax, eax

```

```

0x11ab0a:    ret
0x11aaf3:    mov     rax, qword ptr [rip + 0x11006]
0x11aafa:    test    rax, rax
0x11aafd:    je      0x11ab0a
0x11ab0a:    ret
0x11aaff:    mov     rax, qword ptr [rax + rdx*8]
0x11ab03:    ret
0x11ab04:    nop     dword ptr [rax]
0x11ab0b:    nop     dword ptr [rax + rax]
0x11ab10:    endbr64
0x11ab14:    mov     eax, dword ptr [rip + 0x10fe2]
0x11ab1a:    xor     r8d, r8d
0x11ab1d:    test    eax, eax
0x11ab1f:    je      0x11ab37
0x11ab37:    mov     rax, r8
0x11ab3a:    ret
0x11ab21:    sub     eax, 1
0x11ab24:    mov     rdx, qword ptr [rip + 0x10fd5]
0x11ab2b:    mov     dword ptr [rip + 0x10fcb], eax
0x11ab31:    cdqe
0x11ab33:    mov     r8, qword ptr [rdx + rax*8]
0x11ab37:    mov     rax, r8
0x11ab3a:    ret
0x11ab3b:    nop     dword ptr [rax + rax]
0x11ab40:    endbr64
0x11ab44:    mov     eax, dword ptr [rip + 0x10fb2]
0x11ab4a:    cmp     eax, dword ptr [rip + 0x10f94]
0x11ab50:    je      0x11ab70
0x11ab70:    xor     eax, eax
0x11ab72:    ret
0x11ab52:    add     eax, 1
0x11ab55:    mov     rdx, qword ptr [rip + 0x10fa4]
0x11ab5c:    mov     dword ptr [rip + 0x10f9a], eax
0x11ab62:    cdqe
0x11ab64:    mov     rax, qword ptr [rdx + rax*8]
0x11ab68:    ret
0x11ab69:    nop     dword ptr [rax]
0x11ab73:    nop     word ptr cs:[rax + rax]
0x11ab7e:    nop
0x11ab80:    endbr64
0x11ab84:    mov     rax, qword ptr [rip + 0x10f75]
0x11ab8b:    ret
0x11ab8c:    nop     dword ptr [rax]
0x11ab90:    endbr64
0x11ab94:    sub     edi, dword ptr [rip + 0xfc4e]
0x11ab9a:    cmp     dword ptr [rip + 0x10f44], edi
0x11aba0:    jle     0x11abc0
0x11abc0:    xor     eax, eax

```

```

0x11abc2:    ret
0x11aba2:    test     edi, edi
0x11aba4:    js       0x11abc0
0x11aba6:    mov      rax, qword ptr [rip + 0x10f53]
0x11abad:    test     rax, rax
0x11abb0:    je       0x11abc2
0x11abc2:    ret
0x11abb2:    movsxd   rdi, edi
0x11abb5:    mov      rax, qword ptr [rax + rdi*8]
0x11abb9:    ret
0x11abba:    nop      word ptr [rax + rax]
0x11abc3:    nop      word ptr cs:[rax + rax]
0x11abce:    nop
0x11abd0:    endbr64
0x11abd4:    cmp      edx, dword ptr [rip + 0x10f0a]
0x11abda:    push     rbx
0x11abdb:    mov      ebx, dword ptr [rip + 0x10f1b]
0x11abe1:    jg       0x11abf7
0x11abe3:    test     edx, edx
0x11abe5:    js       0x11abf7
0x11abf7:    xor      edx, edx
0x11abf9:    xor      eax, eax
0x11abfb:    call     0x119de0
0x11abe7:    cmp      qword ptr [rip + 0x10f11], 0
0x11abef:    je       0x11abf7
0x11ac00:    mov      edx, ebx
0x11ac02:    shr      edx, 0x1f
0x11ac05:    cmp      eax, -1
0x11ac08:    je       0x11ac10
0x11abf1:    mov      dword ptr [rip + 0x10f05], edx
0x11abf7:    xor      edx, edx
0x11abf9:    xor      eax, eax
0x11abfb:    call     0x119de0
0x11ac10:    cmp      ebx, dword ptr [rip + 0x10ece]
0x11ac16:    jg       0x11ac2c
0x11ac0a:    mov      eax, dword ptr [rip + 0x10eec]
0x11ac10:    cmp      ebx, dword ptr [rip + 0x10ece]
0x11ac16:    jg       0x11ac2c
0x11ac18:    test     dl, dl
0x11ac1a:    jne      0x11ac2c
0x11ac2c:    pop      rbx
0x11ac2d:    ret
0x11ac1c:    cmp      qword ptr [rip + 0x10edc], 0
0x11ac24:    je       0x11ac2c
0x11ac26:    mov      dword ptr [rip + 0x10ed0], ebx
0x11ac2c:    pop      rbx
0x11ac2d:    ret
0x11ac2e:    nop

```

```

0x11ac30:    endbr64
0x11ac34:    cmp     dword ptr [rip + 0x10eaa], edi
0x11ac3a:    jl      0x11ac60
0x11ac60:    xor     eax, eax
0x11ac62:    ret
0x11ac3c:    test    edi, edi
0x11ac3e:    js      0x11ac60
0x11ac40:    cmp     qword ptr [rip + 0x10eb8], 0
0x11ac48:    je      0x11ac60
0x11ac4a:    mov     dword ptr [rip + 0x10eac], edi
0x11ac50:    mov     eax, 1
0x11ac55:    ret
0x11ac56:    nop     word ptr cs:[rax + rax]
0x11ac63:    nop     word ptr cs:[rax + rax]
0x11ac6e:    nop
0x11ac70:    endbr64
0x11ac74:    movsxd  rax, dword ptr [rsi]
0x11ac77:    movzx   r8d, byte ptr [rip + 0xfb6e]
0x11ac7f:    cmp     byte ptr [rdi + rax], r8b
0x11ac83:    jne     0x11afd0
0x11ac89:    push    r15
0x11ac8b:    lea     ecx, [rax + 1]
0x11ac8e:    mov     r9d, edx
0x11ac91:    mov     rdx, rax
0x11ac94:    push    r14
0x11ac96:    movsxd  r10, ecx
0x11ac99:    push    r13
0x11ac9b:    mov     r13, rsi
0x11ac9e:    lea     rsi, [rdi + r10]
0x11aca2:    push    r12
0x11aca4:    push    rbp
0x11aca5:    push    rbx
0x11aca6:    sub     rsp, 0x18
0x11acaa:    movzx   eax, byte ptr [rsi]
0x11acad:    cmp     r8b, al
0x11acb0:    je      0x11af20
0x11afd0:    xor     eax, eax
0x11afd2:    ret
0x11af20:    mov     ecx, dword ptr [rip + 0x10bbe]
0x11af26:    mov     eax, dword ptr [rip + 0xf8bc]
0x11af2c:    add     edx, 2
0x11af2f:    mov     dword ptr [r13], edx
0x11af33:    lea     eax, [rcx + rax - 1]
0x11af37:    sub     eax, dword ptr [rip + 0xf8ab]
0x11af3d:    cmp     eax, dword ptr [rip + 0x10ba1]
0x11af43:    jge     0x11adfc
0x11acb6:    mov     r8d, 1
0x11acbc:    cmp     al, 0x2d

```

```

0x11acbe:    je      0x11aef8
0x11af49:    test   eax, eax
0x11af4b:    js      0x11adfc
0x11adfc:    add     rsp, 0x18
0x11ae00:    xor     eax, eax
0x11ae02:    pop     rbx
0x11ae03:    pop     rbp
0x11ae04:    pop     r12
0x11ae06:    pop     r13
0x11ae08:    pop     r14
0x11ae0a:    pop     r15
0x11ae0c:    ret
0x11aef8:    lea     ecx, [rdx + 2]
0x11aefb:    mov     r8d, 0xffffffff
0x11af01:    movsxd  r10, ecx
0x11af04:    lea     rsi, [rdi + r10]
0x11af08:    movzx   eax, byte ptr [rsi]
0x11af0b:    lea     edx, [rax - 0x30]
0x11af0e:    cmp     dl, 9
0x11af11:    ja      0x11acd0
0x11acc4:    lea     edx, [rax - 0x30]
0x11acc7:    cmp     dl, 9
0x11acca:    jbe     0x11ae68
0x11af51:    mov     rdx, qword ptr [rip + 0x10ba8]
0x11af58:    test    rdx, rdx
0x11af5b:    je      0x11adfc
0x11af17:    jmp     0x11ae68
0x11acd0:    mov     ebp, 1
0x11acd5:    xor     ebx, ebx
0x11acd7:    cmp     al, 0x3f
0x11acd9:    jne     0x11acef
0x11ae68:    lea     edx, [rcx + 1]
0x11ae6b:    xor     ecx, ecx
0x11ae6d:    movsxd  rdx, edx
0x11ae70:    sub     eax, 0x30
0x11ae73:    lea     ecx, [rcx + rcx*4]
0x11ae76:    mov     r9d, edx
0x11ae79:    movsx   eax, al
0x11ae7c:    lea     ecx, [rax + rcx*2]
0x11ae7f:    movzx   eax, byte ptr [rdi + rdx]
0x11ae83:    add     rdx, 1
0x11ae87:    lea     esi, [rax - 0x30]
0x11ae8a:    cmp     sil, 9
0x11ae8e:    jbe     0x11ae70
0x11af61:    cdqe
0x11af63:    mov     rax, qword ptr [rdx + rax*8]
0x11af67:    test    rax, rax
0x11af6a:    jne     0x11aedef

```



```

0x11acdb:      add     ecx, 1
0x11acde:      xor     ebp, ebp
0x11ace0:      mov     ebx, 1
0x11ace5:      movsxd  r10, ecx
0x11ace8:      lea     rsi, [rdi + r10]
0x11acec:      movzx   eax, byte ptr [rsi]
0x11acef:      test    al, al
0x11acf1:      je      0x11b042
0x11acef:      test    al, al
0x11acf1:      je      0x11b042
0x11ae70:      sub     eax, 0x30
0x11ae73:      lea     ecx, [rcx + rcx*4]
0x11ae76:      mov     r9d, edx
0x11ae79:      movsx   eax, al
0x11ae7c:      lea     ecx, [rax + rcx*2]
0x11ae7f:      movzx   eax, byte ptr [rdi + rdx]
0x11ae83:      add     rdx, 1
0x11ae87:      lea     esi, [rax - 0x30]
0x11ae8a:      cmp     sil, 9
0x11ae8e:      jbe     0x11ae70
0x11ae90:      mov     dword ptr [r13], r9d
0x11ae94:      mov     eax, dword ptr [rip + 0x10c4a]
0x11ae9a:      mov     edx, dword ptr [rip + 0xf948]
0x11aea0:      cmp     r8d, -1
0x11aea4:      jne     0x11aead
0x11af70:      jmp     0x11adfc
0x11aedf:      mov     rax, qword ptr [rax]
0x11aee2:      add     rsp, 0x18
0x11aee6:      pop     rbx
0x11aee7:      pop     rbp
0x11aee8:      pop     r12
0x11aeea:      pop     r13
0x11aeec:      pop     r14
0x11aeef:      pop     r15
0x11aef0:      ret
0x11b042:      mov     r12d, ecx
0x11b045:      mov     r8, rsi
0x11b048:      xor     r15d, r15d
0x11b04b:      mov     edi, 1
0x11b050:      jmp     0x11ae20
0x11acf7:      lea     rdx, [rdi + r10 + 1]
0x11acfc:      mov     r12d, ecx
0x11acff:      mov     r8, rsi
0x11ad02:      movabs  r10, 0x400000100000200
0x11ad0c:      jmp     0x11ad48
0x11aea6:      lea     esi, [rax + rdx]
0x11aea9:      sub     esi, ecx
0x11aeab:      mov     ecx, esi

```

```

0x11aead:    sub     ecx, edx
0x11aeaf:    cmp     ecx, eax
0x11aeb1:    jge     0x11adfc
0x11aead:    sub     ecx, edx
0x11aeaf:    cmp     ecx, eax
0x11aeb1:    jge     0x11adfc
0x11ae20:    mov     qword ptr [rsp + 8], r8
0x11ae25:    mov     qword ptr [rsp], rsi
0x11ae29:    call    0x11d670
0x11ad48:    test    ebx, ebx
0x11ad4a:    je      0x11ad10
0x11aeb7:    test    ecx, ecx
0x11aeb9:    js      0x11adfc
0x11ae2e:    mov     rsi, qword ptr [rsp]
0x11ae32:    mov     rdx, r15
0x11ae35:    mov     rdi, rax
0x11ae38:    mov     r14, rax
0x11ae3b:    call    0x104880
0x11ad10:    cmp     al, 0x3a
0x11ad12:    ja      0x11ad1e
0x11ad4c:    cmp     al, 0xa
0x11ad4e:    je      0x11ae10
0x11aebf:    mov     rax, qword ptr [rip + 0x10c3a]
0x11aec6:    test    rax, rax
0x11aec9:    je      0x11adfc
0x11ae40:    test    ebx, ebx
0x11ae42:    mov     byte ptr [r14 + r15], 0
0x11ae47:    mov     r8, qword ptr [rsp + 8]
0x11ae4c:    je      0x11ada0
0x11ad14:    bt      r10, rax
0x11ad18:    jb      0x11afd8
0x11ad1e:    movsx   edi, al
0x11ad21:    cmp     edi, r9d
0x11ad24:    je      0x11b010
0x11ae10:    mov     r15d, r12d
0x11ae13:    sub     r15d, ecx
0x11ae16:    lea     edi, [r15 + 1]
0x11ae1a:    movsxd  r15, r15d
0x11ae1d:    movsxd  rdi, edi
0x11ae20:    mov     qword ptr [rsp + 8], r8
0x11ae25:    mov     qword ptr [rsp], rsi
0x11ae29:    call    0x11d670
0x11ad54:    cmp     al, 0x3f
0x11ad56:    jne     0x11ad32
0x11aecf:    movsxd  rcx, ecx
0x11aed2:    mov     rax, qword ptr [rax + rcx*8]
0x11aed6:    test    rax, rax
0x11aed9:    je      0x11adfc

```

```

0x11ada0:    mov     dword ptr [r13], r12d
0x11ada4:    mov     r12d, ebx
0x11ada7:    and     r12d, 1
0x11adab:    jmp     0x11add3
0x11ae52:    xor     eax, eax
0x11ae54:    cmp     byte ptr [r8], 0x3f
0x11ae58:    sete    al
0x11ae5b:    add     r12d, eax
0x11ae5e:    jmp     0x11ada0
0x11afd8:    mov     eax, r12d
0x11afdb:    mov     qword ptr [rsp], rsi
0x11afdf:    sub     eax, ecx
0x11afe1:    movsxd  r15, eax
0x11afe4:    add     eax, 1
0x11afe7:    movsxd  rdi, eax
0x11afea:    call    0x11d670
0x11b010:    mov     r15d, r12d
0x11b013:    mov     qword ptr [rsp], rsi
0x11b017:    sub     r15d, ecx
0x11b01a:    lea     edi, [r15 + 1]
0x11b01e:    movsxd  r15, r15d
0x11b021:    movsxd  rdi, edi
0x11b024:    call    0x11d670
0x11ad2a:    cmp     al, 0xa
0x11ad2c:    je      0x11ae10
0x11ad58:    mov     r15d, r12d
0x11ad5b:    mov     qword ptr [rsp + 8], r8
0x11ad60:    sub     r15d, ecx
0x11ad63:    mov     qword ptr [rsp], rsi
0x11ad67:    lea     edi, [r15 + 1]
0x11ad6b:    movsxd  r15, r15d
0x11ad6e:    movsxd  rdi, edi
0x11ad71:    call    0x11d670
0x11ad32:    mov     r8, rdx
0x11ad35:    movzx   eax, byte ptr [rdx]
0x11ad38:    lea     rdx, [rdx + 1]
0x11ad3c:    add     r12d, 1
0x11ad40:    test    al, al
0x11ad42:    je      0x11ae10
0x11add3:    xor     eax, eax
0x11add5:    mov     edx, ebp
0x11add7:    mov     esi, 0xffffffff
0x11addc:    mov     rdi, r14
0x11addf:    call    0x119de0
0x11afef:    mov     r14, rax
0x11aff2:    mov     rsi, qword ptr [rsp]
0x11aff6:    mov     rdx, r15
0x11aff9:    mov     rdi, r14

```

```

0x11affc:      call    0x104880
0x11b029:      mov     r14, rax
0x11b02c:      jmp     0x11aff2
0x11ad76:      mov     rsi, qword ptr [rsp]
0x11ad7a:      mov     rdx, r15
0x11ad7d:      mov     rdi, rax
0x11ad80:      mov     r14, rax
0x11ad83:      call    0x104880
0x11ade4:      test    eax, eax
0x11ade6:      jns     0x11adb0
0x11b001:      mov     byte ptr [r14 + r15], 0
0x11b006:      jmp     0x11ada0
0x11aff2:      mov     rsi, qword ptr [rsp]
0x11aff6:      mov     rdx, r15
0x11aff9:      mov     rdi, r14
0x11affc:      call    0x104880
0x11ad88:      mov     r8, qword ptr [rsp + 8]
0x11ad8d:      xor     eax, eax
0x11ad8f:      mov     byte ptr [r14 + r15], 0
0x11ad94:      cmp     byte ptr [r8], 0x3f
0x11ad98:      sete    al
0x11ad9b:      add     r12d, eax
0x11ad9e:      nop
0x11ada0:      mov     dword ptr [r13], r12d
0x11ada4:      mov     r12d, ebx
0x11ada7:      and     r12d, 1
0x11adab:      jmp     0x11add3
0x11ade8:      mov     eax, dword ptr [rip + 0x10cf6]
0x11adee:      mov     rdi, r14
0x11adf1:      mov     dword ptr [rip + 0x10d05], eax
0x11adf7:      call    0x118d10
0x11adb0:      movsxd  rax, dword ptr [rip + 0x10d45]
0x11adb7:      je      0x11af78
0x11af78:      mov     edx, dword ptr [rip + 0x10b66]
0x11af7e:      cmp     edx, eax
0x11af80:      je      0x11b03b
0x11adbd:      test    r12b, r12b
0x11adc0:      jne     0x11af78
0x11b03b:      xor     ebp, ebp
0x11b03d:      jmp     0x11af97
0x11af86:      mov     rbp, qword ptr [rip + 0x10b73]
0x11af8d:      test    rbp, rbp
0x11af90:      je      0x11af97
0x11adc6:      test    eax, eax
0x11adc8:      je      0x11ade8
0x11af97:      mov     dword ptr [rip + 0x10b5f], edx
0x11af9d:      test    ebx, ebx
0x11af9f:      je      0x11b02e

```

```

0x11af92:    mov     rbp, qword ptr [rbp + rax*8]
0x11af97:    mov     dword ptr [rip + 0x10b5f], edx
0x11af9d:    test    ebx, ebx
0x11af9f:    je      0x11b02e
0x11adca:    sub     eax, 1
0x11adcd:    mov     dword ptr [rip + 0x10d29], eax
0x11add3:    xor     eax, eax
0x11add5:    mov     edx, ebp
0x11add7:    mov     esi, 0xffffffff
0x11addc:    mov     rdi, r14
0x11addf:    call    0x119de0
0x11b02e:    mov     rdi, r14
0x11b031:    call    0x118d10
0x11afa5:    mov     rdi, qword ptr [rip + 0x10b44]
0x11afac:    test    rdi, rdi
0x11afaf:    je      0x11afb6
0x11b036:    jmp     0x11afbd
0x11afb6:    mov     qword ptr [rip + 0x10b33], r14
0x11afbd:    mov     rax, qword ptr [rbp]
0x11afc1:    add     rsp, 0x18
0x11afc5:    pop     rbx
0x11afc6:    pop     rbp
0x11afc7:    pop     r12
0x11afc9:    pop     r13
0x11afcb:    pop     r14
0x11afcd:    pop     r15
0x11afcf:    ret
0x11afb1:    call    0x118d10
0x11afbd:    mov     rax, qword ptr [rbp]
0x11afc1:    add     rsp, 0x18
0x11afc5:    pop     rbx
0x11afc6:    pop     rbp
0x11afc7:    pop     r12
0x11afc9:    pop     r13
0x11afcb:    pop     r14
0x11afcd:    pop     r15
0x11afcf:    ret
0x11ad0e:    nop
0x11adad:    nop     dword ptr [rax]
0x11ae0d:    nop     dword ptr [rax]
0x11ae63:    nop     dword ptr [rax + rax]
0x11aef1:    nop     dword ptr [rax]
0x11af1c:    nop     dword ptr [rax]
0x11af75:    nop     dword ptr [rax]
0x11afd3:    nop     dword ptr [rax + rax]
0x11b00b:    nop     dword ptr [rax + rax]
0x11b055:    nop     word ptr cs:[rax + rax]
0x11b060:    endbr64

```

```

0x11b064:    push    r15
0x11b066:    push    r14
0x11b068:    push    r13
0x11b06a:    push    r12
0x11b06c:    push    rbp
0x11b06d:    push    rbx
0x11b06e:    sub     rsp, 0x28
0x11b072:    movzx   r14d, byte ptr [rdi]
0x11b076:    test    r14b, r14b
0x11b079:    je      0x11b30d
0x11b30d:    mov     qword ptr [rsp], 0
0x11b315:    mov     rax, qword ptr [rsp]
0x11b319:    add     rsp, 0x28
0x11b31d:    pop     rbx
0x11b31e:    pop     rbp
0x11b31f:    pop     r12
0x11b321:    pop     r13
0x11b323:    pop     r14
0x11b325:    pop     r15
0x11b327:    ret
0x11b07f:    mov     qword ptr [rsp], 0
0x11b087:    mov     r12, rdi
0x11b08a:    mov     r13, rdi
0x11b08d:    xor     ebx, ebx
0x11b08f:    mov     dword ptr [rsp + 8], 0
0x11b097:    xor     ebp, ebp
0x11b099:    mov     dword ptr [rsp + 0xc], 0
0x11b0a1:    jmp     0x11b0c0
0x11b0c0:    cmp     r14b, 0x20
0x11b0c4:    je      0x11b0a8
0x11b0a8:    add     ebx, 1
0x11b0ab:    movsxd  rbp, ebx
0x11b0ae:    lea     r13, [r12 + rbp]
0x11b0b2:    movzx   r14d, byte ptr [r13]
0x11b0b7:    test    r14b, r14b
0x11b0ba:    je      0x11b315
0x11b0c6:    cmp     r14b, 9
0x11b0ca:    je      0x11b0a8
0x11b315:    mov     rax, qword ptr [rsp]
0x11b319:    add     rsp, 0x28
0x11b31d:    pop     rbx
0x11b31e:    pop     rbp
0x11b31f:    pop     r12
0x11b321:    pop     r13
0x11b323:    pop     r14
0x11b325:    pop     r15
0x11b327:    ret
0x11b0cc:    cmp     byte ptr [rip + 0x10a0d], r14b

```

```

0x11b0d3:    je      0x11b315
0x11b0d9:    movsx   r15d, r14b
0x11b0dd:    lea     rdi, [rip + 0x7379]
0x11b0e4:    mov     esi, r15d
0x11b0e7:    call    0x104a00
0x11b0ec:    test    rax, rax
0x11b0ef:    je      0x11b190
0x11b190:    mov     esi, r15d
0x11b193:    lea     rdi, [rip + 0x72c7]
0x11b19a:    call    0x104a00
0x11b0f5:    add     ebx, 1
0x11b0f8:    mov     r14d, 1
0x11b0fe:    mov     r9d, 2
0x11b104:    movsxd  rbp, ebx
0x11b107:    lea     rcx, [r12 + rbp]
0x11b10b:    mov     eax, dword ptr [rsp + 8]
0x11b10f:    add     eax, 2
0x11b112:    cmp     eax, dword ptr [rsp + 0xc]
0x11b116:    jge     0x11b2a0
0x11b19f:    test    rax, rax
0x11b1a2:    je      0x11b369
0x11b11c:    movsxd  r8, dword ptr [rsp + 8]
0x11b121:    mov     rdx, qword ptr [rsp]
0x11b125:    mov     rdi, r9
0x11b128:    mov     qword ptr [rsp + 0x18], rcx
0x11b12d:    shl     r8, 3
0x11b131:    lea     r15, [rdx + r8]
0x11b135:    mov     qword ptr [rsp + 0x10], r8
0x11b13a:    call    0x11d670
0x11b2a0:    add     dword ptr [rsp + 0xc], 0xa
0x11b2a5:    movsxd  rsi, dword ptr [rsp + 0xc]
0x11b2aa:    mov     rdi, qword ptr [rsp]
0x11b2ae:    mov     qword ptr [rsp + 0x18], rcx
0x11b2b3:    shl     rsi, 3
0x11b2b7:    mov     qword ptr [rsp + 0x10], r9
0x11b2bc:    call    0x11d690
0x11b369:    mov     esi, r15d
0x11b36c:    lea     rdi, [rip + 0x70f5]
0x11b373:    call    0x104a00
0x11b1a8:    movsx   edx, byte ptr [r12 + rbp + 1]
0x11b1ae:    cmp     dl, r14b
0x11b1b1:    je      0x11b2d8
0x11b13f:    mov     rdx, r14
0x11b142:    mov     rsi, r13
0x11b145:    mov     qword ptr [r15], rax
0x11b148:    mov     rdi, rax
0x11b14b:    call    0x104880
0x11b2c1:    mov     rcx, qword ptr [rsp + 0x18]

```

```

0x11b2c6:    mov     r9, qword ptr [rsp + 0x10]
0x11b2cb:    mov     qword ptr [rsp], rax
0x11b2cf:    jmp     0x11b11c
0x11b378:    test    rax, rax
0x11b37b:    je      0x11b1dd
0x11b2d8:    cmp     edx, 0x24
0x11b2db:    je      0x11b1d3
0x11b1b7:    cmp     edx, 0x26
0x11b1ba:    je      0x11b340
0x11b150:    mov     rax, qword ptr [r15]
0x11b153:    mov     r8, qword ptr [rsp + 0x10]
0x11b158:    mov     rcx, qword ptr [rsp + 0x18]
0x11b15d:    add     dword ptr [rsp + 8], 1
0x11b162:    mov     byte ptr [rax + r14], 0
0x11b167:    mov     rax, qword ptr [rsp]
0x11b16b:    mov     qword ptr [rax + r8 + 8], 0
0x11b174:    movzx   r14d, byte ptr [rcx]
0x11b178:    test    r14b, r14b
0x11b17b:    je      0x11b315
0x11b1dd:    mov     edx, ebx
0x11b1df:    xor     r15d, r15d
0x11b1e2:    movsxd  r8, edx
0x11b1e5:    lea     rcx, [r12 + r8]
0x11b1e9:    mov     rbp, r8
0x11b1ec:    movzx   eax, byte ptr [rcx]
0x11b1ef:    test    al, al
0x11b1f1:    je      0x11b404
0x11b381:    lea     edx, [rbx + 1]
0x11b384:    jmp     0x11b1e2
0x11b1d3:    cmp     r14b, 0x24
0x11b1d7:    jne     0x11b0f5
0x11b2e1:    mov     eax, ebx
0x11b2e3:    cmp     edx, 0x3c
0x11b2e6:    je      0x11b3f0
0x11b340:    mov     eax, r14d
0x11b343:    and     eax, 0xffffffff
0x11b346:    cmp     al, 0x3c
0x11b348:    jne     0x11b1d3
0x11b1c0:    cmp     edx, 0x3e
0x11b1c3:    jne     0x11b3ac
0x11b181:    mov     r13, rcx
0x11b184:    jmp     0x11b0c0
0x11b404:    mov     r14d, edx
0x11b407:    sub     r14d, ebx
0x11b40a:    mov     ebx, edx
0x11b40c:    lea     r9d, [r14 + 1]
0x11b410:    movsxd  r14, r14d
0x11b413:    movsxd  r9, r9d

```



```

0x11b416:      jmp      0x11b10b
0x11b1f7:      mov     qword ptr [rsp + 0x10], r13
0x11b1fc:      lea     r14, [rip + 0x7265]
0x11b203:      mov     r13, r8
0x11b206:      mov     dword ptr [rsp + 0x18], ebx
0x11b20a:      mov     rbx, r12
0x11b20d:      mov     r12d, edx
0x11b210:      jmp     0x11b265
0x11b1e2:      movsxd  r8, edx
0x11b1e5:      lea     rcx, [r12 + r8]
0x11b1e9:      mov     rbp, r8
0x11b1ec:      movzx   eax, byte ptr [rcx]
0x11b1ef:      test    al, al
0x11b1f1:      je      0x11b404
0x11b3f0:      cmp     byte ptr [r12 + rbp + 2], 0x2d
0x11b3f6:      jne     0x11b2ec
0x11b2ec:      add     eax, 2
0x11b2ef:      mov     r14d, eax
0x11b2f2:      movsxd  rbp, eax
0x11b2f5:      sub     r14d, ebx
0x11b2f8:      lea     rcx, [r12 + rbp]
0x11b2fc:      mov     ebx, eax
0x11b2fe:      lea     r9d, [r14 + 1]
0x11b302:      movsxd  r14, r14d
0x11b305:      movsxd  r9, r9d
0x11b308:      jmp     0x11b10b
0x11b34e:      add     ebx, 2
0x11b351:      mov     r14d, 2
0x11b357:      mov     r9d, 3
0x11b35d:      movsxd  rbp, ebx
0x11b360:      lea     rcx, [r12 + rbp]
0x11b364:      jmp     0x11b10b
0x11b1c9:      cmp     r14b, 0x26
0x11b1cd:      je      0x11b34e
0x11b3ac:      cmp     r14b, 0x24
0x11b3b0:      jne     0x11b0f5
0x11b10b:      mov     eax, dword ptr [rsp + 8]
0x11b10f:      add     eax, 2
0x11b112:      cmp     eax, dword ptr [rsp + 0xc]
0x11b116:      jge     0x11b2a0
0x11b265:      cmp     al, 0x5c
0x11b267:      jne     0x11b218
0x11b3fc:      lea     eax, [rbx + 1]
0x11b3ff:      jmp     0x11b2ec
0x11b3b6:      cmp     edx, 0x28
0x11b3b9:      je      0x11b34e
0x11b269:      cmp     byte ptr [rbx + r13 + 1], 0xa
0x11b26f:      lea     esi, [r12 + 1]

```

```

0x11b274:      je      0x11b330
0x11b218:      movsx   ebp, al
0x11b21b:      test    r15d, r15d
0x11b21e:      jne      0x11b290
0x11b3bb:      jmp      0x11b1d3
0x11b330:      lea      esi, [r12 + 2]
0x11b335:      jmp      0x11b24d
0x11b27a:      xor      eax, eax
0x11b27c:      cmp      r15d, 0x27
0x11b280:      setne    al
0x11b283:      add      esi, eax
0x11b285:      jmp      0x11b24d
0x11b220:      mov      esi, ebp
0x11b222:      lea      rdi, [rip + 0x7243]
0x11b229:      call     0x104a00
0x11b290:      xor      eax, eax
0x11b292:      cmp      r15d, ebp
0x11b295:      lea      esi, [r12 + 1]
0x11b29a:      cmov     r15d, eax
0x11b29e:      jmp      0x11b24d
0x11b24d:      movsxd   rbp, esi
0x11b250:      mov      r12d, esi
0x11b253:      lea      rcx, [rbx + rbp]
0x11b257:      mov      r13, rbp
0x11b25a:      movzx    eax, byte ptr [rcx]
0x11b25d:      test     al, al
0x11b25f:      je        0x11b389
0x11b22e:      test     rax, rax
0x11b231:      jne      0x11b3c0
0x11b389:      mov      r12, rbx
0x11b38c:      mov      ebx, dword ptr [rsp + 0x18]
0x11b390:      mov      r14d, esi
0x11b393:      mov      r13, qword ptr [rsp + 0x10]
0x11b398:      sub      r14d, ebx
0x11b39b:      mov      ebx, esi
0x11b39d:      lea      r9d, [r14 + 1]
0x11b3a1:      movsxd   r14, r14d
0x11b3a4:      movsxd   r9, r9d
0x11b3a7:      jmp      0x11b10b
0x11b237:      mov      esi, ebp
0x11b239:      mov      rdi, r14
0x11b23c:      call     0x104a00
0x11b3c0:      mov      edx, r12d
0x11b3c3:      mov      r12, rbx
0x11b3c6:      mov      ebx, dword ptr [rsp + 0x18]
0x11b3ca:      mov      r8, r13
0x11b3cd:      mov      r14d, edx
0x11b3d0:      mov      r13, qword ptr [rsp + 0x10]

```

```

0x11b3d5:    mov     rbp, r8
0x11b3d8:    lea     rcx, [r12 + r8]
0x11b3dc:    sub     r14d, ebx
0x11b3df:    mov     ebx, edx
0x11b3e1:    lea     r9d, [r14 + 1]
0x11b3e5:    movsxd  r14, r14d
0x11b3e8:    movsxd  r9, r9d
0x11b3eb:    jmp     0x11b10b
0x11b241:    lea     esi, [r12 + 1]
0x11b246:    test    rax, rax
0x11b249:    cmovne  r15d, ebp
0x11b24d:    movsxd  rbp, esi
0x11b250:    mov     r12d, esi
0x11b253:    lea     rcx, [rbx + rbp]
0x11b257:    mov     r13, rbp
0x11b25a:    movzx   eax, byte ptr [rcx]
0x11b25d:    test    al, al
0x11b25f:    je      0x11b389
0x11b0a3:    nop     dword ptr [rax + rax]
0x11b189:    nop     dword ptr [rax]
0x11b212:    nop     word ptr [rax + rax]
0x11b287:    nop     word ptr [rax + rax]
0x11b2d4:    nop     dword ptr [rax]
0x11b328:    nop     dword ptr [rax + rax]
0x11b33a:    nop     word ptr [rax + rax]
0x11b41b:    nop     dword ptr [rax + rax]
0x11b420:    endbr64
0x11b424:    push    r15
0x11b426:    xor     eax, eax
0x11b428:    push    r14
0x11b42a:    push    r13
0x11b42c:    push    r12
0x11b42e:    mov     r12d, esi
0x11b431:    push    rbp
0x11b432:    push    rbx
0x11b433:    mov     ebx, edi
0x11b435:    mov     rdi, rdx
0x11b438:    sub     rsp, 8
0x11b43c:    call    0x11b060
0x11b441:    test    rax, rax
0x11b444:    je      0x11b598
0x11b598:    xor     r13d, r13d
0x11b59b:    jmp     0x11b579
0x11b44a:    cmp     qword ptr [rax], 0
0x11b44e:    mov     rbp, rax
0x11b451:    je      0x11b590
0x11b579:    add     rsp, 8
0x11b57d:    mov     rax, r13

```

```

0x11b580:    pop     rbx
0x11b581:    pop     rbp
0x11b582:    pop     r12
0x11b584:    pop     r13
0x11b586:    pop     r14
0x11b588:    pop     r15
0x11b58a:    ret
0x11b590:    xor     r14d, r14d
0x11b593:    jmp     0x11b46f
0x11b457:    mov     eax, 1
0x11b45c:    nop     dword ptr [rax]
0x11b460:    mov     r14d, eax
0x11b463:    add     rax, 1
0x11b467:    cmp     qword ptr [rbp + rax*8 - 8], 0
0x11b46d:    jne     0x11b460
0x11b46f:    lea     eax, [r14 + r12 - 1]
0x11b474:    test    r12d, r12d
0x11b477:    cmovs   r12d, eax
0x11b47b:    lea     eax, [r14 + rbx - 1]
0x11b480:    test    ebx, ebx
0x11b482:    cmovs   ebx, eax
0x11b485:    lea     eax, [r12 + 1]
0x11b48a:    cmp     r12d, 0x24
0x11b48e:    jne     0x11b497
0x11b460:    mov     r14d, eax
0x11b463:    add     rax, 1
0x11b467:    cmp     qword ptr [rbp + rax*8 - 8], 0
0x11b46d:    jne     0x11b460
0x11b490:    lea     r12d, [r14 - 1]
0x11b494:    mov     eax, r14d
0x11b497:    lea     edx, [r14 - 1]
0x11b49b:    cmp     ebx, 0x24
0x11b49e:    cmov     ebx, edx
0x11b4a1:    mov     edx, eax
0x11b4a3:    cmp     ebx, eax
0x11b4a5:    cmovge   edx, ebx
0x11b4a8:    xor     r13d, r13d
0x11b4ab:    cmp     edx, r14d
0x11b4ae:    jg      0x11b547
0x11b497:    lea     edx, [r14 - 1]
0x11b49b:    cmp     ebx, 0x24
0x11b49e:    cmov     ebx, edx
0x11b4a1:    mov     edx, eax
0x11b4a3:    cmp     ebx, eax
0x11b4a5:    cmovge   edx, ebx
0x11b4a8:    xor     r13d, r13d
0x11b4ab:    cmp     edx, r14d
0x11b4ae:    jg      0x11b547

```

```

0x11b4b4:    mov     ecx, ebx
0x11b4b6:    or      ecx, eax
0x11b4b8:    js      0x11b547
0x11b547:    test    r14d, r14d
0x11b54a:    je      0x11b571
0x11b4be:    cmp     ebx, eax
0x11b4c0:    jge     0x11b59d
0x11b571:    mov     rdi, rbp
0x11b574:    call    0x118d10
0x11b54c:    lea     eax, [r14 - 1]
0x11b550:    mov     rbx, rbp
0x11b553:    lea     r12, [rbp + rax*8 + 8]
0x11b558:    nop     dword ptr [rax + rax]
0x11b560:    mov     rdi, qword ptr [rbx]
0x11b563:    add     rbx, 8
0x11b567:    call    0x118d10
0x11b4c6:    movsxd  rbx, ebx
0x11b4c9:    xor     r15d, r15d
0x11b4cc:    mov     r13, rbx
0x11b4cf:    nop
0x11b4d0:    mov     rdi, qword ptr [rbp + r13*8]
0x11b4d5:    add     r13, 1
0x11b4d9:    call    0x1049b0
0x11b59d:    mov     edi, 1
0x11b5a2:    call    0x11d670
0x11b56c:    cmp     r12, rbx
0x11b56f:    jne     0x11b560
0x11b4de:    lea     r15d, [r15 + rax + 1]
0x11b4e3:    lea     eax, [r13 - 1]
0x11b4e7:    cmp     r12d, eax
0x11b4ea:    jg      0x11b4d0
0x11b5a7:    mov     r13, rax
0x11b5aa:    jmp     0x11b547
0x11b560:    mov     rdi, qword ptr [rbx]
0x11b563:    add     rbx, 8
0x11b567:    call    0x118d10
0x11b4ec:    lea     edi, [r15 + 1]
0x11b4f0:    xor     r15d, r15d
0x11b4f3:    movsxd  rdi, edi
0x11b4f6:    call    0x11d670
0x11b4d0:    mov     rdi, qword ptr [rbp + r13*8]
0x11b4d5:    add     r13, 1
0x11b4d9:    call    0x1049b0
0x11b4fb:    mov     r13, rax
0x11b4fe:    nop
0x11b500:    mov     rsi, qword ptr [rbp + rbx*8]
0x11b505:    movsxd  rdi, r15d
0x11b508:    add     rdi, r13

```

```

0x11b50b:    call    0x1048b0
0x11b510:    mov     rdi, qword ptr [rbp + rbx*8]
0x11b515:    call    0x1049b0
0x11b51a:    add     r15d, eax
0x11b51d:    cmp     r12d, ebx
0x11b520:    jle     0x11b53b
0x11b53b:    add     rbx, 1
0x11b53f:    lea     eax, [rbx - 1]
0x11b542:    cmp     r12d, eax
0x11b545:    jg      0x11b500
0x11b522:    lea     eax, [r15 + 1]
0x11b526:    movsxd  r15, r15d
0x11b529:    movsxd  rdx, eax
0x11b52c:    mov     byte ptr [r13 + r15], 0x20
0x11b532:    mov     r15d, eax
0x11b535:    mov     byte ptr [r13 + rdx], 0
0x11b53b:    add     rbx, 1
0x11b53f:    lea     eax, [rbx - 1]
0x11b542:    cmp     r12d, eax
0x11b545:    jg      0x11b500
0x11b500:    mov     rsi, qword ptr [rbp + rbx*8]
0x11b505:    movsxd  rdi, r15d
0x11b508:    add     rdi, r13
0x11b50b:    call    0x1048b0
0x11b58b:    nop     dword ptr [rax + rax]
0x11b5ac:    nop     dword ptr [rax]
0x11b5b0:    endbr64
0x11b5b4:    push    r15
0x11b5b6:    push    r14
0x11b5b8:    push    r13
0x11b5ba:    push    r12
0x11b5bc:    push    rbp
0x11b5bd:    push    rbx
0x11b5be:    mov     rbx, rdi
0x11b5c1:    sub     rsp, 0x88
0x11b5c8:    mov     qword ptr [rsp + 0x20], rdi
0x11b5cd:    mov     qword ptr [rsp + 0x30], rsi
0x11b5d2:    mov     rax, qword ptr fs:[0x28]
0x11b5db:    mov     qword ptr [rsp + 0x78], rax
0x11b5e0:    xor     eax, eax
0x11b5e2:    cmp     byte ptr [rip + 0xf204], 0
0x11b5e9:    je      0x11b656
0x11b656:    mov     rdi, rbx
0x11b659:    call    0x1049b0
0x11b5eb:    mov     edi, 0x100
0x11b5f0:    call    0x11d670
0x11b65e:    lea     rdi, [rax + 1]
0x11b662:    call    0x11d670

```

```

0x11b5f5:    mov     rbx, qword ptr [rsp + 0x20]
0x11b5fa:    mov     byte ptr [rax], 0
0x11b5fd:    mov     r15, rax
0x11b600:    mov     rdi, rbx
0x11b603:    call    0x1049b0
0x11b667:    mov     rsi, rbx
0x11b66a:    mov     rdi, rax
0x11b66d:    call    0x1048b0
0x11b608:    mov     dword ptr [rsp + 0xc], eax
0x11b60c:    mov     r12, rax
0x11b60f:    movzx   eax, byte ptr [rbx]
0x11b612:    cmp     al, byte ptr [rip + 0xf1d4]
0x11b618:    je      0x11c1e0
0x11b672:    mov     rcx, qword ptr [rsp + 0x30]
0x11b677:    mov     qword ptr [rcx], rax
0x11b67a:    xor     eax, eax
0x11b67c:    mov     rdx, qword ptr [rsp + 0x78]
0x11b681:    sub     rdx, qword ptr fs:[0x28]
0x11b68a:    jne     0x11c487
0x11c1e0:    lea     edi, [r12 + 5]
0x11c1e5:    movsxd  rdi, edi
0x11c1e8:    call    0x11d670
0x11b61e:    movzx   ebp, byte ptr [rip + 0xf1c8]
0x11b625:    mov     r13, qword ptr [rip + 0xf1c4]
0x11b62c:    lea     rbx, [rbx + 1]
0x11b630:    test    al, al
0x11b632:    jne     0x11b6c7
0x11b690:    add     rsp, 0x88
0x11b697:    pop     rbx
0x11b698:    pop     rbp
0x11b699:    pop     r12
0x11b69b:    pop     r13
0x11b69d:    pop     r14
0x11b69f:    pop     r15
0x11b6a1:    ret
0x11c487:    call    0x1049d0
0x11c1ed:    mov     r11d, 0x733a
0x11c1f3:    mov     rsi, qword ptr [rsp + 0x20]
0x11c1f8:    mov     rbp, rax
0x11c1fb:    movzx   eax, byte ptr [rip + 0xe5eb]
0x11c202:    mov     word ptr [rbp + 2], r11w
0x11c207:    lea     rdi, [rbp + 4]
0x11c20b:    mov     byte ptr [rbp + 1], al
0x11c20e:    mov     byte ptr [rbp], al
0x11c211:    call    0x1048b0
0x11b6c7:    movsx   esi, byte ptr [rbx]
0x11b6ca:    cmp     al, bpl
0x11b6cd:    jne     0x11b6b8

```

```

0x11b638:    nop    dword ptr [rax + rax]
0x11c216:    lea    eax, [r12 + 4]
0x11c21b:    mov    dword ptr [rsp + 0xc], eax
0x11c21f:    test   eax, eax
0x11c221:    jg     0x11b6e6
0x11b6cf:    test   esi, esi
0x11b6d1:    je     0x11b6b8
0x11b6b8:    movzx  eax, byte ptr [rbx]
0x11b6bb:    add    rbx, 1
0x11b6bf:    test   al, al
0x11b6c1:    je     0x11b640
0x11b640:    test   bpl, bpl
0x11b643:    je     0x11b6d8
0x11c227:    mov    rax, qword ptr [rsp + 0x30]
0x11c22c:    mov    qword ptr [rax], r15
0x11c22f:    cmp    qword ptr [rsp + 0x20], rbp
0x11c234:    je     0x11b67a
0x11b6e6:    mov    dword ptr [rsp + 4], 0x100
0x11b6ee:    xor    r13d, r13d
0x11b6f1:    xor    ebx, ebx
0x11b6f3:    mov    dword ptr [rsp + 0x28], 0
0x11b6fb:    mov    dword ptr [rsp + 0x3c], 0
0x11b703:    mov    dword ptr [rsp + 8], 0
0x11b70b:    jmp    0x11b758
0x11b6d3:    jmp    0x11b6a8
0x11b6d8:    mov    rbp, qword ptr [rsp + 0x20]
0x11b6dd:    test   r12d, r12d
0x11b6e0:    jle    0x11c40a
0x11b649:    mov    rdi, r15
0x11b64c:    call   0x118d10
0x11b67a:    xor    eax, eax
0x11b67c:    mov    rdx, qword ptr [rsp + 0x78]
0x11b681:    sub    rdx, qword ptr fs:[0x28]
0x11b68a:    jne    0x11c487
0x11c23a:    mov    rdi, rbp
0x11c23d:    call   0x118d10
0x11b758:    movsxd r14, r13d
0x11b75b:    mov    r10d, dword ptr [rsp + 8]
0x11b760:    lea    rdx, [rbp + r14]
0x11b765:    movzx  r12d, byte ptr [rdx]
0x11b769:    test   r10d, r10d
0x11b76c:    je     0x11b710
0x11b6a8:    mov    rdi, r13
0x11b6ab:    call   0x104a00
0x11c40a:    mov    rax, qword ptr [rsp + 0x30]
0x11c40f:    mov    qword ptr [rax], r15
0x11c412:    jmp    0x11b67a
0x11b651:    mov    rbx, qword ptr [rsp + 0x20]

```



```

0x11b656:    mov     rdi, rbx
0x11b659:    call    0x1049b0
0x11c242:    mov     dword ptr [rsp + 0x3c], 0
0x11c24a:    jmp     0x11b7c0
0x11b710:    cmp     r12b, byte ptr [rip + 0xf0d6]
0x11b717:    je      0x11b7d0
0x11b76e:    cmp     ebx, dword ptr [rsp + 4]
0x11b772:    jge     0x11be10
0x11b6b0:    test    rax, rax
0x11b6b3:    je      0x11b6d8
0x11b7c0:    mov     edx, dword ptr [rsp + 0x3c]
0x11b7c4:    xor     eax, eax
0x11b7c6:    test    edx, edx
0x11b7c8:    setne   al
0x11b7cb:    jmp     0x11b67c
0x11b7d0:    lea     rcx, [rbp + r14 + 1]
0x11b7d5:    mov     qword ptr [rsp + 0x10], rdx
0x11b7da:    movsx   esi, byte ptr [rcx]
0x11b7dd:    test    esi, esi
0x11b7df:    je      0x11b830
0x11b71d:    cmp     r12b, 0xfd
0x11b721:    je      0x11b7d0
0x11b778:    lea     eax, [rbx + 1]
0x11b77b:    mov     byte ptr [r15 + rbx], r12b
0x11b77f:    add     r13d, 1
0x11b783:    movsxd  rdx, eax
0x11b786:    mov     dword ptr [rsp + 8], 0
0x11b78e:    movsxd  rbx, eax
0x11b791:    mov     byte ptr [r15 + rdx], 0
0x11b796:    cmp     r13d, dword ptr [rsp + 0xc]
0x11b79b:    jl      0x11b758
0x11be10:    add     dword ptr [rsp + 4], 0x40
0x11be15:    movsxd  rsi, dword ptr [rsp + 4]
0x11be1a:    mov     rdi, r15
0x11be1d:    call    0x11d690
0x11b6b5:    nop     dword ptr [rax]
0x11b67c:    mov     rdx, qword ptr [rsp + 0x78]
0x11b681:    sub     rdx, qword ptr fs:[0x28]
0x11b68a:    jne     0x11c487
0x11b830:    mov     edi, 0x80
0x11b835:    mov     qword ptr [rsp + 0x18], rcx
0x11b83a:    call    0x11d670
0x11b7e1:    mov     rdi, qword ptr [rip + 0xf008]
0x11b7e8:    mov     qword ptr [rsp + 0x18], rcx
0x11b7ed:    call    0x104a00
0x11b727:    cmp     r12b, 0x5c
0x11b72b:    je      0x11bde0
0x11b79d:    mov     rax, qword ptr [rsp + 0x30]

```

```

0x11b7a2:    mov     qword ptr [rax], r15
0x11b7a5:    cmp     qword ptr [rsp + 0x20], rbp
0x11b7aa:    je      0x11b7b4
0x11be22:    mov     r15, rax
0x11be25:    jmp     0x11b778
0x11b83f:    mov     rcx, qword ptr [rsp + 0x18]
0x11b844:    mov     dword ptr [rsp + 0x6c], r13d
0x11b849:    mov     qword ptr [rsp + 0x10], rax
0x11b84e:    movzx   r12d, byte ptr [rcx]
0x11b852:    test    r12b, r12b
0x11b855:    je      0x11be30
0x11b7f2:    mov     rcx, qword ptr [rsp + 0x18]
0x11b7f7:    test    rax, rax
0x11b7fa:    je      0x11b830
0x11bde0:    cmp     ebx, dword ptr [rsp + 4]
0x11bde4:    jge     0x11bff0
0x11b731:    cmp     ebx, dword ptr [rsp + 4]
0x11b735:    jge     0x11bfe0
0x11b7b4:    mov     ecx, dword ptr [rsp + 0x28]
0x11b7b8:    test    ecx, ecx
0x11b7ba:    jne     0x11c35b
0x11b7ac:    mov     rdi, rbp
0x11b7af:    call    0x118d10
0x11be30:    xor     edx, edx
0x11be32:    test    r13d, r13d
0x11be35:    je      0x11be4d
0x11b85b:    movsx   esi, r12b
0x11b85f:    lea     rdi, [rip + 0x6c11]
0x11b866:    call    0x104a00
0x11b7fc:    cmp     ebx, dword ptr [rsp + 4]
0x11b800:    jl      0x11b73b
0x11bdea:    lea     eax, [rbx + 1]
0x11bded:    mov     byte ptr [r15 + rbx], 0x5c
0x11bdf2:    movsxd  rdx, eax
0x11bdf5:    mov     dword ptr [rsp + 8], 1
0x11bdfd:    movsxd  rbx, eax
0x11be00:    mov     byte ptr [r15 + rdx], 0
0x11be05:    jmp     0x11b74d
0x11bff0:    add     dword ptr [rsp + 4], 0x40
0x11bff5:    movsxd  rsi, dword ptr [rsp + 4]
0x11bffa:    mov     rdi, r15
0x11bffd:    call    0x11d690
0x11b73b:    lea     eax, [rbx + 1]
0x11b73e:    mov     byte ptr [r15 + rbx], r12b
0x11b742:    movsxd  rdx, eax
0x11b745:    movsxd  rbx, eax
0x11b748:    mov     byte ptr [r15 + rdx], 0
0x11b74d:    add     r13d, 1

```

```

0x11b751:    cmp     r13d, dword ptr [rsp + 0xc]
0x11b756:    jge     0x11b79d
0x11bfe0:    mov     qword ptr [rsp + 0x10], rdx
0x11bfe5:    jmp     0x11b806
0x11c35b:    xor     eax, eax
0x11c35d:    mov     rdi, r15
0x11c360:    call    0x11a380
0x11be4d:    lea     rsi, [rsp + 0x6c]
0x11be52:    mov     rdi, rbp
0x11be55:    xor     eax, eax
0x11be57:    call    0x11ac70
0x11be37:    movzx   eax, byte ptr [rbp + r14 - 1]
0x11be3d:    cmp     al, 0x27
0x11be3f:    je      0x11c078
0x11b86b:    test    rax, rax
0x11b86e:    je      0x11bfc0
0x11b806:    add     dword ptr [rsp + 4], 0x40
0x11b80b:    movsxd  rsi, dword ptr [rsp + 4]
0x11b810:    mov     rdi, r15
0x11b813:    call    0x11d690
0x11b74d:    add     r13d, 1
0x11b751:    cmp     r13d, dword ptr [rsp + 0xc]
0x11b756:    jge     0x11b79d
0x11c002:    mov     r15, rax
0x11c005:    jmp     0x11bdea
0x11c365:    mov     eax, 2
0x11c36a:    jmp     0x11b67c
0x11be5c:    mov     r12d, dword ptr [rsp + 0x6c]
0x11be61:    mov     r14, rax
0x11be64:    jmp     0x11b8b3
0x11c078:    movsx   edx, al
0x11c07b:    jmp     0x11be4d
0x11be45:    cmp     al, 0x22
0x11be47:    je      0x11c078
0x11bfc0:    cmp     r12b, 0x23
0x11bfc4:    jne     0x11be30
0x11b874:    lea     eax, [r13 + 1]
0x11b878:    lea     rsi, [rsp + 0x70]
0x11b87d:    xor     edx, edx
0x11b87f:    mov     byte ptr [rsp + 0x77], 0
0x11b884:    mov     dword ptr [rsp + 0x6c], eax
0x11b888:    movzx   eax, byte ptr [rip + 0xef5e]
0x11b88f:    lea     rdi, [rsp + 0x75]
0x11b894:    mov     dword ptr [rsp + 0x70], 0
0x11b89c:    mov     byte ptr [rsp + 0x76], al
0x11b8a0:    mov     byte ptr [rsp + 0x75], al
0x11b8a4:    xor     eax, eax
0x11b8a6:    call    0x11ac70

```

```

0x11b818:    mov     rdx, qword ptr [rsp + 0x10]
0x11b81d:    mov     r15, rax
0x11b820:    movzx   r12d, byte ptr [rdx]
0x11b824:    jmp     0x11b73b
0x11b8b3:    test    r14, r14
0x11b8b6:    je      0x11c435
0x11bfca:    lea     r12d, [r13 + 2]
0x11bfce:    mov     r14, r15
0x11bfd1:    mov     dword ptr [rsp + 0x6c], r12d
0x11bfd6:    jmp     0x11b8b3
0x11b8ab:    mov     r12d, dword ptr [rsp + 0x6c]
0x11b8b0:    mov     r14, rax
0x11b8b3:    test    r14, r14
0x11b8b6:    je      0x11c435
0x11c435:    xor     ecx, ecx
0x11c437:    mov     edx, r12d
0x11c43a:    mov     esi, r13d
0x11c43d:    jmp     0x11bdbe
0x11b8bc:    movsxd  rax, r12d
0x11b8bf:    xor     esi, esi
0x11b8c1:    mov     edx, r12d
0x11b8c4:    movzx   eax, byte ptr [rbp + rax]
0x11b8c9:    cmp     al, 0x3a
0x11b8cb:    jne     0x11b8df
0x11bdbe:    mov     rdi, rbp
0x11bdc1:    xor     eax, eax
0x11bdc3:    call    0x119fe0
0x11b8cd:    lea     edx, [r12 + 1]
0x11b8d2:    mov     esi, 1
0x11b8d7:    movsxd  rax, edx
0x11b8da:    movzx   eax, byte ptr [rbp + rax]
0x11b8df:    cmp     al, 0x25
0x11b8e1:    je      0x11c010
0x11b8df:    cmp     al, 0x25
0x11b8e1:    je      0x11c010
0x11bdc8:    mov     rdi, qword ptr [rsp + 0x10]
0x11bdcd:    mov     rbx, rax
0x11bdd0:    call    0x118d10
0x11c010:    mov     r13, qword ptr [rip + 0xfad9]
0x11c017:    add     edx, 1
0x11c01a:    mov     dword ptr [rsp + 0x6c], edx
0x11c01e:    test    r13, r13
0x11c021:    je      0x11c080
0x11b8e7:    cmp     al, 0x2d
0x11b8e9:    je      0x11c100
0x11bdd5:    jmp     0x11ba16
0x11c080:    mov     edi, 1
0x11c085:    call    0x11d670

```

```

0x11c023:    mov     rdi, r13
0x11c026:    call    0x1049b0
0x11c100:    xor     edi, edi
0x11c102:    lea     r8d, [rdx + 1]
0x11c106:    movsxd  rax, r8d
0x11c109:    movzx   eax, byte ptr [rbp + rax]
0x11c10e:    lea     ecx, [rax - 0x30]
0x11c111:    cmp     cl, 9
0x11c114:    jbe     0x11c283
0x11b8ef:    jg      0x11bd48
0x11ba16:    mov     rax, qword ptr [rsp + 0x30]
0x11ba1b:    mov     rdi, r15
0x11ba1e:    mov     qword ptr [rax], rbx
0x11ba21:    call    0x118d10
0x11c08a:    mov     byte ptr [rax], 0
0x11c08d:    mov     r13, rax
0x11c090:    jmp     0x11b92c
0x11c02b:    lea     rdi, [rax + 1]
0x11c02f:    call    0x11d670
0x11c283:    add     edx, 2
0x11c286:    xor     esi, esi
0x11c288:    movsxd  rdx, edx
0x11c28b:    nop     dword ptr [rax + rax]
0x11c290:    sub     eax, 0x30
0x11c293:    lea     ecx, [rsi + rsi*4]
0x11c296:    mov     r8d, edx
0x11c299:    movsx   eax, al
0x11c29c:    lea     esi, [rax + rcx*2]
0x11c29f:    movzx   eax, byte ptr [rbp + rdx]
0x11c2a4:    add     rdx, 1
0x11c2a8:    lea     ecx, [rax - 0x30]
0x11c2ab:    cmp     cl, 9
0x11c2ae:    jbe     0x11c290
0x11c11a:    cmp     al, 0x24
0x11c11c:    je      0x11c3e5
0x11b8f5:    cmp     al, 0x24
0x11b8f7:    je      0x11bf80
0x11bd48:    cmp     al, 0x39
0x11bd4a:    jg      0x11bd80
0x11ba26:    mov     eax, 0xffffffff
0x11ba2b:    cmp     qword ptr [rsp + 0x20], rbp
0x11ba30:    je      0x11b67c
0x11b92c:    lea     rax, [rip + 0x101ed]
0x11b933:    cmp     r13, rax
0x11b936:    je      0x11bdae
0x11c034:    mov     rsi, r13
0x11c037:    mov     rdi, rax
0x11c03a:    call    0x1048b0

```

```

0x11c290:    sub     eax, 0x30
0x11c293:    lea     ecx, [rsi + rsi*4]
0x11c296:    mov     r8d, edx
0x11c299:    movsx   eax, al
0x11c29c:    lea     esi, [rax + rcx*2]
0x11c29f:    movzx   eax, byte ptr [rbp + rdx]
0x11c2a4:    add     rdx, 1
0x11c2a8:    lea     ecx, [rax - 0x30]
0x11c2ab:    cmp     cl, 9
0x11c2ae:    jbe     0x11c290
0x11c2b0:    cmp     esi, 0x24
0x11c2b3:    sete    al
0x11c2b6:    cmp     esi, edi
0x11c2b8:    setge   dl
0x11c2bb:    or      eax, edx
0x11c2bd:    mov     dword ptr [rsp + 0x6c], r8d
0x11c2c2:    test    al, al
0x11c2c4:    je      0x11bdb3
0x11c3e5:    add     edx, 2
0x11c3e8:    mov     esi, 0x24
0x11c3ed:    mov     dword ptr [rsp + 0x6c], edx
0x11c3f1:    jmp     0x11bd98
0x11c122:    mov     dword ptr [rsp + 0x6c], r8d
0x11c127:    test    al, al
0x11c129:    je      0x11c3f6
0x11bf80:    add     edx, 1
0x11bf83:    mov     esi, 0x24
0x11bf88:    mov     edi, 0x24
0x11bf8d:    xor     eax, eax
0x11bf8f:    mov     dword ptr [rsp + 0x6c], edx
0x11bf93:    mov     rdx, r14
0x11bf96:    call    0x11b420
0x11b8fd:    cmp     al, 0x2a
0x11b8ff:    jne     0x11bd54
0x11bd4c:    cmp     al, 0x2f
0x11bd4e:    jg      0x11c150
0x11bd80:    cmp     al, 0x5e
0x11bd82:    jne     0x11bd54
0x11ba36:    mov     rdi, rbp
0x11ba39:    mov     dword ptr [rsp + 4], eax
0x11ba3d:    call    0x118d10
0x11bdae:    mov     r8d, dword ptr [rsp + 0x6c]
0x11bdb3:    mov     ecx, 1
0x11bdb8:    mov     edx, r8d
0x11bdbb:    mov     esi, r12d
0x11bdbe:    mov     rdi, rbp
0x11bdc1:    xor     eax, eax
0x11bdc3:    call    0x119fe0

```

```

0x11b93c:    mov     rdi, r13
0x11b93f:    call   0x1049b0
0x11c03f:    mov     r13, rax
0x11c042:    jmp     0x11b92c
0x11bdb3:    mov     ecx, 1
0x11bdb8:    mov     edx, r8d
0x11bdbb:    mov     esi, r12d
0x11bdbe:    mov     rdi, rbp
0x11bdc1:    xor     eax, eax
0x11bdc3:    call   0x119fe0
0x11c2ca:    jmp     0x11bd98
0x11bd98:    mov     rdx, r14
0x11bd9b:    xor     eax, eax
0x11bd9d:    call   0x11b420
0x11c3f6:    mov     esi, 0xffffffff
0x11c3fb:    jmp     0x11bd98
0x11c12f:    cmp     al, 0x3a
0x11c131:    je      0x11c3f6
0x11bf9b:    mov     r13, rax
0x11bf9e:    lea     rax, [rip + 0xfb7b]
0x11bfa5:    cmp     r13, rax
0x11bfa8:    je      0x11bdae
0x11b905:    add     edx, 1
0x11b908:    mov     esi, 0x24
0x11b90d:    mov     edi, 1
0x11b912:    xor     eax, eax
0x11b914:    mov     dword ptr [rsp + 0x6c], edx
0x11b918:    mov     rdx, r14
0x11b91b:    call   0x11b420
0x11bd54:    mov     rdi, r14
0x11bd57:    call   0x1049b0
0x11c150:    test    esi, esi
0x11c152:    je      0x11bd54
0x11bd84:    add     edx, 1
0x11bd87:    mov     edi, 1
0x11bd8c:    mov     esi, 1
0x11bd91:    mov     dword ptr [rsp + 0x6c], edx
0x11bd95:    nop     dword ptr [rax]
0x11bd98:    mov     rdx, r14
0x11bd9b:    xor     eax, eax
0x11bd9d:    call   0x11b420
0x11ba42:    mov     eax, dword ptr [rsp + 4]
0x11ba46:    jmp     0x11b67c
0x11b944:    lea     rdi, [rax + 1]
0x11b948:    call   0x11d670
0x11bda2:    mov     r13, rax
0x11bda5:    test    rax, rax
0x11bda8:    jne     0x11b92c

```

```

0x11c137:    test    edi, edi
0x11c139:    jne     0x11bdb3
0x11bfae:    test    r13, r13
0x11bfb1:    jne     0x11b93c
0x11b920:    mov     r13, rax
0x11b923:    test    rax, rax
0x11b926:    je      0x11c080
0x11bd5c:    lea     rdi, [rax + 1]
0x11bd60:    call    0x11d670
0x11c158:    add     edx, 1
0x11c15b:    xor     edi, edi
0x11c15d:    movsxd  rcx, edx
0x11c160:    sub     eax, 0x30
0x11c163:    lea     edx, [rdi + rdi*4]
0x11c166:    movsx   eax, al
0x11c169:    lea     edi, [rax + rdx*2]
0x11c16c:    movzx   eax, byte ptr [rbp + rcx]
0x11c171:    mov     edx, ecx
0x11c173:    add     rcx, 1
0x11c177:    lea     r8d, [rax - 0x30]
0x11c17b:    cmp     r8b, 9
0x11c17f:    jbe     0x11c160
0x11b94d:    mov     rsi, r13
0x11b950:    mov     rdi, rax
0x11b953:    call    0x1048b0
0x11c13f:    xor     esi, esi
0x11c141:    jmp     0x11bd98
0x11bfb7:    jmp     0x11bd54
0x11bd65:    mov     rsi, r14
0x11bd68:    mov     rdi, rax
0x11bd6b:    call    0x1048b0
0x11c160:    sub     eax, 0x30
0x11c163:    lea     edx, [rdi + rdi*4]
0x11c166:    movsx   eax, al
0x11c169:    lea     edi, [rax + rdx*2]
0x11c16c:    movzx   eax, byte ptr [rbp + rcx]
0x11c171:    mov     edx, ecx
0x11c173:    add     rcx, 1
0x11c177:    lea     r8d, [rax - 0x30]
0x11c17b:    cmp     r8b, 9
0x11c17f:    jbe     0x11c160
0x11c181:    cmp     al, 0x5e
0x11c183:    je      0x11c478
0x11b958:    mov     rdi, r13
0x11b95b:    mov     r14, rax
0x11b95e:    call    0x118d10
0x11bd70:    mov     r14, rax
0x11bd73:    jmp     0x11b963

```



```

0x11c478:    cmp     edi, 1
0x11c47b:    lea     r8d, [rdx + 1]
0x11c47f:    setle   al
0x11c482:    jmp     0x11c2bd
0x11c189:    cmp     al, 0x2a
0x11c18b:    je      0x11c250
0x11b963:    movsxd  rcx, dword ptr [rsp + 0x6c]
0x11b968:    cmp     byte ptr [rbp + rcx], 0x3a
0x11b96d:    mov     dword ptr [rsp + 0x44], ecx
0x11b971:    jne     0x11c3d4
0x11c2bd:    mov     dword ptr [rsp + 0x6c], r8d
0x11c2c2:    test    al, al
0x11c2c4:    je      0x11bdb3
0x11c250:    cmp     al, 0x5e
0x11c252:    je      0x11c478
0x11c191:    cmp     al, 0x2d
0x11c193:    je      0x11c102
0x11b977:    mov     dword ptr [rsp + 0x18], 0
0x11b97f:    lea     r12, [rip + 0x6afe]
0x11b986:    mov     dword ptr [rsp + 0x38], 0
0x11b98e:    mov     dword ptr [rsp + 0x40], ebx
0x11b992:    mov     qword ptr [rsp + 0x28], r15
0x11b997:    mov     r15, r14
0x11b99a:    mov     r14d, ecx
0x11b99d:    nop     dword ptr [rax]
0x11b9a0:    movzx   eax, byte ptr [rbp + rcx + 1]
0x11b9a5:    cmp     al, 0x67
0x11b9a7:    jne     0x11b9bf
0x11c3d4:    mov     dword ptr [rsp + 0x28], 0
0x11c3dc:    mov     edx, dword ptr [rsp + 0x44]
0x11c3e0:    jmp     0x11bad1
0x11c258:    add     edx, 1
0x11c25b:    mov     esi, 0x24
0x11c260:    mov     dword ptr [rsp + 0x6c], edx
0x11c264:    jmp     0x11bd98
0x11c102:    lea     r8d, [rdx + 1]
0x11c106:    movsxd  rax, r8d
0x11c109:    movzx   eax, byte ptr [rbp + rax]
0x11c10e:    lea     ecx, [rax - 0x30]
0x11c111:    cmp     cl, 9
0x11c114:    jbe     0x11c283
0x11c199:    mov     dword ptr [rsp + 0x6c], edx
0x11c19d:    mov     esi, edi
0x11c19f:    jmp     0x11bd98
0x11b9a9:    add     r14d, 1
0x11b9ad:    mov     dword ptr [rsp + 0x18], 1
0x11b9b5:    mov     dword ptr [rsp + 0x6c], r14d
0x11b9ba:    movzx   eax, byte ptr [rbp + rcx + 2]

```

```

0x11b9bf:    cmp     al, 0x26
0x11b9c1:    je      0x11be70
0x11b9bf:    cmp     al, 0x26
0x11b9c1:    je      0x11be70
0x11bad1:    sub     edx, 1
0x11bad4:    mov     rdi, r14
0x11bad7:    mov     dword ptr [rsp + 0x6c], edx
0x11badb:    call    0x1049b0
0x11be70:    add     dword ptr [rsp + 0x6c], 2
0x11be75:    mov     rdi, r15
0x11be78:    call    0x1049b0
0x11b9c7:    sub     eax, 0x65
0x11b9ca:    cmp     al, 0xf
0x11b9cc:    ja      0x11b9e0
0x11bae0:    cmp     eax, 0x80
0x11bae5:    jg      0x11c269
0x11be7d:    mov     r13d, dword ptr [rip + 0xfc88]
0x11be84:    cmp     eax, r13d
0x11be87:    jl      0x11c400
0x11b9ce:    movzx   eax, al
0x11b9d1:    movsxd  rax, dword ptr [r12 + rax*4]
0x11b9d5:    add     rax, r12
0x11b9d8:    jmp     rax
0x11b9e0:    mov     esi, dword ptr [rsp + 0x6c]
0x11b9e4:    mov     r14, r15
0x11b9e7:    mov     r15, qword ptr [rsp + 0x28]
0x11b9ec:    mov     ecx, 3
0x11b9f1:    lea     edx, [rsi + 2]
0x11b9f4:    add     esi, 1
0x11b9f7:    mov     rdi, rbp
0x11b9fa:    xor     eax, eax
0x11b9fc:    call    0x119fe0
0x11baeb:    mov     r12, qword ptr [rsp + 0x10]
0x11baf0:    mov     rsi, r14
0x11baf3:    mov     rdi, r12
0x11baf6:    call    0x1048b0
0x11c269:    add     eax, 1
0x11c26c:    mov     rdi, qword ptr [rsp + 0x10]
0x11c271:    movsxd  rsi, eax
0x11c274:    call    0x11d690
0x11c400:    mov     r14, r15
0x11c403:    mov     r15, qword ptr [rsp + 0x28]
0x11c408:    jmp     0x11c3c2
0x11be8d:    mov     qword ptr [rsp + 0x48], rbp
0x11be92:    xor     r14d, r14d
0x11be95:    mov     rbp, r15
0x11be98:    mov     r15d, eax
0x11be9b:    mov     dword ptr [rsp + 0x50], 1

```

```

0x11bea3:    jmp     0x11beba
0x11bb80:    mov     esi, 0x2e
0x11bb85:    mov     rdi, r15
0x11bb88:    call    0x104a20
0x11bba0:    mov     eax, dword ptr [rsp + 0x6c]
0x11bba4:    add     dword ptr [rsp + 0x38], 1
0x11bba9:    lea     r14d, [rax + 2]
0x11bbad:    jmp     0x11baa0
0x11bb30:    mov     esi, 0x2e
0x11bb35:    mov     rdi, r15
0x11bb38:    call    0x104a20
0x11ba50:    mov     esi, 0x2f
0x11ba55:    mov     rdi, r15
0x11ba58:    call    0x104a20
0x11bbb8:    mov     ebx, dword ptr [rsp + 0x6c]
0x11bbbc:    mov     rdi, rbp
0x11bbbf:    lea     r14d, [rbx + 2]
0x11bbc3:    call    0x1049b0
0x11bb58:    mov     esi, 0x2f
0x11bb5d:    mov     rdi, r15
0x11bb60:    call    0x104a20
0x11ba01:    mov     rdi, qword ptr [rsp + 0x10]
0x11ba06:    mov     rbx, rax
0x11ba09:    call    0x118d10
0x11bafb:    mov     rdi, r14
0x11bafe:    call    0x118d10
0x11c279:    mov     qword ptr [rsp + 0x10], rax
0x11c27e:    jmp     0x11baeb
0x11c3c2:    mov     edx, dword ptr [rsp + 0x6c]
0x11c3c6:    mov     ecx, 2
0x11c3cb:    mov     esi, dword ptr [rsp + 0x44]
0x11c3cf:    jmp     0x11b9f7
0x11beba:    mov     rsi, qword ptr [rip + 0xfc57]
0x11bec1:    movsxd  rbx, r14d
0x11bec4:    lea     rdi, [rbp + rbx]
0x11bec9:    movzx   eax, byte ptr [rsi]
0x11becc:    cmp     byte ptr [rdi], al
0x11bece:    jne     0x11bea8
0x11bb8d:    mov     r13, rax
0x11bb90:    test    rax, rax
0x11bb93:    jne     0x11ba6a
0x11baa0:    mov     dword ptr [rsp + 0x6c], r14d
0x11baa5:    movsxd  rcx, r14d
0x11baa8:    cmp     byte ptr [rbp + rcx], 0x3a
0x11baad:    je      0x11b9a0
0x11bb3d:    test    rax, rax
0x11bb40:    je      0x11bb70
0x11ba5d:    lea     r13, [rax + 1]

```

```

0x11ba61:    test    rax, rax
0x11ba64:    je      0x11bb70
0x11bbc8:    cmp     r14d, eax
0x11bbcb:    jge     0x11baa0
0x11bb65:    test    rax, rax
0x11bb68:    jne     0x11bb42
0x11ba0e:    mov     rdi, r14
0x11ba11:    call    0x118d10
0x11bb03:    add     dword ptr [rsp + 0x3c], 1
0x11bb08:    cmp     byte ptr [r12], 0
0x11bb0d:    mov     r13d, dword ptr [rsp + 0x6c]
0x11bb12:    jne     0x11c098
0x11b9f7:    mov     rdi, rbp
0x11b9fa:    xor     eax, eax
0x11b9fc:    call    0x119fe0
0x11bed0:    movsxd  rdx, r13d
0x11bed3:    call    0x104890
0x11bea8:    add     r14d, 1
0x11beac:    lea     eax, [r13 + r14]
0x11beb1:    cmp     eax, r15d
0x11beb4:    jg      0x11c050
0x11bb99:    jmp     0x11bb70
0x11ba6a:    mov     rdi, r13
0x11ba6d:    call    0x1049b0
0x11b9a0:    movzx   eax, byte ptr [rbp + rcx + 1]
0x11b9a5:    cmp     al, 0x67
0x11b9a7:    jne     0x11b9bf
0x11bab3:    xor     eax, eax
0x11bab5:    cmp     dword ptr [rsp + 0x38], 1
0x11baba:    mov     edx, r14d
0x11babd:    mov     r14, r15
0x11bac0:    sete    al
0x11bac3:    mov     r15, qword ptr [rsp + 0x28]
0x11bac8:    movsxd  rbx, dword ptr [rsp + 0x40]
0x11bacd:    mov     dword ptr [rsp + 0x28], eax
0x11bad1:    sub     edx, 1
0x11bad4:    mov     rdi, r14
0x11bad7:    mov     dword ptr [rsp + 0x6c], edx
0x11badb:    call    0x1049b0
0x11bb70:    mov     eax, dword ptr [rsp + 0x6c]
0x11bb74:    lea     r14d, [rax + 2]
0x11bb78:    jmp     0x11baa0
0x11bb42:    mov     byte ptr [rax], 0
0x11bb45:    mov     eax, dword ptr [rsp + 0x6c]
0x11bb49:    lea     r14d, [rax + 2]
0x11bb4d:    jmp     0x11baa0
0x11bbd1:    movsxd  rax, ebx
0x11bbd4:    lea     r14, [rsp + 0x6c]

```

```

0x11bbd9:      add     ebx, 3
0x11bbdc:      xor     ecx, ecx
0x11bbde:      movsx   r13d, byte ptr [rbp + rax + 2]
0x11bbe4:      lea     r8, [rip + 0xff21]
0x11bbeb:      mov     rsi, r14
0x11bbee:      mov     rdi, rbp
0x11bbf1:      xor     eax, eax
0x11bbf3:      mov     dword ptr [rsp + 0x6c], ebx
0x11bbf7:      mov     edx, r13d
0x11bbfa:      call    0x119c70
0x11bb6a:      nop     word ptr [rax + rax]
0x11bb18:      mov     rdi, qword ptr [rsp + 0x10]
0x11bb1d:      call    0x118d10
0x11c098:      mov     rdi, qword ptr [rsp + 0x10]
0x11c09d:      call    0x1049b0
0x11bed8:      test    eax, eax
0x11beda:      jne     0x11bea8
0x11c050:      cmp     dword ptr [rsp + 0x18], 1
0x11c055:      mov     r15, rbp
0x11c058:      mov     r14d, dword ptr [rsp + 0x6c]
0x11c05d:      mov     rbp, qword ptr [rsp + 0x48]
0x11c062:      jle     0x11c417
0x11ba72:      lea     rdi, [rax + 1]
0x11ba76:      call    0x11d670
0x11bbff:      mov     rbx, rax
0x11bc02:      test    rax, rax
0x11bc05:      je      0x11c387
0x11bb22:      jmp     0x11b74d
0x11c0a2:      lea     r12d, [rbx + rax]
0x11c0a6:      mov     eax, dword ptr [rsp + 4]
0x11c0aa:      cmp     r12d, eax
0x11c0ad:      jl      0x11c0e0
0x11bedc:      mov     ecx, dword ptr [rip + 0xfc26]
0x11bee2:      add     ecx, r15d
0x11bee5:      sub     ecx, r13d
0x11bee8:      lea     edi, [rcx + 1]
0x11beeb:      mov     dword ptr [rsp + 0x50], ecx
0x11beef:      movsxd  rdi, edi
0x11bef2:      call    0x11d670
0x11c417:      mov     esi, dword ptr [rsp + 0x50]
0x11c41b:      test    esi, esi
0x11c41d:      je      0x11baa5
0x11c068:      mov     dword ptr [rsp + 0x18], 0
0x11c070:      jmp     0x11baa5
0x11ba7b:      mov     rsi, r13
0x11ba7e:      mov     rdi, rax
0x11ba81:      call    0x1048b0
0x11c387:      cmp     qword ptr [rip + 0xf789], 0

```

```

0x11c38f:    mov     eax, dword ptr [rip + 0xf777]
0x11c395:    jne     0x11bc29
0x11bc0b:    mov     rdi, qword ptr [rip + 0xff06]
0x11bc12:    test    rdi, rdi
0x11bc15:    je      0x11bc1c
0x11c0e0:    movsxd  rdi, ebx
0x11c0e3:    mov     rsi, qword ptr [rsp + 0x10]
0x11c0e8:    movsxd  rbx, r12d
0x11c0eb:    add     rdi, r15
0x11c0ee:    call    0x1048b0
0x11c0af:    movsxd  r14, eax
0x11c0b2:    mov     rdi, r15
0x11c0b5:    mov     r15d, eax
0x11c0b8:    sub     r14, -0x80
0x11c0bc:    nop     dword ptr [rax]
0x11c0c0:    mov     rsi, r14
0x11c0c3:    sub     r15d, -0x80
0x11c0c7:    sub     r14, -0x80
0x11c0cb:    call    0x11d690
0x11bef7:    mov     rdx, rbx
0x11befa:    mov     rsi, rbp
0x11befd:    mov     rdi, rax
0x11bf00:    mov     r13, rax
0x11bf03:    call    0x104880
0x11baa5:    movsxd  rcx, r14d
0x11baa8:    cmp     byte ptr [rbp + rcx], 0x3a
0x11baad:    je      0x11b9a0
0x11c423:    mov     edx, r14d
0x11c426:    mov     ecx, 2
0x11c42b:    mov     r14, r15
0x11c42e:    mov     r15, qword ptr [rsp + 0x28]
0x11c433:    jmp     0x11c3cb
0x11ba86:    mov     rdi, r15
0x11ba89:    mov     rbx, rax
0x11ba8c:    call    0x118d10
0x11c39b:    mov     rbx, qword ptr [rip + 0xf74e]
0x11c3a2:    test    rbx, rbx
0x11c3a5:    je      0x11c3b0
0x11bc29:    test    eax, eax
0x11bc2b:    je      0x11c400
0x11bc1c:    mov     qword ptr [rip + 0xfef5], rbx
0x11bc23:    mov     eax, dword ptr [rip + 0xfef3]
0x11bc29:    test    eax, eax
0x11bc2b:    je      0x11c400
0x11bc17:    call    0x118d10
0x11c0f3:    jmp     0x11bb18
0x11c0d0:    mov     rdi, rax
0x11c0d3:    cmp     r12d, r15d

```

```

0x11c0d6:    jge     0x11c0c0
0x11bf08:    movsxd  rdx, dword ptr [rip + 0xfbf9]
0x11bf0f:    lea     rdi, [r13 + rbx]
0x11bf14:    mov     rsi, qword ptr [rip + 0xfbf5]
0x11bf1b:    call    0x104880
0x11c3cb:    mov     esi, dword ptr [rsp + 0x44]
0x11c3cf:    jmp     0x11b9f7
0x11ba91:    mov     eax, dword ptr [rsp + 0x6c]
0x11ba95:    mov     r15, rbx
0x11ba98:    lea     r14d, [rax + 2]
0x11ba9c:    nop     dword ptr [rax]
0x11baa0:    mov     dword ptr [rsp + 0x6c], r14d
0x11baa5:    movsxd  rcx, r14d
0x11baa8:    cmp     byte ptr [rbp + rcx], 0x3a
0x11baad:    je      0x11b9a0
0x11c3b0:    mov     dword ptr [rip + 0xf752], 0
0x11c3ba:    mov     r14, r15
0x11c3bd:    mov     r15, qword ptr [rsp + 0x28]
0x11c3c2:    mov     edx, dword ptr [rsp + 0x6c]
0x11c3c6:    mov     ecx, 2
0x11c3cb:    mov     esi, dword ptr [rsp + 0x44]
0x11c3cf:    jmp     0x11b9f7
0x11c3a7:    cmp     byte ptr [rbx], 0
0x11c3aa:    jne     0x11c442
0x11bc31:    mov     rdi, qword ptr [rip + 0xfed8]
0x11bc38:    test    rdi, rdi
0x11bc3b:    je      0x11bc42
0x11c0d8:    mov     dword ptr [rsp + 4], r15d
0x11c0dd:    mov     r15, rax
0x11c0e0:    movsxd  rdi, ebx
0x11c0e3:    mov     rsi, qword ptr [rsp + 0x10]
0x11c0e8:    movsxd  rbx, r12d
0x11c0eb:    add     rdi, r15
0x11c0ee:    call    0x1048b0
0x11c0c0:    mov     rsi, r14
0x11c0c3:    sub     r15d, -0x80
0x11c0c7:    sub     r14, -0x80
0x11c0cb:    call    0x11d690
0x11bf20:    movsxd  rsi, dword ptr [rip + 0xfbe5]
0x11bf27:    movsxd  rdi, dword ptr [rip + 0xfbda]
0x11bf2e:    lea     edx, [rsi + r14]
0x11bf32:    add     rdi, rbx
0x11bf35:    add     rsi, rbx
0x11bf38:    sub     r15d, edx
0x11bf3b:    add     rsi, rbp
0x11bf3e:    add     rdi, r13
0x11bf41:    movsxd  rdx, r15d
0x11bf44:    call    0x104880

```

```

0x11c442:    mov     rdi, rbx
0x11c445:    call    0x1049b0
0x11bc42:    mov     rsi, r14
0x11bc45:    mov     rdi, rbp
0x11bc48:    xor     eax, eax
0x11bc4a:    mov     ecx, 1
0x11bc4f:    lea     r8, [rip + 0xfeb2]
0x11bc56:    mov     edx, r13d
0x11bc59:    call    0x119c70
0x11bc3d:    call    0x118d10
0x11bf49:    movsxd  rcx, dword ptr [rsp + 0x50]
0x11bf4e:    mov     rdi, rbp
0x11bf51:    mov     byte ptr [r13 + rcx], 0
0x11bf57:    call    0x118d10
0x11c44a:    lea     rdi, [rax + 1]
0x11c44e:    call    0x11d670
0x11bc5e:    mov     esi, 0x26
0x11bc63:    mov     rdi, rax
0x11bc66:    mov     qword ptr [rip + 0xfea3], rax
0x11bc6d:    call    0x104a00
0x11bf5c:    mov     edi, dword ptr [rsp + 0x18]
0x11bf60:    test    edi, edi
0x11bf62:    jne     0x11c1a8
0x11c453:    mov     rsi, rbx
0x11c456:    mov     rdi, rax
0x11c459:    call    0x1048b0
0x11bc72:    test    rax, rax
0x11bc75:    je      0x11be75
0x11bf68:    mov     rbp, qword ptr [rsp + 0x48]
0x11bf6d:    mov     r14d, dword ptr [rsp + 0x6c]
0x11bf72:    mov     r15, r13
0x11bf75:    jmp     0x11baa5
0x11c1a8:    mov     rdi, r13
0x11c1ab:    mov     rbp, r13
0x11c1ae:    add     r14d, dword ptr [rip + 0xf953]
0x11c1b5:    call    0x1049b0
0x11c45e:    mov     rdi, rax
0x11c461:    mov     qword ptr [rip + 0xf6b0], rax
0x11c468:    call    0x1049b0
0x11be75:    mov     rdi, r15
0x11be78:    call    0x1049b0
0x11bc7b:    mov     edx, dword ptr [rip + 0xfe8b]
0x11bc81:    add     edx, dword ptr [rip + 0xfe81]
0x11bc87:    movsxd  rdi, edx
0x11bc8a:    mov     dword ptr [rsp + 0x48], edx
0x11bc8e:    call    0x11d670
0x11c1ba:    add     dword ptr [rsp + 0x18], 1
0x11c1bf:    mov     r13d, dword ptr [rip + 0xf946]

```



```

0x11c1c6:    mov     dword ptr [rsp + 0x50], 0
0x11c1ce:    mov     r15d, eax
0x11c1d1:    jmp     0x11bea8
0x11c46d:    mov     dword ptr [rip + 0xf699], eax
0x11c473:    jmp     0x11bc29
0x11bc93:    mov     r9d, dword ptr [rip + 0xfe6e]
0x11bc9a:    mov     r10, qword ptr [rip + 0xfe6f]
0x11bca1:    mov     r8, rax
0x11bca4:    test    r9d, r9d
0x11bca7:    jle     0x11c2d0
0x11c2d0:    xor     ebx, ebx
0x11c2d2:    mov     byte ptr [rax], 0
0x11c2d5:    mov     rdi, r10
0x11c2d8:    mov     qword ptr [rsp + 0x48], r8
0x11c2dd:    call    0x118d10
0x11bcad:    mov     qword ptr [rsp + 0x50], rbp
0x11bcb2:    xor     ebx, ebx
0x11bcb4:    xor     r13d, r13d
0x11bcb7:    mov     rbp, rax
0x11bcba:    mov     qword ptr [rsp + 0x58], r15
0x11bcbf:    mov     r15d, dword ptr [rsp + 0x48]
0x11bcc4:    jmp     0x11bd01
0x11c2e2:    mov     r8, qword ptr [rsp + 0x48]
0x11c2e7:    mov     dword ptr [rip + 0xf81b], ebx
0x11c2ed:    mov     qword ptr [rip + 0xf81c], r8
0x11c2f4:    jmp     0x11be75
0x11bd01:    movsxd  r14, r13d
0x11bd04:    movzx   eax, byte ptr [r10 + r14]
0x11bd09:    cmp     al, 0x26
0x11bd0b:    jne     0x11bcd0
0x11bd0d:    mov     eax, dword ptr [rip + 0xfdf9]
0x11bd13:    lea     ecx, [rax + rbx]
0x11bd16:    cmp     ecx, r15d
0x11bd19:    jge     0x11c33d
0x11bcd0:    cmp     al, 0x5c
0x11bcd2:    je      0x11c300
0x11bd1f:    movsxd  rdi, ebx
0x11bd22:    mov     rsi, qword ptr [rip + 0xfdef]
0x11bd29:    mov     qword ptr [rsp + 0x48], r10
0x11bd2e:    add     rdi, rbp
0x11bd31:    call    0x1048b0
0x11c33d:    lea     r15d, [rax + r15*2]
0x11c341:    mov     rdi, rbp
0x11c344:    movsxd  rsi, r15d
0x11c347:    call    0x11d690
0x11c300:    cmp     byte ptr [r10 + r14 + 1], 0x26
0x11c306:    lea     rax, [r14 + 1]
0x11c30a:    jne     0x11bcd8

```

```

0x11bcd8:    cmp     r15d, ebx
0x11bcd9:    jle     0x11c320
0x11bd36:    mov     r10, qword ptr [rsp + 0x48]
0x11bd3b:    add     ebx, dword ptr [rip + 0xfdc9]
0x11bd41:    jmp     0x11bcf0
0x11c34c:    mov     r10, qword ptr [rip + 0xf7bd]
0x11c353:    mov     rbp, rax
0x11c356:    jmp     0x11bd1f
0x11c310:    add     r13d, 1
0x11c314:    mov     r14, rax
0x11c317:    jmp     0x11bcd8
0x11c320:    add     r15d, r15d
0x11c323:    mov     rdi, rbp
0x11c326:    movsxd  rsi, r15d
0x11c329:    call    0x11d690
0x11bce1:    movzx   ecx, byte ptr [r10 + r14]
0x11bce6:    movsxd  rax, ebx
0x11bce9:    add     ebx, 1
0x11bcec:    mov     byte ptr [rbp + rax], cl
0x11bcf0:    add     r13d, 1
0x11bcf4:    cmp     r13d, dword ptr [rip + 0xfe0d]
0x11bcfb:    jge     0x11c36f
0x11bcf0:    add     r13d, 1
0x11bcf4:    cmp     r13d, dword ptr [rip + 0xfe0d]
0x11bcfb:    jge     0x11c36f
0x11c32e:    mov     r10, qword ptr [rip + 0xf7db]
0x11c335:    mov     rbp, rax
0x11c338:    jmp     0x11bce1
0x11c36f:    mov     r8, rbp
0x11c372:    movsxd  rax, ebx
0x11c375:    mov     r15, qword ptr [rsp + 0x58]
0x11c37a:    mov     rbp, qword ptr [rsp + 0x50]
0x11c37f:    add     rax, r8
0x11c382:    jmp     0x11c2d2
0x11c2d2:    mov     byte ptr [rax], 0
0x11c2d5:    mov     rdi, r10
0x11c2d8:    mov     qword ptr [rsp + 0x48], r8
0x11c2dd:    call    0x118d10
0x11b6a2:    nop     word ptr [rax + rax]
0x11b6d5:    nop     dword ptr [rax]
0x11b70d:    nop     dword ptr [rax]
0x11b829:    nop     dword ptr [rax]
0x11b9db:    nop     dword ptr [rax + rax]
0x11ba4b:    nop     dword ptr [rax + rax]
0x11bb27:    nop     word ptr [rax + rax]
0x11bb52:    nop     word ptr [rax + rax]
0x11bb7d:    nop     dword ptr [rax]
0x11bb9b:    nop     dword ptr [rax + rax]

```

```

0x11bbb2:    nop    word ptr [rax + rax]
0x11bcc6:    nop    word ptr cs:[rax + rax]
0x11bd43:    nop    dword ptr [rax + rax]
0x11bd78:    nop    dword ptr [rax + rax]
0x11bdda:    nop    word ptr [rax + rax]
0x11be0a:    nop    word ptr [rax + rax]
0x11be2a:    nop    word ptr [rax + rax]
0x11be69:    nop    dword ptr [rax]
0x11bea5:    nop    dword ptr [rax]
0x11bf7a:    nop    word ptr [rax + rax]
0x11bfbc:    nop    dword ptr [rax]
0x11bfdb:    nop    dword ptr [rax + rax]
0x11bfea:    nop    word ptr [rax + rax]
0x11c00a:    nop    word ptr [rax + rax]
0x11c047:    nop    word ptr [rax + rax]
0x11c075:    nop    dword ptr [rax]
0x11c095:    nop    dword ptr [rax]
0x11c0f8:    nop    dword ptr [rax + rax]
0x11c146:    nop    word ptr cs:[rax + rax]
0x11c1a4:    nop    dword ptr [rax]
0x11c1d6:    nop    word ptr cs:[rax + rax]
0x11c24f:    nop
0x11c2cf:    nop
0x11c2f9:    nop    dword ptr [rax]
0x11c31c:    nop    dword ptr [rax]
0x11c48c:    nop    dword ptr [rax]
0x11c490:    endbr64
0x11c494:    push   rbx
0x11c495:    mov    rbx, rdx
0x11c498:    add    rsp, -0x80
0x11c49c:    mov    rax, qword ptr fs:[0x28]
0x11c4a5:    mov    qword ptr [rsp + 0x78], rax
0x11c4aa:    xor    eax, eax
0x11c4ac:    mov    rsi, rsp
0x11c4af:    call   0x104cb0
0x11c4b4:    test   eax, eax
0x11c4b6:    js     0x11c514
0x11c514:    mov    eax, 0xffffffff
0x11c519:    jmp    0x11c4fe
0x11c4b8:    mov    rax, qword ptr [rsp + 8]
0x11c4bd:    test   rax, rax
0x11c4c0:    cmovne rax, qword ptr [rsp]
0x11c4c5:    mov    qword ptr [rbx], rax
0x11c4c8:    mov    rax, qword ptr [rsp + 0x10]
0x11c4cd:    mov    qword ptr [rbx + 8], rax
0x11c4d1:    mov    rax, qword ptr [rsp + 0x18]
0x11c4d6:    mov    qword ptr [rbx + 0x10], rax
0x11c4da:    mov    rax, qword ptr [rsp + 0x20]

```

```

0x11c4df:    mov     qword ptr [rbx + 0x18], rax
0x11c4e3:    shr     rax, 0x3f
0x11c4e7:    mov     byte ptr [rbx + 0x20], al
0x11c4ea:    mov     rax, qword ptr [rsp + 0x28]
0x11c4ef:    mov     qword ptr [rbx + 0x28], rax
0x11c4f3:    mov     rax, qword ptr [rsp + 0x30]
0x11c4f8:    mov     qword ptr [rbx + 0x30], rax
0x11c4fc:    xor     eax, eax
0x11c4fe:    mov     rdx, qword ptr [rsp + 0x78]
0x11c503:    sub     rdx, qword ptr fs:[0x28]
0x11c50c:    jne     0x11c51b
0x11c4fe:    mov     rdx, qword ptr [rsp + 0x78]
0x11c503:    sub     rdx, qword ptr fs:[0x28]
0x11c50c:    jne     0x11c51b
0x11c50e:    sub     rsp, -0x80
0x11c512:    pop     rbx
0x11c513:    ret
0x11c51b:    call    0x1049d0
0x11c520:    endbr64
0x11c524:    push    r15
0x11c526:    push    r14
0x11c528:    push    r13
0x11c52a:    mov     r13, r8
0x11c52d:    push    r12
0x11c52f:    mov     r12, rcx
0x11c532:    push    rbp
0x11c533:    push    rbx
0x11c534:    mov     rbx, rdi
0x11c537:    sub     rsp, 0x98
0x11c53e:    mov     qword ptr [rsp], rsi
0x11c542:    mov     qword ptr [rsp + 0x18], r8
0x11c547:    mov     dword ptr [rsp + 0x48], edx
0x11c54b:    mov     rax, qword ptr fs:[0x28]
0x11c554:    mov     qword ptr [rsp + 0x88], rax
0x11c55c:    mov     eax, edx
0x11c55e:    and     edx, 0x20
0x11c561:    mov     dword ptr [rsp + 0x30], edx
0x11c565:    and     eax, 3
0x11c568:    cmp     edx, 1
0x11c56b:    mov     dword ptr [rsp + 0x10], eax
0x11c56f:    sbb     eax, eax
0x11c571:    and     eax, 0xffffffff8
0x11c574:    add     eax, 0x400
0x11c579:    mov     dword ptr [rsp + 0x34], eax
0x11c57d:    call    0x104900
0x11c582:    mov     r15, qword ptr [rax]
0x11c585:    mov     r14, rax
0x11c588:    mov     rdi, r15

```

```

0x11c58b:    call    0x1049b0
0x11c590:    mov     rcx, qword ptr [r14 + 0x10]
0x11c594:    mov     r14, qword ptr [r14 + 8]
0x11c598:    mov     rbp, rax
0x11c59b:    sub     rax, 1
0x11c59f:    cmp     rax, 0x10
0x11c5a3:    mov     eax, 1
0x11c5a8:    mov     rdi, r14
0x11c5ab:    mov     qword ptr [rsp + 0x38], rcx
0x11c5b0:    cmovae  rbp, rax
0x11c5b4:    lea     rax, [rip + 0x5123]
0x11c5bb:    cmovae  r15, rax
0x11c5bf:    call    0x1049b0
0x11c5c4:    cmp     rax, 0x10
0x11c5c8:    lea     rax, [rip + 0x5e16]
0x11c5cf:    cmova   r14, rax
0x11c5d3:    mov     rax, qword ptr [rsp]
0x11c5d7:    add     rax, 0x288
0x11c5dd:    mov     qword ptr [rsp + 8], rax
0x11c5e2:    cmp     r13, r12
0x11c5e5:    ja      0x11c740
0x11c5eb:    mov     rax, r12
0x11c5ee:    xor     edx, edx
0x11c5f0:    div     r13
0x11c5f3:    test    rdx, rdx
0x11c5f6:    jne     0x11c604
0x11c740:    test    r12, r12
0x11c743:    je      0x11c604
0x11c5f8:    mul     rbx
0x11c5fb:    mov     rcx, rax
0x11c5fe:    jno     0x11cb98
0x11c604:    mov     qword ptr [rsp + 0x20], r12
0x11c609:    fild    qword ptr [rsp + 0x20]
0x11c60d:    test    r12, r12
0x11c610:    js      0x11c928
0x11c749:    mov     rax, qword ptr [rsp + 0x18]
0x11c74e:    xor     edx, edx
0x11c750:    div     r12
0x11c753:    mov     r8, rax
0x11c756:    test    rdx, rdx
0x11c759:    jne     0x11c604
0x11cb98:    mov     r10d, dword ptr [rsp + 0x48]
0x11cb9d:    xor     esi, esi
0x11cb9f:    xor     edi, edi
0x11cba1:    and     r10d, 0x10
0x11cba5:    jne     0x11c79c
0x11c928:    fadd    dword ptr [rip + 0x5bfe]
0x11c92e:    jmp     0x11c616

```

```

0x11c616:    mov     rax, qword ptr [rsp + 0x18]
0x11c61b:    mov     qword ptr [rsp + 0x20], rax
0x11c620:    fild    qword ptr [rsp + 0x20]
0x11c624:    test    rax, rax
0x11c627:    js      0x11c908
0x11c75f:    mov     rax, rbx
0x11c762:    xor     edx, edx
0x11c764:    div     r8
0x11c767:    mov     rcx, rax
0x11c76a:    lea     rax, [rdx + rdx*4]
0x11c76e:    xor     edx, edx
0x11c770:    add     rax, rax
0x11c773:    div     r8
0x11c776:    add     rdx, rdx
0x11c779:    mov     edi, eax
0x11c77b:    cmp     r8, rdx
0x11c77e:    jbe     0x11cde8
0x11cbab:    mov     ebx, 0xffffffff
0x11cbb0:    jmp     0x11c7af
0x11c79c:    mov     r8d, dword ptr [rsp + 0x34]
0x11c7a1:    xor     ebx, ebx
0x11c7a3:    mov     r11, r8
0x11c7a6:    cmp     r8, rcx
0x11c7a9:    jbe     0x11cacf
0x11c908:    fadd    dword ptr [rip + 0x5c1e]
0x11c90e:    mov     qword ptr [rsp + 0x20], rbx
0x11c913:    fdivp   st(1)
0x11c915:    fild    qword ptr [rsp + 0x20]
0x11c919:    test    rbx, rbx
0x11c91c:    jns     0x11c641
0x11c62d:    mov     qword ptr [rsp + 0x20], rbx
0x11c632:    fdivp   st(1)
0x11c634:    fild    qword ptr [rsp + 0x20]
0x11c638:    test    rbx, rbx
0x11c63b:    js      0x11c8f0
0x11cde8:    mov     esi, 2
0x11cded:    mov     eax, 3
0x11cdf2:    cmovb   esi, eax
0x11cdf5:    jmp     0x11c78d
0x11c784:    xor     esi, esi
0x11c786:    test    rdx, rdx
0x11c789:    setne   sil
0x11c78d:    mov     r10d, dword ptr [rsp + 0x48]
0x11c792:    and     r10d, 0x10
0x11c796:    je      0x11cbab
0x11c7af:    cmp     dword ptr [rsp + 0x10], 1
0x11c7b4:    mov     r8, qword ptr [rsp + 8]
0x11c7b9:    je      0x11cdc0

```

```

0x11cacf:    mov     rax, rcx
0x11cad2:    xor     edx, edx
0x11cad4:    mov     ecx, esi
0x11cad6:    div     r8
0x11cad9:    sar     ecx, 1
0x11cadb:    mov     r9, rax
0x11cade:    lea     eax, [rdx + rdx*4]
0x11cae1:    xor     edx, edx
0x11cae3:    lea     eax, [rdi + rax*2]
0x11cae6:    div     r11d
0x11cae9:    lea     edx, [rcx + rdx*2]
0x11caec:    mov     edi, eax
0x11caee:    mov     rcx, r9
0x11caf1:    add     esi, edx
0x11caf3:    cmp     r11d, edx
0x11caf6:    ja      0x11cab0
0x11c922:    jmp     0x11c8f0
0x11c641:    fmulp   st(1)
0x11c643:    test    byte ptr [rsp + 0x48], 0x10
0x11c648:    je      0x11c938
0x11c8f0:    fadd    dword ptr [rip + 0x5c36]
0x11c8f6:    fmulp   st(1)
0x11c8f8:    test    byte ptr [rsp + 0x48], 0x10
0x11c8fd:    jne     0x11c64e
0x11c78d:    mov     r10d, dword ptr [rsp + 0x48]
0x11c792:    and     r10d, 0x10
0x11c796:    je      0x11cbab
0x11cdc0:    mov     rax, rcx
0x11cdc3:    movsxd  rsi, esi
0x11cdc6:    and     eax, 1
0x11cdc9:    add     rax, rsi
0x11cdcc:    setne   al
0x11cdcf:    movzx   eax, al
0x11cdd2:    add     eax, edi
0x11cdd4:    cmp     eax, 5
0x11cdd7:    jg      0x11c7cf
0x11c7bf:    mov     r11d, dword ptr [rsp + 0x10]
0x11c7c4:    test    r11d, r11d
0x11c7c7:    jne     0x11c7e8
0x11caf8:    cmp     r11d, esi
0x11cafb:    setb    sil
0x11caff:    movzx   esi, sil
0x11cb03:    add     esi, 2
0x11cb06:    jmp     0x11caba
0x11cab0:    test    esi, esi
0x11cab2:    setne   sil
0x11cab6:    movzx   esi, sil
0x11caba:    add     ebx, 1

```

```

0x11cabd:    cmp     r8, r9
0x11cac0:    ja     0x11cf18
0x11c938:    cmp     dword ptr [rsp + 0x10], 1
0x11c93d:    je     0x11c9ca
0x11c64e:    fild    dword ptr [rsp + 0x34]
0x11c652:    xor     ebx, ebx
0x11c654:    fld     st(0)
0x11c656:    jmp     0x11c664
0x11c903:    jmp     0x11c938
0x11cddd:    jmp     0x11c7e8
0x11c7cf:    add     rcx, 1
0x11c7d3:    test    r10d, r10d
0x11c7d6:    je     0x11c7e8
0x11c7c9:    add     esi, edi
0x11c7cb:    test    esi, esi
0x11c7cd:    jle     0x11c7e8
0x11c7e8:    movabs  rsi, 0xffffffffffffffff
0x11c7f2:    mov     rbp, r8
0x11c7f5:    nop     dword ptr [rax]
0x11c7f8:    mov     rax, rcx
0x11c7fb:    sub     rbp, 1
0x11c7ff:    mul     rsi
0x11c802:    mov     rax, rcx
0x11c805:    shr     rdx, 3
0x11c809:    lea     rdi, [rdx + rdx*4]
0x11c80d:    add     rdi, rdi
0x11c810:    sub     rax, rdi
0x11c813:    add     eax, 0x30
0x11c816:    mov     byte ptr [rbp], al
0x11c819:    mov     rax, rcx
0x11c81c:    mov     rcx, rdx
0x11c81f:    cmp     rax, 9
0x11c823:    ja     0x11c7f8
0x11caba:    add     ebx, 1
0x11cabd:    cmp     r8, r9
0x11cac0:    ja     0x11cf18
0x11cac6:    cmp     ebx, 8
0x11cac9:    je     0x11c7af
0x11cf18:    cmp     r9, 9
0x11cf1c:    ja     0x11c7af
0x11c9ca:    sub     rsp, 0x10
0x11c9ce:    mov     rdx, -1
0x11c9d5:    mov     esi, 1
0x11c9da:    xor     eax, eax
0x11c9dc:    mov     rbx, qword ptr [rsp + 0x10]
0x11c9e1:    lea     rcx, [rip + 0x5adc]
0x11c9e8:    fstp    xword ptr [rsp]
0x11c9eb:    mov     rdi, rbx

```



```

0x11c9ee:    call    0x104f10
0x11c943:    fld     xword ptr [rip + 0x5bf7]
0x11c949:    fcomip  st(1)
0x11c94b:    jbe     0x11c9ca
0x11c664:    fld     st(0)
0x11c666:    add     ebx, 1
0x11c669:    fmul    st(2)
0x11c66b:    fxch    st(3)
0x11c66d:    fcomi   st(3)
0x11c66f:    jb      0x11c680
0x11c7d8:    mov     eax, dword ptr [rsp + 0x34]
0x11c7dc:    cmp     rax, rcx
0x11c7df:    je      0x11cec2
0x11c825:    test    byte ptr [rsp + 0x48], 4
0x11c82a:    je      0x11ca2c
0x11c7f8:    mov     rax, rcx
0x11c7fb:    sub     rbp, 1
0x11c7ff:    mul     rsi
0x11c802:    mov     rax, rcx
0x11c805:    shr     rdx, 3
0x11c809:    lea     rdi, [rdx + rdx*4]
0x11c80d:    add     rdi, rdi
0x11c810:    sub     rax, rdi
0x11c813:    add     eax, 0x30
0x11c816:    mov     byte ptr [rbp], al
0x11c819:    mov     rax, rcx
0x11c81c:    mov     rcx, rdx
0x11c81f:    cmp     rax, 9
0x11c823:    ja      0x11c7f8
0x11cf22:    cmp     dword ptr [rsp + 0x10], 1
0x11cf27:    je      0x11cfa0
0x11c9f3:    mov     rdi, rbx
0x11c9f6:    mov     ebx, 0xffffffff
0x11c9fb:    call    0x1049b0
0x11c94d:    fld     dword ptr [rip + 0x5bdd]
0x11c953:    fxch    st(1)
0x11c955:    fcomi   st(1)
0x11c957:    jae     0x11ce60
0x11c680:    fstp    st(2)
0x11c682:    fstp    st(2)
0x11c684:    lea     r15, [rbp + 1]
0x11c688:    fdivrp  st(1)
0x11c68a:    cmp     dword ptr [rsp + 0x30], 1
0x11c68f:    adc     rbp, 2
0x11c693:    cmp     dword ptr [rsp + 0x10], 1
0x11c698:    je      0x11cc70
0x11c671:    cmp     ebx, 8
0x11c674:    jne     0x11c660

```

```

0x11cec2:    cmp     ebx, 8
0x11cec5:    je      0x11c7e8
0x11c7e5:    nop     dword ptr [rax]
0x11ca2c:    test    byte ptr [rsp + 0x48], 0x80
0x11ca31:    je      0x11ca4f
0x11c830:    mov     rdi, r14
0x11c833:    mov     r12, r8
0x11c836:    mov     qword ptr [rsp + 0x40], r8
0x11c83b:    mov     r13, -1
0x11c842:    call    0x1049b0
0x11cfa0:    mov     edx, eax
0x11cfa2:    and     edx, 1
0x11cfa5:    add     edx, esi
0x11cfa7:    cmp     edx, 2
0x11cfaa:    jg      0x11cf37
0x11cf29:    mov     r12d, dword ptr [rsp + 0x10]
0x11cf2e:    test    r12d, r12d
0x11cf31:    jne     0x11cfac
0x11ca00:    pop     rsi
0x11ca01:    pop     rdi
0x11ca02:    mov     rdx, rax
0x11ca05:    mov     r12, rax
0x11ca08:    mov     rbp, qword ptr [rsp + 8]
0x11ca0d:    mov     rsi, qword ptr [rsp]
0x11ca11:    sub     rbp, rdx
0x11ca14:    mov     rdi, rbp
0x11ca17:    call    0x104d40
0x11c95d:    fstp    st(1)
0x11c95f:    fnstcw  word ptr [rsp + 0x4e]
0x11c963:    movzx   eax, word ptr [rsp + 0x4e]
0x11c968:    or      ah, 0xc
0x11c96b:    mov     word ptr [rsp + 0x4c], ax
0x11c970:    fld     st(0)
0x11c972:    fldcw   word ptr [rsp + 0x4c]
0x11c976:    fistp   qword ptr [rsp + 0x20]
0x11c97a:    fldcw   word ptr [rsp + 0x4e]
0x11c97e:    mov     rax, qword ptr [rsp + 0x20]
0x11c983:    mov     qword ptr [rsp + 0x20], rax
0x11c988:    fild    qword ptr [rsp + 0x20]
0x11c98c:    test    rax, rax
0x11c98f:    jns     0x11c997
0x11ce60:    fnstcw  word ptr [rsp + 0x4e]
0x11ce64:    fsubr    st(1), st(0)
0x11ce66:    fxch    st(1)
0x11ce68:    movzx   eax, word ptr [rsp + 0x4e]
0x11ce6d:    or      ah, 0xc
0x11ce70:    mov     word ptr [rsp + 0x4c], ax
0x11ce75:    fldcw   word ptr [rsp + 0x4c]

```

```

0x11ce79:      fistp    qword ptr [rsp + 0x20]
0x11ce7d:      fldcw   word ptr [rsp + 0x4e]
0x11ce81:      mov     rax, qword ptr [rsp + 0x20]
0x11ce86:      btc     rax, 0x3f
0x11ce8b:      jmp     0x11c983
0x11cc70:      sub     rsp, 0x10
0x11cc74:      mov     rdx, -1
0x11cc7b:      mov     esi, 1
0x11cc80:      xor     eax, eax
0x11cc82:      fld     st(0)
0x11cc84:      fstp     xword ptr [rsp]
0x11cc87:      mov     r12, qword ptr [rsp + 0x10]
0x11cc8c:      lea     rcx, [rip + 0x5837]
0x11cc93:      mov     rdi, r12
0x11cc96:      fstp     xword ptr [rsp + 0x30]
0x11cc9a:      call    0x104f10
0x11c69e:      fld     xword ptr [rip + 0x5e9c]
0x11c6a4:      fcomip   st(1)
0x11c6a6:      jbe     0x11cb10
0x11c676:      fstp     st(2)
0x11c678:      fstp     st(2)
0x11c67a:      jmp     0x11c684
0x11c660:      fstp     st(1)
0x11c662:      fxch     st(2)
0x11c664:      fld     st(0)
0x11c666:      add     ebx, 1
0x11c669:      fmul     st(2)
0x11c66b:      fxch     st(3)
0x11c66d:      fcomi    st(3)
0x11c66f:      jb      0x11c680
0x11cecb:      add     ebx, 1
0x11cece:      test    byte ptr [rsp + 0x48], 8
0x11ced3:      je      0x11d039
0x11ca4f:      mov     rax, qword ptr [rsp + 8]
0x11ca54:      mov     byte ptr [rax], 0
0x11ca57:      mov     rax, qword ptr [rsp + 0x88]
0x11ca5f:      sub     rax, qword ptr fs:[0x28]
0x11ca68:      jne     0x11d0eb
0x11ca33:      cmp     ebx, -1
0x11ca36:      je      0x11cbb8
0x11c847:      sub     r12, rbp
0x11c84a:      mov     rsi, rbp
0x11c84d:      mov     ecx, 0x29
0x11c852:      mov     r15, rax
0x11c855:      lea     rax, [rsp + 0x50]
0x11c85a:      mov     rdx, r12
0x11c85d:      mov     rdi, rax
0x11c860:      mov     qword ptr [rsp + 0x10], rax

```

```

0x11c865:    call    0x104b70
0x11cfac:    test    eax, eax
0x11cfae:    jne     0x11d117
0x11cf37:    lea     edx, [rax + 1]
0x11cf3a:    cmp     eax, 9
0x11cf3d:    je      0x11d024
0x11cf33:    test    esi, esi
0x11cf35:    je      0x11cfac
0x11ca1c:    lea     r8, [rbp + r12]
0x11ca21:    test    byte ptr [rsp + 0x48], 4
0x11ca26:    jne     0x11c830
0x11c991:    fadd    dword ptr [rip + 0x5b95]
0x11c997:    mov     r8d, dword ptr [rsp + 0x10]
0x11c99c:    test    r8d, r8d
0x11c99f:    jne     0x11c9c8
0x11c997:    mov     r8d, dword ptr [rsp + 0x10]
0x11c99c:    test    r8d, r8d
0x11c99f:    jne     0x11c9c8
0x11c983:    mov     qword ptr [rsp + 0x20], rax
0x11c988:    fild    qword ptr [rsp + 0x20]
0x11c98c:    test    rax, rax
0x11c98f:    jns     0x11c997
0x11cc9f:    mov     rdi, r12
0x11cca2:    call    0x1049b0
0x11cb10:    fld     st(0)
0x11cb12:    jmp     0x11cb20
0x11c6ac:    fld     dword ptr [rip + 0x5e7e]
0x11c6b2:    fxch    st(1)
0x11c6b4:    fcomi   st(1)
0x11c6b6:    jae     0x11ce00
0x11c684:    lea     r15, [rbp + 1]
0x11c688:    fdivrp  st(1)
0x11c68a:    cmp     dword ptr [rsp + 0x30], 1
0x11c68f:    adc     rbp, 2
0x11c693:    cmp     dword ptr [rsp + 0x10], 1
0x11c698:    je      0x11cc70
0x11d039:    mov     rax, rbp
0x11d03c:    mov     byte ptr [r8 - 1], 0x30
0x11d041:    not     rax
0x11d044:    add     r8, rax
0x11d047:    mov     eax, ebp
0x11d049:    cmp     ebp, 8
0x11d04c:    jae     0x11d080
0x11ced9:    mov     byte ptr [r8 - 1], 0x31
0x11cede:    lea     rbp, [r8 - 1]
0x11cee2:    jmp     0x11ca21
0x11ca6e:    add     rsp, 0x98
0x11ca75:    mov     rax, rbp

```

```

0x11ca78:    pop     rbx
0x11ca79:    pop     rbp
0x11ca7a:    pop     r12
0x11ca7c:    pop     r13
0x11ca7e:    pop     r14
0x11ca80:    pop     r15
0x11ca82:    ret
0x11d0eb:    call    0x1049d0
0x11cbb8:    mov     rcx, qword ptr [rsp + 0x18]
0x11cbbd:    cmp     rcx, 1
0x11cbc1:    jbe     0x11cfca
0x11ca3c:    mov     eax, dword ptr [rsp + 0x48]
0x11ca40:    and     eax, 0x100
0x11ca45:    mov     esi, eax
0x11ca47:    or      esi, ebx
0x11ca49:    jne     0x11cd08
0x11c86a:    mov     dword ptr [rsp + 0x20], ebx
0x11c86e:    mov     rbp, qword ptr [rsp + 0x40]
0x11c873:    mov     rbx, r12
0x11c876:    mov     r12, qword ptr [rsp + 0x38]
0x11c87b:    jmp     0x11c8bb
0x11cfb4:    mov     r8, qword ptr [rsp + 8]
0x11cfb9:    test    byte ptr [rsp + 0x48], 8
0x11cfbe:    jne     0x11cf7c
0x11d117:    add     eax, 0x30
0x11d11a:    jmp     0x11cf46
0x11d024:    lea     rcx, [r9 + 1]
0x11d028:    cmp     r9, 9
0x11d02c:    je      0x11d0cc
0x11cf43:    lea     eax, [rdx + 0x30]
0x11cf46:    mov     rdi, qword ptr [rsp]
0x11cf4a:    lea     r8, [rdi + 0x287]
0x11cf51:    mov     byte ptr [rdi + 0x287], al
0x11cf57:    mov     eax, ebp
0x11cf59:    sub     r8, rbp
0x11cf5c:    cmp     ebp, 8
0x11cf5f:    jae     0x11cfd1
0x11c9a1:    fxch    st(1)
0x11c9a3:    fucomip st(1)
0x11c9a5:    jp      0x11c9ad
0x11c9c8:    fstp    st(1)
0x11c9ca:    sub     rsp, 0x10
0x11c9ce:    mov     rdx, -1
0x11c9d5:    mov     esi, 1
0x11c9da:    xor     eax, eax
0x11c9dc:    mov     rbx, qword ptr [rsp + 0x10]
0x11c9e1:    lea     rcx, [rip + 0x5adc]
0x11c9e8:    fstp    xword ptr [rsp]

```

```

0x11c9eb:    mov     rdi, rbx
0x11c9ee:    call   0x104f10
0x11cca7:    pop     rsi
0x11cca8:    pop     rdi
0x11cca9:    fld     xword ptr [rsp + 0x20]
0x11ccad:    cmp     rax, rbp
0x11ccb0:    mov     rdx, rax
0x11ccb3:    jbe     0x11cb67
0x11cb20:    fstp    xword ptr [rsp + 0x20]
0x11cb24:    sub     rsp, 0x10
0x11cb28:    mov     esi, 1
0x11cb2d:    xor     eax, eax
0x11cb2f:    mov     r12, qword ptr [rsp + 0x10]
0x11cb34:    lea     rcx, [rip + 0x598f]
0x11cb3b:    mov     rdx, -1
0x11cb42:    mov     rdi, r12
0x11cb45:    fstp    xword ptr [rsp]
0x11cb48:    call   0x104f10
0x11c6bc:    fstp    st(1)
0x11c6be:    fnstcw  word ptr [rsp + 0x4e]
0x11c6c2:    movzx   eax, word ptr [rsp + 0x4e]
0x11c6c7:    or      ah, 0xc
0x11c6ca:    mov     word ptr [rsp + 0x4c], ax
0x11c6cf:    fld     st(0)
0x11c6d1:    fldcw   word ptr [rsp + 0x4c]
0x11c6d5:    fistp   qword ptr [rsp + 0x20]
0x11c6d9:    fldcw   word ptr [rsp + 0x4e]
0x11c6dd:    mov     rax, qword ptr [rsp + 0x20]
0x11c6e2:    mov     qword ptr [rsp + 0x20], rax
0x11c6e7:    fild    qword ptr [rsp + 0x20]
0x11c6eb:    test    rax, rax
0x11c6ee:    jns     0x11c6f6
0x11ce00:    fnstcw  word ptr [rsp + 0x4e]
0x11ce04:    fsubr   st(1), st(0)
0x11ce06:    fxch    st(1)
0x11ce08:    movzx   eax, word ptr [rsp + 0x4e]
0x11ce0d:    or      ah, 0xc
0x11ce10:    mov     word ptr [rsp + 0x4c], ax
0x11ce15:    fldcw   word ptr [rsp + 0x4c]
0x11ce19:    fistp   qword ptr [rsp + 0x20]
0x11ce1d:    fldcw   word ptr [rsp + 0x4e]
0x11ce21:    mov     rax, qword ptr [rsp + 0x20]
0x11ce26:    btc     rax, 0x3f
0x11ce2b:    jmp     0x11c6e2
0x11d04e:    and     ebp, 4
0x11d051:    jne     0x11d0f0
0x11d080:    mov     rax, qword ptr [r15]
0x11d083:    lea     rcx, [r8 + 8]

```

```

0x11d087:    and     rcx, 0xfffffffffffffff8
0x11d08b:    mov     qword ptr [r8], rax
0x11d08e:    mov     eax, ebp
0x11d090:    mov     rdx, qword ptr [r15 + rax - 8]
0x11d095:    mov     qword ptr [r8 + rax - 8], rdx
0x11d09a:    mov     rax, r8
0x11d09d:    sub     rax, rcx
0x11d0a0:    sub     r15, rax
0x11d0a3:    add     eax, ebp
0x11d0a5:    and     eax, 0xffffffff8
0x11d0a8:    cmp     eax, 8
0x11d0ab:    jb      0x11ced9
0x11ca21:    test    byte ptr [rsp + 0x48], 4
0x11ca26:    jne     0x11c830
0x11cfca:    xor     ebx, ebx
0x11cfcc:    jmp     0x11ca3c
0x11cbc7:    mov     edx, dword ptr [rsp + 0x34]
0x11cbcb:    mov     ebx, 1
0x11cbd0:    mov     eax, 1
0x11cbd5:    nop     dword ptr [rax]
0x11cbd8:    imul    rax, rdx
0x11cbdc:    cmp     rcx, rax
0x11cbdf:    jbe     0x11cbe9
0x11cd08:    test    byte ptr [rsp + 0x48], 0x40
0x11cd0d:    je      0x11cc10
0x11c8bb:    movzx   eax, byte ptr [r12]
0x11c8c0:    test    al, al
0x11c8c2:    je      0x11c880
0x11cfc0:    mov     eax, 0x30
0x11cfc5:    jmp     0x11cf46
0x11cf7c:    xor     edi, edi
0x11cf7e:    cmp     dword ptr [rsp + 0x10], 1
0x11cf83:    jne     0x11c7bf
0x11cf46:    mov     rdi, qword ptr [rsp]
0x11cf4a:    lea     r8, [rdi + 0x287]
0x11cf51:    mov     byte ptr [rdi + 0x287], al
0x11cf57:    mov     eax, ebp
0x11cf59:    sub     r8, rbp
0x11cf5c:    cmp     ebp, 8
0x11cf5f:    jae     0x11cfd1
0x11d0cc:    mov     r8, qword ptr [rsp + 8]
0x11d0d1:    jmp     0x11cf7a
0x11d032:    xor     esi, esi
0x11d034:    jmp     0x11cfb4
0x11cf61:    test    bpl, 4
0x11cf65:    jne     0x11d0d6
0x11cfd1:    mov     rax, qword ptr [r15]
0x11cfd4:    lea     rsi, [r8 + 8]

```

```

0x11cfd8:    mov     rdi, r15
0x11cfdb:    and     rsi, 0xfffffffffffffff8
0x11cfd8:    mov     qword ptr [r8], rax
0x11cfe2:    mov     eax, ebp
0x11cfe4:    mov     rdx, qword ptr [r15 + rax - 8]
0x11cfe9:    mov     qword ptr [r8 + rax - 8], rdx
0x11cfee:    mov     rax, r8
0x11cff1:    sub     rax, rsi
0x11cff4:    sub     rdi, rax
0x11cff7:    add     eax, ebp
0x11cff9:    and     eax, 0xffffffff
0x11cffc:    cmp     eax, 8
0x11cfff:    jb      0x11cf7a
0x11c9ad:    fstp    st(0)
0x11c9af:    add     rax, 1
0x11c9b3:    mov     qword ptr [rsp + 0x10], rax
0x11c9b8:    fild    qword ptr [rsp + 0x10]
0x11c9bc:    js      0x11cee7
0x11c9a7:    je      0x11c9ca
0x11cb67:    test    byte ptr [rsp + 0x48], 8
0x11cb6c:    je      0x11cb88
0x11ccb9:    fld     dword ptr [rip + 0x5875]
0x11ccbf:    fmul    st(1), st(0)
0x11ccc1:    fdivp   st(1)
0x11ccc3:    sub     rsp, 0x10
0x11ccc7:    mov     rdx, -1
0x11ccce:    xor     eax, eax
0x11ccd0:    mov     r15, qword ptr [rsp + 0x10]
0x11ccd5:    lea     rcx, [rip + 0x57e8]
0x11ccdc:    mov     esi, 1
0x11cce1:    mov     rdi, r15
0x11cce4:    fstp    xword ptr [rsp]
0x11cce7:    call    0x104f10
0x11cb4d:    mov     rdi, r12
0x11cb50:    call    0x1049b0
0x11c6f0:    fadd    dword ptr [rip + 0x5e36]
0x11c6f6:    mov     ecx, dword ptr [rsp + 0x10]
0x11c6fa:    test    ecx, ecx
0x11c6fc:    jne     0x11cb18
0x11c6f6:    mov     ecx, dword ptr [rsp + 0x10]
0x11c6fa:    test    ecx, ecx
0x11c6fc:    jne     0x11cb18
0x11c6e2:    mov     qword ptr [rsp + 0x20], rax
0x11c6e7:    fild    qword ptr [rsp + 0x20]
0x11c6eb:    test    rax, rax
0x11c6ee:    jns     0x11c6f6
0x11d057:    test    eax, eax
0x11d059:    je      0x11ced9

```



```

0x11d0f0:    mov     edx, dword ptr [r15]
0x11d0f3:    mov     dword ptr [r8], edx
0x11d0f6:    mov     edx, dword ptr [r15 + rax - 4]
0x11d0fb:    mov     dword ptr [r8 + rax - 4], edx
0x11d100:    jmp     0x11ced9
0x11d0b1:    and     eax, 0xffffffff8
0x11d0b4:    xor     edx, edx
0x11d0b6:    mov     esi, edx
0x11d0b8:    add     edx, 8
0x11d0bb:    mov     rdi, qword ptr [r15 + rsi]
0x11d0bf:    mov     qword ptr [rcx + rsi], rdi
0x11d0c3:    cmp     edx, eax
0x11d0c5:    jb     0x11d0b6
0x11cbe9:    mov     esi, dword ptr [rsp + 0x48]
0x11cbec:    mov     eax, esi
0x11cbef:    and     eax, 0x100
0x11cbf4:    and     esi, 0x40
0x11cbf7:    je     0x11cc18
0x11cbe1:    add     ebx, 1
0x11cbe4:    cmp     ebx, 8
0x11cbe7:    jne     0x11cbd8
0x11cc10:    test    ebx, ebx
0x11cc12:    je     0x11d105
0x11cd13:    jmp     0x11cbf9
0x11c880:    cmp     r13, rbx
0x11c883:    mov     rax, qword ptr [rsp + 0x10]
0x11c888:    cmova   r13, rbx
0x11c88c:    sub     rbx, r13
0x11c88f:    lea     rsi, [rax + rbx]
0x11c893:    sub     rbp, r13
0x11c896:    mov     rdx, r13
0x11c899:    mov     rdi, rbp
0x11c89c:    call    0x104bb0
0x11c8c4:    cmp     al, 0x7e
0x11c8c6:    ja     0x11ca88
0x11cf89:    jmp     0x11c7e8
0x11cf7a:    xor     esi, esi
0x11cf7c:    xor     edi, edi
0x11cf7e:    cmp     dword ptr [rsp + 0x10], 1
0x11cf83:    jne     0x11c7bf
0x11cf6b:    test    eax, eax
0x11cf6d:    je     0x11cf7a
0x11d0d6:    mov     edx, dword ptr [r15]
0x11d0d9:    mov     dword ptr [r8], edx
0x11d0dc:    mov     edx, dword ptr [r15 + rax - 4]
0x11d0e1:    mov     dword ptr [r8 + rax - 4], edx
0x11d0e6:    jmp     0x11cf7a
0x11d005:    and     eax, 0xffffffff8

```

```

0x11d008:    mov     r9d, eax
0x11d00b:    xor     eax, eax
0x11d00d:    mov     edx, eax
0x11d00f:    add     eax, 8
0x11d012:    mov     r11, qword ptr [rdi + rdx]
0x11d016:    mov     qword ptr [rsi + rdx], r11
0x11d01a:    cmp     eax, r9d
0x11d01d:    jb      0x11d00d
0x11cee7:    fadd    dword ptr [rip + 0x563f]
0x11ceed:    jmp     0x11c9ca
0x11c9c2:    jmp     0x11c9ca
0x11c9a9:    fstp    st(0)
0x11c9ab:    jmp     0x11c9af
0x11cb88:    fstp    st(0)
0x11cb8a:    mov     r12, rdx
0x11cb8d:    sub     r12, r15
0x11cb90:    jmp     0x11ca08
0x11cb6e:    mov     rax, qword ptr [rsp]
0x11cb72:    cmp     byte ptr [rax + rdx - 1], 0x30
0x11cb77:    je      0x11ce40
0x11ccec:    mov     rdi, r15
0x11ccef:    call    0x1049b0
0x11cb55:    mov     rdx, rax
0x11cb58:    pop     rax
0x11cb59:    pop     rcx
0x11cb5a:    fld     xword ptr [rsp + 0x20]
0x11cb5e:    cmp     rdx, rbp
0x11cb61:    ja      0x11cd20
0x11c702:    fxch    st(1)
0x11c704:    fucomi  st(1)
0x11c706:    jp      0x11c712
0x11cb18:    fxch    st(1)
0x11cb1a:    jmp     0x11cb20
0x11d05f:    movzx   edx, byte ptr [r15]
0x11d063:    mov     byte ptr [r8], dl
0x11d066:    test    al, 2
0x11d068:    je      0x11ced9
0x11d0b6:    mov     esi, edx
0x11d0b8:    add     edx, 8
0x11d0bb:    mov     rdi, qword ptr [r15 + rsi]
0x11d0bf:    mov     qword ptr [rcx + rsi], rdi
0x11d0c3:    cmp     edx, eax
0x11d0c5:    jb      0x11d0b6
0x11d0c7:    jmp     0x11ced9
0x11cc18:    mov     rsi, qword ptr [rsp + 8]
0x11cc1d:    mov     r9d, dword ptr [rsp + 0x30]
0x11cc22:    lea     rdx, [rsi + 1]
0x11cc26:    test    r9d, r9d

```

```

0x11cc29:    jne     0x11cc34
0x11cbf9:    mov     rsi, qword ptr [rsp]
0x11cbfd:    lea     rdi, [rsi + 0x289]
0x11cc04:    mov     byte ptr [rsi + 0x288], 0x20
0x11cc0b:    mov     qword ptr [rsp + 8], rdi
0x11cc10:    test    ebx, ebx
0x11cc12:    je      0x11d105
0x11cbd8:    imul    rax, rdx
0x11cbdc:    cmp     rcx, rax
0x11cbdf:    jbe     0x11cbe9
0x11d105:    mov     rdx, qword ptr [rsp + 8]
0x11d10a:    test    eax, eax
0x11d10c:    jne     0x11cc5a
0x11c8a1:    test    rbx, rbx
0x11c8a4:    je      0x11caa0
0x11c8cc:    cmp     rax, rbx
0x11c8cf:    cmova   rax, rbx
0x11c8d3:    sub     rbx, rax
0x11c8d6:    mov     r13, rax
0x11c8d9:    mov     rax, qword ptr [rsp + 0x10]
0x11c8de:    lea     rsi, [rax + rbx]
0x11c8e2:    add     r12, 1
0x11c8e6:    jmp     0x11c893
0x11ca88:    mov     r13, rbx
0x11ca8b:    mov     rsi, qword ptr [rsp + 0x10]
0x11ca90:    xor     ebx, ebx
0x11ca92:    jmp     0x11c8e2
0x11cf6f:    movzx   edx, byte ptr [r15]
0x11cf73:    mov     byte ptr [r8], dl
0x11cf76:    test    al, 2
0x11cf78:    jne     0x11cf8e
0x11d00d:    mov     edx, eax
0x11d00f:    add     eax, 8
0x11d012:    mov     r11, qword ptr [rdi + rdx]
0x11d016:    mov     qword ptr [rsi + rdx], r11
0x11d01a:    cmp     eax, r9d
0x11d01d:    jb      0x11d00d
0x11d01f:    jmp     0x11cf7a
0x11c9af:    add     rax, 1
0x11c9b3:    mov     qword ptr [rsp + 0x10], rax
0x11c9b8:    fild    qword ptr [rsp + 0x10]
0x11c9bc:    js      0x11cee7
0x11ca08:    mov     rbp, qword ptr [rsp + 8]
0x11ca0d:    mov     rsi, qword ptr [rsp]
0x11ca11:    sub     rbp, rdx
0x11ca14:    mov     rdi, rbp
0x11ca17:    call    0x104d40
0x11ce40:    fld     dword ptr [rip + 0x56ee]

```

```

0x11ce46:    cmp     dword ptr [rsp + 0x10], 1
0x11ce4b:    fmul    st(1), st(0)
0x11ce4d:    jne     0x11cd28
0x11cb7d:    fstp    st(0)
0x11cb7f:    jmp     0x11cb8a
0x11ccf4:    pop     r13
0x11ccf6:    pop     r15
0x11ccf8:    mov     rdx, rax
0x11ccfb:    mov     r12, rax
0x11ccfe:    jmp     0x11ca08
0x11cd20:    fld     dword ptr [rip + 0x580e]
0x11cd26:    fmul    st(1), st(0)
0x11cd28:    fld     xword ptr [rip + 0x5812]
0x11cd2e:    fcomip  st(2)
0x11cd30:    jbe     0x11ccc1
0x11c712:    fstp    st(1)
0x11c714:    nop     dword ptr [rax]
0x11c718:    add     rax, 1
0x11c71c:    mov     qword ptr [rsp + 0x20], rax
0x11c721:    fild    qword ptr [rsp + 0x20]
0x11c725:    jns     0x11cb1c
0x11c708:    je      0x11cb20
0x11d06e:    movzx   edx, word ptr [r15 + rax - 2]
0x11d074:    mov     word ptr [r8 + rax - 2], dx
0x11d07a:    jmp     0x11ced9
0x11cc2b:    cmp     ebx, 1
0x11cc2e:    je      0x11cf00
0x11cc34:    movsxd  rbx, ebx
0x11cc37:    lea     rcx, [rip + 0x58e2]
0x11cc3e:    movzx   ecx, byte ptr [rcx + rbx]
0x11cc42:    mov     byte ptr [rsi], cl
0x11cc44:    test    eax, eax
0x11cc46:    je      0x11cf0b
0x11d112:    jmp     0x11ca4f
0x11cc5a:    lea     rax, [rdx + 1]
0x11cc5e:    mov     byte ptr [rdx], 0x42
0x11cc61:    mov     qword ptr [rsp + 8], rax
0x11cc66:    jmp     0x11ca4f
0x11caa0:    mov     ebx, dword ptr [rsp + 0x20]
0x11caa4:    jmp     0x11ca2c
0x11c8aa:    sub     rbp, r15
0x11c8ad:    mov     rdx, r15
0x11c8b0:    mov     rsi, r14
0x11c8b3:    mov     rdi, rbp
0x11c8b6:    call    0x104bb0
0x11c893:    sub     rbp, r13
0x11c896:    mov     rdx, r13
0x11c899:    mov     rdi, rbp

```

```

0x11c89c:    call    0x104bb0
0x11c8e2:    add     r12, 1
0x11c8e6:    jmp     0x11c893
0x11cf8e:    movzx   edx, word ptr [r15 + rax - 2]
0x11cf94:    mov     word ptr [r8 + rax - 2], dx
0x11cf9a:    jmp     0x11cf7a
0x11ce53:    jmp     0x11ccc1
0x11cd28:    fld     xword ptr [rip + 0x5812]
0x11cd2e:    fcomip  st(2)
0x11cd30:    jbe     0x11ccc1
0x11cb8a:    mov     r12, rdx
0x11cb8d:    sub     r12, r15
0x11cb90:    jmp     0x11ca08
0x11ccc1:    fdivp   st(1)
0x11ccc3:    sub     rsp, 0x10
0x11ccc7:    mov     rdx, -1
0x11ccce:    xor     eax, eax
0x11ccd0:    mov     r15, qword ptr [rsp + 0x10]
0x11ccd5:    lea     rcx, [rip + 0x57e8]
0x11ccdc:    mov     esi, 1
0x11cce1:    mov     rdi, r15
0x11cce4:    fstp    xword ptr [rsp]
0x11cce7:    call    0x104f10
0x11cd32:    fld     dword ptr [rip + 0x57f8]
0x11cd38:    fxch    st(2)
0x11cd3a:    fcomi   st(2)
0x11cd3c:    jae     0x11ce90
0x11c72b:    fadd    dword ptr [rip + 0x5dfb]
0x11c731:    fxch    st(1)
0x11c733:    jmp     0x11cb20
0x11cb1c:    fxch    st(1)
0x11cb1e:    nop
0x11cb20:    fstp    xword ptr [rsp + 0x20]
0x11cb24:    sub     rsp, 0x10
0x11cb28:    mov     esi, 1
0x11cb2d:    xor     eax, eax
0x11cb2f:    mov     r12, qword ptr [rsp + 0x10]
0x11cb34:    lea     rcx, [rip + 0x598f]
0x11cb3b:    mov     rdx, -1
0x11cb42:    mov     rdi, r12
0x11cb45:    fstp    xword ptr [rsp]
0x11cb48:    call    0x104f10
0x11c70e:    fstp    st(1)
0x11c710:    jmp     0x11c718
0x11cf00:    mov     byte ptr [rsi], 0x6b
0x11cf03:    test    eax, eax
0x11cf05:    jne     0x11cc5a
0x11cf0b:    mov     qword ptr [rsp + 8], rdx

```

```

0x11cf10:      jmp      0x11ca4f
0x11cc4c:      mov      r8d, dword ptr [rsp + 0x30]
0x11cc51:      test     r8d, r8d
0x11cc54:      jne       0x11ce30
0x11cd42:      fstp      st(2)
0x11cd44:      fxch      st(1)
0x11cd46:      fnstcw    word ptr [rsp + 0x4e]
0x11cd4a:      movzx     eax, word ptr [rsp + 0x4e]
0x11cd4f:      or        ah, 0xc
0x11cd52:      mov       word ptr [rsp + 0x4c], ax
0x11cd57:      fld       st(0)
0x11cd59:      fldcw     word ptr [rsp + 0x4c]
0x11cd5d:      fistp     qword ptr [rsp + 0x20]
0x11cd61:      fldcw     word ptr [rsp + 0x4e]
0x11cd65:      mov       rax, qword ptr [rsp + 0x20]
0x11cd6a:      mov       qword ptr [rsp + 0x20], rax
0x11cd6f:      fild      qword ptr [rsp + 0x20]
0x11cd73:      test      rax, rax
0x11cd76:      jns       0x11cd7e
0x11ce90:      fnstcw    word ptr [rsp + 0x4e]
0x11ce94:      fsubr     st(2), st(0)
0x11ce96:      fxch      st(2)
0x11ce98:      movzx     eax, word ptr [rsp + 0x4e]
0x11ce9d:      or        ah, 0xc
0x11cea0:      mov       word ptr [rsp + 0x4c], ax
0x11cea5:      fldcw     word ptr [rsp + 0x4c]
0x11cea9:      fistp     qword ptr [rsp + 0x20]
0x11cead:      fldcw     word ptr [rsp + 0x4e]
0x11ceb1:      fxch      st(1)
0x11ceb3:      mov       rax, qword ptr [rsp + 0x20]
0x11ceb8:      btc       rax, 0x3f
0x11cebd:      jmp       0x11cd6a
0x11c718:      add       rax, 1
0x11c71c:      mov       qword ptr [rsp + 0x20], rax
0x11c721:      fild      qword ptr [rsp + 0x20]
0x11c725:      jns       0x11cb1c
0x11ce30:      mov       byte ptr [rsi + 1], 0x69
0x11ce34:      lea       rdx, [rsi + 2]
0x11ce38:      jmp       0x11cc5a
0x11cd78:      fadd      dword ptr [rip + 0x57ae]
0x11cd7e:      mov       edx, dword ptr [rsp + 0x10]
0x11cd82:      test      edx, edx
0x11cd84:      jne       0x11cdb0
0x11cd7e:      mov       edx, dword ptr [rsp + 0x10]
0x11cd82:      test      edx, edx
0x11cd84:      jne       0x11cdb0
0x11cd6a:      mov       qword ptr [rsp + 0x20], rax
0x11cd6f:      fild      qword ptr [rsp + 0x20]

```

```

0x11cd73:      test    rax, rax
0x11cd76:      jns     0x11cd7e
0x11cd86:      fucomi  st(1)
0x11cd88:      fstp    st(1)
0x11cd8a:      jp      0x11cd92
0x11cdb0:      fstp    st(1)
0x11cdb2:      fxch    st(1)
0x11cdb4:      jmp     0x11ccc1
0x11cd92:      fstp    st(0)
0x11cd94:      add     rax, 1
0x11cd98:      mov     qword ptr [rsp + 0x10], rax
0x11cd9d:      fild    qword ptr [rsp + 0x10]
0x11cda1:      js      0x11cef2
0x11cd8c:      je      0x11cdb2
0x11cef2:      fadd    dword ptr [rip + 0x5634]
0x11cef8:      fxch    st(1)
0x11cefa:      jmp     0x11ccc1
0x11cda7:      jmp     0x11cdb2
0x11cdb2:      fxch    st(1)
0x11cdb4:      jmp     0x11ccc1
0x11cd8e:      fstp    st(0)
0x11cd90:      jmp     0x11cd94
0x11cd94:      add     rax, 1
0x11cd98:      mov     qword ptr [rsp + 0x10], rax
0x11cd9d:      fild    qword ptr [rsp + 0x10]
0x11cda1:      js      0x11cef2
0x11c658:      nop     dword ptr [rax + rax]
0x11c67c:      nop     dword ptr [rax]
0x11c738:      nop     dword ptr [rax + rax]
0x11c87d:      nop     dword ptr [rax]
0x11c8e8:      nop     dword ptr [rax + rax]
0x11c905:      nop     dword ptr [rax]
0x11c924:      nop     dword ptr [rax]
0x11c933:      nop     dword ptr [rax + rax]
0x11c9c4:      nop     dword ptr [rax]
0x11ca83:      nop     dword ptr [rax + rax]
0x11ca97:      nop     word ptr [rax + rax]
0x11caa6:      nop     word ptr cs:[rax + rax]
0x11cb08:      nop     dword ptr [rax + rax]
0x11cb14:      nop     dword ptr [rax]
0x11cb81:      nop     dword ptr [rax]
0x11cb95:      nop     dword ptr [rax]
0x11cbb5:      nop     dword ptr [rax]
0x11cc6b:      nop     dword ptr [rax + rax]
0x11cd03:      nop     dword ptr [rax + rax]
0x11cd18:      nop     dword ptr [rax + rax]
0x11cda9:      nop     dword ptr [rax]
0x11cdb9:      nop     dword ptr [rax]

```

```

0x11cde2:      nop      word ptr [rax + rax]
0x11cdfa:      nop      word ptr [rax + rax]
0x11ce3d:      nop      dword ptr [rax]
0x11ce58:      nop      dword ptr [rax + rax]
0x11ceff:      nop
0x11cf15:      nop      dword ptr [rax]
0x11cf9c:      nop      dword ptr [rax]
0x11d07f:      nop
0x11d11f:      nop
0x11d120:      endbr64
0x11d124:      push     r14
0x11d126:      push     r13
0x11d128:      mov      r13, rsi
0x11d12b:      push     r12
0x11d12d:      push     rbp
0x11d12e:      mov      rbp, rdx
0x11d131:      push     rbx
0x11d132:      mov      rbx, rdi
0x11d135:      sub      rsp, 0x10
0x11d139:      mov      rax, qword ptr fs:[0x28]
0x11d142:      mov      qword ptr [rsp + 8], rax
0x11d147:      xor      eax, eax
0x11d149:      test     rdi, rdi
0x11d14c:      je       0x11d270
0x11d270:      lea      rdi, [rip + 0x5259]
0x11d277:      call     0x1047e0
0x11d152:      xor      r12d, r12d
0x11d155:      cmp      byte ptr [rbx], 0x27
0x11d158:      jne      0x11d164
0x11d27c:      mov      rbx, rax
0x11d27f:      test     rax, rax
0x11d282:      jne      0x11d152
0x11d15a:      add      rbx, 1
0x11d15e:      mov      r12d, 4
0x11d164:      lea      r14, [rip + 0x53ad]
0x11d16b:      mov      ecx, 4
0x11d170:      lea      rsi, [rip + 0xb819]
0x11d177:      mov      rdi, rbx
0x11d17a:      mov      rdx, r14
0x11d17d:      call     0x11dc80
0x11d164:      lea      r14, [rip + 0x53ad]
0x11d16b:      mov      ecx, 4
0x11d170:      lea      rsi, [rip + 0xb819]
0x11d177:      mov      rdi, rbx
0x11d17a:      mov      rdx, r14
0x11d17d:      call     0x11dc80
0x11d288:      lea      rdi, [rip + 0x524c]
0x11d28f:      call     0x1047e0

```



```

0x11d182:    test    eax, eax
0x11d184:    js      0x11d1c0
0x11d294:    mov     rbx, rax
0x11d297:    test    rax, rax
0x11d29a:    jne     0x11d152
0x11d1c0:    xor     edx, edx
0x11d1c2:    mov     rsi, rsp
0x11d1c5:    lea     r8, [rip + 0x5329]
0x11d1cc:    mov     rcx, rbp
0x11d1cf:    mov     rdi, rbx
0x11d1d2:    call    0x11d7e0
0x11d186:    cdqe
0x11d188:    mov     qword ptr [rbp], 1
0x11d190:    or      r12d, dword ptr [r14 + rax*4]
0x11d194:    xor     eax, eax
0x11d196:    mov     dword ptr [r13], r12d
0x11d19a:    mov     rdx, qword ptr [rsp + 8]
0x11d19f:    sub     rdx, qword ptr fs:[0x28]
0x11d1a8:    jne     0x11d2fd
0x11d2a0:    lea     rdi, [rip + 0x523e]
0x11d2a7:    call    0x1047e0
0x11d1d7:    test    eax, eax
0x11d1d9:    jne     0x11d228
0x11d1ae:    add     rsp, 0x10
0x11d1b2:    pop     rbx
0x11d1b3:    pop     rbp
0x11d1b4:    pop     r12
0x11d1b6:    pop     r13
0x11d1b8:    pop     r14
0x11d1ba:    ret
0x11d2fd:    call    0x1049d0
0x11d2ac:    test    rax, rax
0x11d2af:    je      0x11d2e6
0x11d1db:    movzx   ecx, byte ptr [rbx]
0x11d1de:    lea     edx, [rcx - 0x30]
0x11d1e1:    mov     rcx, qword ptr [rsp]
0x11d1e5:    cmp     dl, 9
0x11d1e8:    ja      0x11d200
0x11d228:    mov     dword ptr [r13], 0
0x11d230:    mov     rdx, qword ptr [rbp]
0x11d234:    test    rdx, rdx
0x11d237:    jne     0x11d19a
0x11d2e6:    mov     qword ptr [rbp], 0x400
0x11d2ee:    xor     eax, eax
0x11d2f0:    mov     dword ptr [r13], 0
0x11d2f8:    jmp     0x11d19a
0x11d2b1:    mov     qword ptr [rbp], 0x200
0x11d2b9:    xor     eax, eax

```

```

0x11d2bb:    mov     dword ptr [r13], 0
0x11d2c3:    jmp     0x11d19a
0x11d1ea:    jmp     0x11d217
0x11d200:    cmp     rcx, rbx
0x11d203:    jne     0x11d1f0
0x11d23d:    lea     rdi, [rip + 0x52a1]
0x11d244:    call    0x1047e0
0x11d19a:    mov     rdx, qword ptr [rsp + 8]
0x11d19f:    sub     rdx, qword ptr fs:[0x28]
0x11d1a8:    jne     0x11d2fd
0x11d217:    mov     rdx, qword ptr [rbp]
0x11d21b:    mov     dword ptr [r13], r12d
0x11d21f:    jmp     0x11d234
0x11d205:    cmp     byte ptr [rcx - 1], 0x42
0x11d209:    je      0x11d2d0
0x11d1f0:    movzx   edi, byte ptr [rbx + 1]
0x11d1f4:    add     rbx, 1
0x11d1f8:    lea     edx, [rdi - 0x30]
0x11d1fb:    cmp     dl, 9
0x11d1fe:    jbe     0x11d217
0x11d249:    cmp     rax, 1
0x11d24d:    sbb     rax, rax
0x11d250:    and     eax, 0x200
0x11d255:    add     rax, 0x200
0x11d25b:    mov     qword ptr [rbp], rax
0x11d25f:    mov     eax, 4
0x11d264:    jmp     0x11d19a
0x11d234:    test    rdx, rdx
0x11d237:    jne     0x11d19a
0x11d2d0:    or      r12d, 0x180
0x11d2d7:    cmp     byte ptr [rcx - 2], 0x69
0x11d2db:    jne     0x11d217
0x11d20f:    or      r12b, 0x80
0x11d213:    or      r12d, 0x20
0x11d217:    mov     rdx, qword ptr [rbp]
0x11d21b:    mov     dword ptr [r13], r12d
0x11d21f:    jmp     0x11d234
0x11d2e1:    jmp     0x11d213
0x11d213:    or      r12d, 0x20
0x11d217:    mov     rdx, qword ptr [rbp]
0x11d21b:    mov     dword ptr [r13], r12d
0x11d21f:    jmp     0x11d234
0x11d1bb:    nop     dword ptr [rax + rax]
0x11d1ec:    nop     dword ptr [rax]
0x11d221:    nop     dword ptr [rax]
0x11d269:    nop     dword ptr [rax]
0x11d2c8:    nop     dword ptr [rax + rax]
0x11d302:    nop     word ptr cs:[rax + rax]

```

```

0x11d30c:      nop      dword ptr [rax]
0x11d310:      endbr64
0x11d314:      push     r12
0x11d316:      push     rbp
0x11d317:      mov      ebp, edi
0x11d319:      push     rbx
0x11d31a:      mov      rbx, qword ptr [rip + 0xe81f]
0x11d321:      test     rbx, rbx
0x11d324:      jne      0x11d339
0x11d326:      jmp      0x11d350
0x11d339:      cmp      dword ptr [rbx], ebp
0x11d33b:      jne      0x11d330
0x11d350:      mov      edi, ebp
0x11d352:      lea      r12, [rip + 0x508c]
0x11d359:      call     0x104970
0x11d33d:      xor      eax, eax
0x11d33f:      cmp      byte ptr [rbx + 0x10], 0
0x11d343:      je       0x11d349
0x11d330:      mov      rbx, qword ptr [rbx + 8]
0x11d334:      test     rbx, rbx
0x11d337:      je       0x11d350
0x11d35e:      mov      edi, 0x11
0x11d363:      test     rax, rax
0x11d366:      je       0x11d377
0x11d349:      pop      rbx
0x11d34a:      pop      rbp
0x11d34b:      pop      r12
0x11d34d:      ret
0x11d345:      lea      rax, [rbx + 0x10]
0x11d349:      pop      rbx
0x11d34a:      pop      rbp
0x11d34b:      pop      r12
0x11d34d:      ret
0x11d377:      call     0x11d670
0x11d368:      mov      r12, qword ptr [rax]
0x11d36b:      mov      rdi, r12
0x11d36e:      call     0x1049b0
0x11d37c:      mov      rsi, r12
0x11d37f:      mov      dword ptr [rax], ebp
0x11d381:      lea      rdi, [rax + 0x10]
0x11d385:      mov      rbx, rax
0x11d388:      call     0x1048b0
0x11d373:      lea      rdi, [rax + 0x11]
0x11d377:      call     0x11d670
0x11d38d:      mov      rax, qword ptr [rip + 0xe7ac]
0x11d394:      mov      qword ptr [rip + 0xe7a5], rbx
0x11d39b:      mov      qword ptr [rbx + 8], rax
0x11d39f:      jmp      0x11d33d

```

```

0x11d328:    nop    dword ptr [rax + rax]
0x11d34e:    nop
0x11d3a1:    nop    word ptr cs:[rax + rax]
0x11d3ac:    nop    dword ptr [rax]
0x11d3b0:    endbr64
0x11d3b4:    push   r12
0x11d3b6:    push   rbp
0x11d3b7:    mov    rbp, rdi
0x11d3ba:    push   rbx
0x11d3bb:    mov    rbx, qword ptr [rip + 0xe77e]
0x11d3c2:    test   rbx, rbx
0x11d3c5:    je     0x11d400
0x11d400:    mov    rbx, qword ptr [rip + 0xe731]
0x11d407:    test   rbx, rbx
0x11d40a:    je     0x11d448
0x11d3c7:    movzx  r12d, byte ptr [rdi]
0x11d3cb:    jmp     0x11d3d9
0x11d448:    mov    rdi, rbp
0x11d44b:    call   0x104b60
0x11d40c:    movzx  r12d, byte ptr [rbp]
0x11d411:    jmp     0x11d421
0x11d3d9:    cmp    byte ptr [rbx + 0x10], r12b
0x11d3dd:    jne     0x11d3d0
0x11d450:    mov    rdi, rbp
0x11d453:    mov    r12, rax
0x11d456:    call   0x1049b0
0x11d421:    cmp    byte ptr [rbx + 0x10], r12b
0x11d425:    jne     0x11d418
0x11d3df:    lea    rdi, [rbx + 0x10]
0x11d3e3:    mov    rsi, rbp
0x11d3e6:    call   0x104b30
0x11d3d0:    mov    rbx, qword ptr [rbx + 8]
0x11d3d4:    test   rbx, rbx
0x11d3d7:    je     0x11d400
0x11d45b:    lea    rdi, [rax + 0x11]
0x11d45f:    call   0x11d670
0x11d427:    lea    rdi, [rbx + 0x10]
0x11d42b:    mov    rsi, rbp
0x11d42e:    call   0x104b30
0x11d418:    mov    rbx, qword ptr [rbx + 8]
0x11d41c:    test   rbx, rbx
0x11d41f:    je     0x11d448
0x11d3eb:    test   eax, eax
0x11d3ed:    jne     0x11d3d0
0x11d464:    mov    rsi, rbp
0x11d467:    lea    rdi, [rax + 0x10]
0x11d46b:    mov    rbx, rax
0x11d46e:    call   0x1048b0

```

```

0x11d433:    test    eax, eax
0x11d435:    jne     0x11d418
0x11d3ef:    mov     r12, rbx
0x11d3f2:    pop     rbx
0x11d3f3:    pop     rbp
0x11d3f4:    mov     rax, r12
0x11d3f7:    pop     r12
0x11d3f9:    ret
0x11d473:    test    r12, r12
0x11d476:    je      0x11d4a0
0x11d437:    xor     r12d, r12d
0x11d43a:    pop     rbx
0x11d43b:    pop     rbp
0x11d43c:    mov     rax, r12
0x11d43f:    pop     r12
0x11d441:    ret
0x11d4a0:    mov     rax, qword ptr [rip + 0xe691]
0x11d4a7:    mov     qword ptr [rip + 0xe68a], rbx
0x11d4ae:    mov     qword ptr [rbx + 8], rax
0x11d4b2:    mov     rax, r12
0x11d4b5:    pop     rbx
0x11d4b6:    pop     rbp
0x11d4b7:    pop     r12
0x11d4b9:    ret
0x11d478:    mov     eax, dword ptr [r12 + 0x10]
0x11d47d:    mov     dword ptr [rbx], eax
0x11d47f:    mov     rax, qword ptr [rip + 0xe6ba]
0x11d486:    mov     qword ptr [rip + 0xe6b3], rbx
0x11d48d:    mov     qword ptr [rbx + 8], rax
0x11d491:    jmp     0x11d3ef
0x11d3cd:    nop     dword ptr [rax]
0x11d3fa:    nop     word ptr [rax + rax]
0x11d413:    nop     dword ptr [rax + rax]
0x11d442:    nop     word ptr [rax + rax]
0x11d496:    nop     word ptr cs:[rax + rax]
0x11d4ba:    nop     word ptr [rax + rax]
0x11d4c0:    endbr64
0x11d4c4:    push    r12
0x11d4c6:    push    rbp
0x11d4c7:    mov     ebp, edi
0x11d4c9:    push    rbx
0x11d4ca:    mov     rbx, qword ptr [rip + 0xe65f]
0x11d4d1:    test    rbx, rbx
0x11d4d4:    jne     0x11d4e9
0x11d4d6:    jmp     0x11d500
0x11d4e9:    cmp     dword ptr [rbx], ebp
0x11d4eb:    jne     0x11d4e0
0x11d500:    mov     edi, ebp

```

```

0x11d502:    lea     r12, [rip + 0x4edc]
0x11d509:    call    0x104a10
0x11d4ed:    xor     eax, eax
0x11d4ef:    cmp     byte ptr [rbx + 0x10], 0
0x11d4f3:    je      0x11d4f9
0x11d4e0:    mov     rbx, qword ptr [rbx + 8]
0x11d4e4:    test    rbx, rbx
0x11d4e7:    je      0x11d500
0x11d50e:    mov     edi, 0x11
0x11d513:    test    rax, rax
0x11d516:    je      0x11d527
0x11d4f9:    pop     rbx
0x11d4fa:    pop     rbp
0x11d4fb:    pop     r12
0x11d4fd:    ret
0x11d4f5:    lea     rax, [rbx + 0x10]
0x11d4f9:    pop     rbx
0x11d4fa:    pop     rbp
0x11d4fb:    pop     r12
0x11d4fd:    ret
0x11d527:    call    0x11d670
0x11d518:    mov     r12, qword ptr [rax]
0x11d51b:    mov     rdi, r12
0x11d51e:    call    0x1049b0
0x11d52c:    mov     rsi, r12
0x11d52f:    mov     dword ptr [rax], ebp
0x11d531:    lea     rdi, [rax + 0x10]
0x11d535:    mov     rbx, rax
0x11d538:    call    0x1048b0
0x11d523:    lea     rdi, [rax + 0x11]
0x11d527:    call    0x11d670
0x11d53d:    mov     rax, qword ptr [rip + 0xe5ec]
0x11d544:    mov     qword ptr [rip + 0xe5e5], rbx
0x11d54b:    mov     qword ptr [rbx + 8], rax
0x11d54f:    jmp     0x11d4ed
0x11d4d8:    nop     dword ptr [rax + rax]
0x11d4fe:    nop
0x11d551:    nop     word ptr cs:[rax + rax]
0x11d55c:    nop     dword ptr [rax]
0x11d560:    endbr64
0x11d564:    push    r12
0x11d566:    push    rbp
0x11d567:    mov     rbp, rdi
0x11d56a:    push    rbx
0x11d56b:    mov     rbx, qword ptr [rip + 0xe5be]
0x11d572:    test    rbx, rbx
0x11d575:    je      0x11d5b0
0x11d5b0:    mov     rbx, qword ptr [rip + 0xe571]

```

```

0x11d5b7:    test    rbx, rbx
0x11d5ba:    je      0x11d5f8
0x11d577:    movzx   r12d, byte ptr [rdi]
0x11d57b:    jmp     0x11d589
0x11d5f8:    mov     rdi, rbp
0x11d5fb:    call    0x104bc0
0x11d5bc:    movzx   r12d, byte ptr [rbp]
0x11d5c1:    jmp     0x11d5d1
0x11d589:    cmp     byte ptr [rbx + 0x10], r12b
0x11d58d:    jne     0x11d580
0x11d600:    mov     rdi, rbp
0x11d603:    mov     r12, rax
0x11d606:    call    0x1049b0
0x11d5d1:    cmp     byte ptr [rbx + 0x10], r12b
0x11d5d5:    jne     0x11d5c8
0x11d58f:    lea     rdi, [rbx + 0x10]
0x11d593:    mov     rsi, rbp
0x11d596:    call    0x104b30
0x11d580:    mov     rbx, qword ptr [rbx + 8]
0x11d584:    test    rbx, rbx
0x11d587:    je      0x11d5b0
0x11d60b:    lea     rdi, [rax + 0x11]
0x11d60f:    call    0x11d670
0x11d5d7:    lea     rdi, [rbx + 0x10]
0x11d5db:    mov     rsi, rbp
0x11d5de:    call    0x104b30
0x11d5c8:    mov     rbx, qword ptr [rbx + 8]
0x11d5cc:    test    rbx, rbx
0x11d5cf:    je      0x11d5f8
0x11d59b:    test    eax, eax
0x11d59d:    jne     0x11d580
0x11d614:    mov     rsi, rbp
0x11d617:    lea     rdi, [rax + 0x10]
0x11d61b:    mov     rbx, rax
0x11d61e:    call    0x1048b0
0x11d5e3:    test    eax, eax
0x11d5e5:    jne     0x11d5c8
0x11d59f:    mov     r12, rbx
0x11d5a2:    pop     rbx
0x11d5a3:    pop     rbp
0x11d5a4:    mov     rax, r12
0x11d5a7:    pop     r12
0x11d5a9:    ret
0x11d623:    test    r12, r12
0x11d626:    je      0x11d650
0x11d5e7:    xor     r12d, r12d
0x11d5ea:    pop     rbx
0x11d5eb:    pop     rbp

```

```

0x11d5ec:    mov     rax, r12
0x11d5ef:    pop     r12
0x11d5f1:    ret
0x11d650:    mov     rax, qword ptr [rip + 0xe4d1]
0x11d657:    mov     qword ptr [rip + 0xe4ca], rbx
0x11d65e:    mov     qword ptr [rbx + 8], rax
0x11d662:    mov     rax, r12
0x11d665:    pop     rbx
0x11d666:    pop     rbp
0x11d667:    pop     r12
0x11d669:    ret
0x11d628:    mov     eax, dword ptr [r12 + 0x10]
0x11d62d:    mov     dword ptr [rbx], eax
0x11d62f:    mov     rax, qword ptr [rip + 0xe4fa]
0x11d636:    mov     qword ptr [rip + 0xe4f3], rbx
0x11d63d:    mov     qword ptr [rbx + 8], rax
0x11d641:    jmp     0x11d59f
0x11d57d:    nop     dword ptr [rax]
0x11d5aa:    nop     word ptr [rax + rax]
0x11d5c3:    nop     dword ptr [rax + rax]
0x11d5f2:    nop     word ptr [rax + rax]
0x11d646:    nop     word ptr cs:[rax + rax]
0x11d66a:    nop     word ptr [rax + rax]
0x11d670:    endbr64
0x11d674:    push    rbx
0x11d675:    mov     rbx, rdi
0x11d678:    call    0x104c20
0x11d67d:    test    rax, rax
0x11d680:    jne     0x11d687
0x11d682:    test    rbx, rbx
0x11d685:    jne     0x11d689
0x11d687:    pop     rbx
0x11d688:    ret
0x11d689:    call    0x118cf0
0x11d68e:    nop
0x11d690:    endbr64
0x11d694:    push    rbx
0x11d695:    mov     rbx, rsi
0x11d698:    call    0x104ce0
0x11d69d:    test    rax, rax
0x11d6a0:    jne     0x11d6a7
0x11d6a2:    test    rbx, rbx
0x11d6a5:    jne     0x11d6a9
0x11d6a7:    pop     rbx
0x11d6a8:    ret
0x11d6a9:    call    0x118cf0
0x11d6ae:    nop
0x11d6b0:    endbr64

```



```

0x11d6b4:    push    rbx
0x11d6b5:    mov     rax, rsi
0x11d6b8:    mov     rsi, qword ptr [rsi]
0x11d6bb:    test    rdi, rdi
0x11d6be:    je      0x11d6f0
0x11d6f0:    mov     ebx, 1
0x11d6f5:    test    rsi, rsi
0x11d6f8:    jne     0x11d6dc
0x11d6c0:    movabs  rdx, 0xaaaaaaaaaaaaa9
0x11d6ca:    cmp     rsi, rdx
0x11d6cd:    ja      0x11d701
0x11d6fa:    mov     esi, 0x40
0x11d6ff:    jmp     0x11d6dc
0x11d6dc:    mov     qword ptr [rax], rsi
0x11d6df:    call    0x104ce0
0x11d6cf:    lea     rdx, [rsi + 1]
0x11d6d3:    shr     rdx, 1
0x11d6d6:    add     rsi, rdx
0x11d6d9:    setne   bl
0x11d6dc:    mov     qword ptr [rax], rsi
0x11d6df:    call    0x104ce0
0x11d701:    call    0x118cf0
0x11d6e4:    test    rax, rax
0x11d6e7:    jne     0x11d6ed
0x11d6e9:    test    bl, bl
0x11d6eb:    jne     0x11d701
0x11d6ed:    pop     rbx
0x11d6ee:    ret
0x11d706:    nop     word ptr cs:[rax + rax]
0x11d6ef:    nop
0x11d710:    endbr64
0x11d714:    push    rbx
0x11d715:    mov     esi, 1
0x11d71a:    mov     rbx, rdi
0x11d71d:    call    0x104b10
0x11d722:    test    rax, rax
0x11d725:    jne     0x11d72c
0x11d727:    test    rbx, rbx
0x11d72a:    jne     0x11d72e
0x11d72c:    pop     rbx
0x11d72d:    ret
0x11d72e:    call    0x118cf0
0x11d733:    nop     word ptr cs:[rax + rax]
0x11d73e:    nop
0x11d740:    endbr64
0x11d744:    sub     rsp, 8
0x11d748:    call    0x104b10
0x11d74d:    test    rax, rax

```

```

0x11d750:    je      0x11d757
0x11d757:    call    0x118cf0
0x11d752:    add     rsp, 8
0x11d756:    ret
0x11d75c:    nop     dword ptr [rax]
0x11d760:    endbr64
0x11d764:    push    r12
0x11d766:    mov     r12, rsi
0x11d769:    push    rbp
0x11d76a:    mov     rbp, rdi
0x11d76d:    mov     rdi, rsi
0x11d770:    sub     rsp, 8
0x11d774:    call    0x104c20
0x11d779:    mov     rdi, rax
0x11d77c:    test    rax, rax
0x11d77f:    jne     0x11d786
0x11d781:    test    r12, r12
0x11d784:    jne     0x11d798
0x11d786:    add     rsp, 8
0x11d78a:    mov     rdx, r12
0x11d78d:    mov     rsi, rbp
0x11d790:    pop     rbp
0x11d791:    pop     r12
0x11d793:    jmp     0x104bb0
0x11d798:    call    0x118cf0
0x11d79d:    nop     dword ptr [rax]
0x11d7a0:    endbr64
0x11d7a4:    push    r12
0x11d7a6:    push    rbp
0x11d7a7:    mov     rbp, rdi
0x11d7aa:    sub     rsp, 8
0x11d7ae:    call    0x1049b0
0x11d7b3:    lea     r12, [rax + 1]
0x11d7b7:    mov     rdi, r12
0x11d7ba:    call    0x104c20
0x11d7bf:    test    rax, rax
0x11d7c2:    je      0x11d7d9
0x11d7d9:    call    0x118cf0
0x11d7c4:    add     rsp, 8
0x11d7c8:    mov     rdx, r12
0x11d7cb:    mov     rsi, rbp
0x11d7ce:    mov     rdi, rax
0x11d7d1:    pop     rbp
0x11d7d2:    pop     r12
0x11d7d4:    jmp     0x104bb0
0x11d7de:    nop
0x11d7e0:    endbr64
0x11d7e4:    push    r15

```

```

0x11d7e6:    mov     r15, rsi
0x11d7e9:    push    r14
0x11d7eb:    mov     r14, rcx
0x11d7ee:    push    r13
0x11d7f0:    mov     r13d, edx
0x11d7f3:    push    r12
0x11d7f5:    mov     r12, r8
0x11d7f8:    push    rbp
0x11d7f9:    mov     rbp, rdi
0x11d7fc:    push    rbx
0x11d7fd:    sub     rsp, 0x28
0x11d801:    movzx   ebx, byte ptr [rdi]
0x11d804:    mov     rax, qword ptr fs:[0x28]
0x11d80d:    mov     qword ptr [rsp + 0x18], rax
0x11d812:    xor     eax, eax
0x11d814:    lea     rax, [rsp + 0x10]
0x11d819:    test    rsi, rsi
0x11d81c:    cmov     r15, rax
0x11d820:    call    0x104f00
0x11d825:    mov     rcx, qword ptr [rax]
0x11d828:    mov     rax, rbp
0x11d82b:    jmp     0x11d838
0x11d838:    movzx   edx, bl
0x11d83b:    test    byte ptr [rcx + rdx*2 + 1], 0x20
0x11d840:    jne     0x11d830
0x11d842:    cmp     bl, 0x2d
0x11d845:    je      0x11d8ca
0x11d830:    movzx   ebx, byte ptr [rax + 1]
0x11d834:    add     rax, 1
0x11d838:    movzx   edx, bl
0x11d83b:    test    byte ptr [rcx + rdx*2 + 1], 0x20
0x11d840:    jne     0x11d830
0x11d8ca:    mov     ebp, 4
0x11d8cf:    jmp     0x11d896
0x11d84b:    call    0x104860
0x11d896:    mov     rax, qword ptr [rsp + 0x18]
0x11d89b:    sub     rax, qword ptr fs:[0x28]
0x11d8a4:    jne     0x11dbf9
0x11d850:    mov     edx, r13d
0x11d853:    mov     rsi, r15
0x11d856:    mov     rdi, rbp
0x11d859:    mov     dword ptr [rax], 0
0x11d85f:    mov     qword ptr [rsp + 8], rax
0x11d864:    call    0x104dd0
0x11d8aa:    add     rsp, 0x28
0x11d8ae:    mov     eax, ebp
0x11d8b0:    pop     rbx
0x11d8b1:    pop     rbp

```

```

0x11d8b2:    pop     r12
0x11d8b4:    pop     r13
0x11d8b6:    pop     r14
0x11d8b8:    pop     r15
0x11d8ba:    ret
0x11dbf9:    call    0x1049d0
0x11d869:    mov     r13, qword ptr [r15]
0x11d86c:    mov     rcx, qword ptr [rsp + 8]
0x11d871:    mov     rbx, rax
0x11d874:    cmp     r13, rbp
0x11d877:    je      0x11d8d8
0x11d8d8:    test    r12, r12
0x11d8db:    je      0x11d8ca
0x11d879:    mov     eax, dword ptr [rcx]
0x11d87b:    test    eax, eax
0x11d87d:    jne     0x11d8c0
0x11d8dd:    movzx   edx, byte ptr [rbp]
0x11d8e1:    test    dl, dl
0x11d8e3:    je      0x11d8ca
0x11d87f:    xor     ebp, ebp
0x11d881:    test    r12, r12
0x11d884:    je      0x11d893
0x11d8c0:    mov     ebp, 1
0x11d8c5:    cmp     eax, 0x22
0x11d8c8:    je      0x11d881
0x11d8e5:    movsx   esi, dl
0x11d8e8:    mov     rdi, r12
0x11d8eb:    mov     byte ptr [rsp + 8], dl
0x11d8ef:    xor     ebp, ebp
0x11d8f1:    call    0x104a00
0x11d893:    mov     qword ptr [r14], rbx
0x11d896:    mov     rax, qword ptr [rsp + 0x18]
0x11d89b:    sub     rax, qword ptr fs:[0x28]
0x11d8a4:    jne     0x11dbf9
0x11d886:    movzx   edx, byte ptr [r13]
0x11d88b:    test    dl, dl
0x11d88d:    jne     0x11d958
0x11d881:    test    r12, r12
0x11d884:    je      0x11d893
0x11d8f6:    mov     ebx, 1
0x11d8fb:    movzx   edx, byte ptr [rsp + 8]
0x11d900:    test    rax, rax
0x11d903:    je      0x11d8ca
0x11d958:    movsx   esi, dl
0x11d95b:    mov     rdi, r12
0x11d95e:    mov     byte ptr [rsp + 8], dl
0x11d962:    call    0x104a00
0x11d905:    mov     esi, 0x30

```

```

0x11d90a:    mov     rdi, r12
0x11d90d:    mov     byte ptr [rsp + 8], dl
0x11d911:    call    0x104a00
0x11d967:    movzx   edx, byte ptr [rsp + 8]
0x11d96c:    test    rax, rax
0x11d96f:    jne     0x11d905
0x11d916:    movzx   edx, byte ptr [rsp + 8]
0x11d91b:    test    rax, rax
0x11d91e:    je      0x11d97c
0x11d971:    mov     qword ptr [r14], rbx
0x11d974:    or      ebp, 2
0x11d977:    jmp     0x11d896
0x11d97c:    mov     esi, 1
0x11d981:    mov     ecx, 0x400
0x11d986:    sub     edx, 0x42
0x11d989:    cmp     dl, 0x35
0x11d98c:    ja      0x11d971
0x11d920:    movzx   eax, byte ptr [r13 + 1]
0x11d925:    cmp     al, 0x44
0x11d927:    je      0x11d9a2
0x11d98e:    lea     rdi, [rip + 0x4c93]
0x11d995:    movzx   edx, dl
0x11d998:    movsxd  rax, dword ptr [rdi + rdx*4]
0x11d99c:    add     rax, rdi
0x11d99f:    jmp     rax
0x11d9a2:    mov     esi, 2
0x11d9a7:    mov     ecx, 0x3e8
0x11d9ac:    jmp     0x11d986
0x11d929:    cmp     al, 0x69
0x11d92b:    je      0x11dbe0
0x11da20:    movsxd  rsi, esi
0x11da23:    mov     rax, rbx
0x11da26:    mul     rcx
0x11da29:    jo      0x11dbfe
0x11da02:    movsxd  rdx, ecx
0x11da05:    movsxd  rsi, esi
0x11da08:    mov     rax, rbx
0x11da0b:    mul     rdx
0x11da0e:    mov     rbx, rax
0x11da11:    jno     0x11d9d7
0x11db06:    movsxd  rsi, esi
0x11db09:    mov     edi, 7
0x11db0e:    xor     r8d, r8d
0x11db11:    nop     dword ptr [rax]
0x11db18:    mov     rax, rbx
0x11db1b:    mul     rcx
0x11db1e:    mov     rbx, rax
0x11db21:    jo      0x11dc4a

```

```

0x11da4a:    movsxd  rsi, esi
0x11da4d:    mov     edi, 5
0x11da52:    xor     r8d, r8d
0x11da55:    nop     dword ptr [rax]
0x11da58:    mov     rax, rbx
0x11da5b:    mul     rcx
0x11da5e:    mov     rbx, rax
0x11da61:    jo      0x11dc1d
0x11d9ed:    movsxd  rsi, esi
0x11d9f0:    mov     rax, rbx
0x11d9f3:    mul     rcx
0x11d9f6:    jo      0x11da13
0x11d9ae:    movsxd  rsi, esi
0x11d9b1:    mov     edi, 4
0x11d9b6:    xor     r8d, r8d
0x11d9b9:    nop     dword ptr [rax]
0x11d9c0:    mov     rax, rbx
0x11d9c3:    mul     rcx
0x11d9c6:    mov     rbx, rax
0x11d9c9:    jo      0x11dc0e
0x11daed:    movsxd  rsi, esi
0x11daf0:    mov     rax, rbx
0x11daf3:    shr     rax, 0x37
0x11daf7:    jne     0x11da13
0x11da71:    movsxd  rsi, esi
0x11da74:    mov     edi, 6
0x11da79:    xor     r8d, r8d
0x11da7c:    nop     dword ptr [rax]
0x11da80:    mov     rax, rbx
0x11da83:    mul     rcx
0x11da86:    mov     rbx, rax
0x11da89:    jo      0x11dc2c
0x11dab2:    movsxd  rsi, esi
0x11dab5:    mov     edi, 8
0x11daba:    xor     r8d, r8d
0x11dabd:    nop     dword ptr [rax]
0x11dac0:    mov     rax, rbx
0x11dac3:    mul     rcx
0x11dac6:    mov     rbx, rax
0x11dac9:    jo      0x11dc3b
0x11db31:    movsxd  rsi, esi
0x11db34:    jmp     0x11d9d7
0x11dad9:    movsxd  rsi, esi
0x11dadc:    test    rbx, rbx
0x11dadf:    js      0x11da13
0x11da99:    movsxd  rsi, esi
0x11da9c:    mov     rax, rbx
0x11da9f:    shr     rax, 0x36

```

```

0x11daa3:    jne     0x11da13
0x11d986:    sub     edx, 0x42
0x11d989:    cmp     dl, 0x35
0x11d98c:    ja      0x11d971
0x11dbe0:    xor     esi, esi
0x11dbe2:    cmp     byte ptr [r13 + 2], 0x42
0x11dbe7:    mov     ecx, 0x400
0x11dbec:    sete    sil
0x11dbf0:    lea     esi, [rsi + rsi + 1]
0x11dbf4:    jmp     0x11d986
0x11d931:    cmp     al, 0x42
0x11d933:    je      0x11d9a2
0x11dbfe:    mov     eax, 1
0x11dc03:    or      rbx, 0xffffffffffffffff
0x11dc07:    or      ebp, eax
0x11dc09:    jmp     0x11d9d7
0x11da2f:    mul     rcx
0x11da32:    jo      0x11dbfe
0x11da13:    mov     ebp, 1
0x11da18:    or      rbx, 0xffffffffffffffff
0x11da1c:    jmp     0x11d9d7
0x11d9d7:    add     rsi, r13
0x11d9da:    mov     eax, ebp
0x11d9dc:    or      eax, 2
0x11d9df:    mov     qword ptr [r15], rsi
0x11d9e2:    cmp     byte ptr [rsi], 0
0x11d9e5:    cmovne  ebp, eax
0x11d9e8:    jmp     0x11d893
0x11dc4a:    mov     r8d, 1
0x11dc50:    or      rbx, 0xffffffffffffffff
0x11dc54:    jmp     0x11db27
0x11db27:    sub     edi, 1
0x11db2a:    jne     0x11db18
0x11dc1d:    mov     r8d, 1
0x11dc23:    or      rbx, 0xffffffffffffffff
0x11dc27:    jmp     0x11da67
0x11da67:    sub     edi, 1
0x11da6a:    jne     0x11da58
0x11d9f8:    mul     rcx
0x11d9fb:    jo      0x11da13
0x11dc0e:    mov     r8d, 1
0x11dc14:    or      rbx, 0xffffffffffffffff
0x11dc18:    jmp     0x11d9cf
0x11d9cf:    sub     edi, 1
0x11d9d2:    jne     0x11d9c0
0x11dafd:    shl     rbx, 9
0x11db01:    jmp     0x11d9d7
0x11dc2c:    mov     r8d, 1

```

```

0x11dc32:    or      rbx, 0xffffffffffffffff
0x11dc36:    jmp     0x11da8f
0x11da8f:    sub     edi, 1
0x11da92:    jne     0x11da80
0x11dc3b:    mov     r8d, 1
0x11dc41:    or      rbx, 0xffffffffffffffff
0x11dc45:    jmp     0x11dacf
0x11dacf:    sub     edi, 1
0x11dad2:    jne     0x11dac0
0x11dae5:    add     rbx, rbx
0x11dae8:    jmp     0x11d9d7
0x11daa9:    shl     rbx, 0xa
0x11daad:    jmp     0x11d9d7
0x11d935:    sub     edx, 0x42
0x11d938:    cmp     dl, 0x35
0x11d93b:    ja      0x11d971
0x11da38:    mul     rcx
0x11da3b:    jo      0x11dbfe
0x11db2c:    jmp     0x11d9d4
0x11db18:    mov     rax, rbx
0x11db1b:    mul     rcx
0x11db1e:    mov     rbx, rax
0x11db21:    jo      0x11dc4a
0x11da6c:    jmp     0x11d9d4
0x11da58:    mov     rax, rbx
0x11da5b:    mul     rcx
0x11da5e:    mov     rbx, rax
0x11da61:    jo      0x11dc1d
0x11d9fd:    mov     rbx, rax
0x11da00:    jmp     0x11d9d7
0x11d9d4:    or      ebp, r8d
0x11d9d7:    add     rsi, r13
0x11d9da:    mov     eax, ebp
0x11d9dc:    or      eax, 2
0x11d9df:    mov     qword ptr [r15], rsi
0x11d9e2:    cmp     byte ptr [rsi], 0
0x11d9e5:    cmovne  ebp, eax
0x11d9e8:    jmp     0x11d893
0x11d9c0:    mov     rax, rbx
0x11d9c3:    mul     rcx
0x11d9c6:    mov     rbx, rax
0x11d9c9:    jo      0x11dc0e
0x11da94:    jmp     0x11d9d4
0x11da80:    mov     rax, rbx
0x11da83:    mul     rcx
0x11da86:    mov     rbx, rax
0x11da89:    jo      0x11dc2c
0x11dad4:    jmp     0x11d9d4

```



```

0x11dac0:    mov     rax, rbx
0x11dac3:    mul     rcx
0x11dac6:    mov     rbx, rax
0x11dac9:    jo      0x11dc3b
0x11d93d:    lea     rcx, [rip + 0x4c0c]
0x11d944:    movzx   edx, dl
0x11d947:    movsxd  rax, dword ptr [rcx + rdx*4]
0x11d94b:    add     rax, rcx
0x11d94e:    jmp     rax
0x11da41:    mov     rbx, rax
0x11da44:    xor     eax, eax
0x11da46:    or      ebp, eax
0x11da48:    jmp     0x11d9d7
0x11dbc0:    mov     esi, 1
0x11dbc5:    jmp     0x11daf0
0x11dba2:    mov     esi, 1
0x11dba7:    jmp     0x11da9c
0x11db84:    mov     esi, 1
0x11db89:    mov     ecx, 0x400
0x11db8e:    jmp     0x11da4d
0x11db66:    mov     esi, 1
0x11db6b:    mov     ecx, 0x400
0x11db70:    jmp     0x11da23
0x11db48:    mov     esi, 1
0x11db4d:    mov     ecx, 0x400
0x11db52:    jmp     0x11d9f0
0x11dbca:    mov     esi, 1
0x11dbcf:    mov     ecx, 0x400
0x11dbd4:    jmp     0x11db09
0x11dbac:    mov     esi, 1
0x11dbb1:    jmp     0x11dadc
0x11db93:    mov     esi, 1
0x11db98:    mov     ecx, 0x400
0x11db9d:    jmp     0x11da74
0x11db75:    mov     esi, 1
0x11db7a:    mov     ecx, 0x400
0x11db7f:    jmp     0x11dab5
0x11dbb6:    mov     esi, 1
0x11dbbb:    jmp     0x11d9d7
0x11db57:    mov     esi, 1
0x11db5c:    mov     edx, 0x400
0x11db61:    jmp     0x11da08
0x11db39:    mov     esi, 1
0x11db3e:    mov     ecx, 0x400
0x11db43:    jmp     0x11d9b1
0x11daf0:    mov     rax, rbx
0x11daf3:    shr     rax, 0x37
0x11daf7:    jne     0x11da13

```

```

0x11da9c:    mov     rax, rbx
0x11da9f:    shr     rax, 0x36
0x11daa3:    jne     0x11da13
0x11da4d:    mov     edi, 5
0x11da52:    xor     r8d, r8d
0x11da55:    nop     dword ptr [rax]
0x11da58:    mov     rax, rbx
0x11da5b:    mul     rcx
0x11da5e:    mov     rbx, rax
0x11da61:    jo      0x11dc1d
0x11da23:    mov     rax, rbx
0x11da26:    mul     rcx
0x11da29:    jo      0x11dbfe
0x11d9f0:    mov     rax, rbx
0x11d9f3:    mul     rcx
0x11d9f6:    jo      0x11da13
0x11db09:    mov     edi, 7
0x11db0e:    xor     r8d, r8d
0x11db11:    nop     dword ptr [rax]
0x11db18:    mov     rax, rbx
0x11db1b:    mul     rcx
0x11db1e:    mov     rbx, rax
0x11db21:    jo      0x11dc4a
0x11dad3:    test    rbx, rbx
0x11dadf:    js      0x11da13
0x11da74:    mov     edi, 6
0x11da79:    xor     r8d, r8d
0x11da7c:    nop     dword ptr [rax]
0x11da80:    mov     rax, rbx
0x11da83:    mul     rcx
0x11da86:    mov     rbx, rax
0x11da89:    jo      0x11dc2c
0x11dab5:    mov     edi, 8
0x11daba:    xor     r8d, r8d
0x11dabd:    nop     dword ptr [rax]
0x11dac0:    mov     rax, rbx
0x11dac3:    mul     rcx
0x11dac6:    mov     rbx, rax
0x11dac9:    jo      0x11dc3b
0x11da08:    mov     rax, rbx
0x11da0b:    mul     rdx
0x11da0e:    mov     rbx, rax
0x11da11:    jno     0x11d9d7
0x11d9b1:    mov     edi, 4
0x11d9b6:    xor     r8d, r8d
0x11d9b9:    nop     dword ptr [rax]
0x11d9c0:    mov     rax, rbx
0x11d9c3:    mul     rcx

```

```

0x11d9c6:    mov     rbx, rax
0x11d9c9:    jo      0x11dc0e
0x11d82d:    nop     dword ptr [rax]
0x11d8bb:    nop     dword ptr [rax + rax]
0x11d8d1:    nop     dword ptr [rax]
0x11d951:    nop     dword ptr [rax]
0x11da1e:    nop
0x11dbd9:    nop     dword ptr [rax]
0x11dc59:    nop     dword ptr [rax]
0x11dc60:    endbr64
0x11dc64:    push    rax
0x11dc65:    pop     rax
0x11dc66:    sub     rsp, 8
0x11dc6a:    mov     edi, dword ptr [rip + 0xcb90]
0x11dc70:    call    0x104e20
0x11dc75:    nop     word ptr cs:[rax + rax]
0x11dc80:    endbr64
0x11dc84:    push    r15
0x11dc86:    push    r14
0x11dc88:    mov     r14, rdi
0x11dc8b:    push    r13
0x11dc8d:    mov     r13, rcx
0x11dc90:    push    r12
0x11dc92:    push    rbp
0x11dc93:    mov     rbp, rdx
0x11dc96:    push    rbx
0x11dc97:    mov     rbx, rsi
0x11dc9a:    sub     rsp, 0x28
0x11dc9e:    mov     qword ptr [rsp + 8], rsi
0x11dca3:    mov     qword ptr [rsp + 0x18], rdx
0x11dca8:    call    0x1049b0
0x11dcad:    mov     r15, qword ptr [rbx]
0x11dcb0:    test    r15, r15
0x11dcb3:    je      0x11dd86
0x11dd86:    mov     qword ptr [rsp], -1
0x11dd8e:    jmp     0x11dd65
0x11dcb9:    mov     byte ptr [rsp + 0x17], 0
0x11dcbe:    mov     r12, rax
0x11dcc1:    xor     ebx, ebx
0x11dcc3:    mov     qword ptr [rsp], -1
0x11dccb:    jmp     0x11dd18
0x11dd65:    mov     rax, qword ptr [rsp]
0x11dd69:    add     rsp, 0x28
0x11dd6d:    pop     rbx
0x11dd6e:    pop     rbp
0x11dd6f:    pop     r12
0x11dd71:    pop     r13
0x11dd73:    pop     r14

```

```

0x11dd75:    pop     r15
0x11dd77:    ret
0x11dd18:    mov     rdx, r12
0x11dd1b:    mov     rsi, r14
0x11dd1e:    mov     rdi, r15
0x11dd21:    call    0x104890
0x11dd26:    test    eax, eax
0x11dd28:    jne     0x11dd03
0x11dd2a:    mov     rdi, r15
0x11dd2d:    call    0x1049b0
0x11dd03:    mov     rax, qword ptr [rsp + 8]
0x11dd08:    add     rbx, 1
0x11dd0c:    add     rbp, r13
0x11dd0f:    mov     r15, qword ptr [rax + rbx*8]
0x11dd13:    test    r15, r15
0x11dd16:    je      0x11dd50
0x11dd32:    cmp     rax, r12
0x11dd35:    je      0x11dd80
0x11dd50:    cmp     byte ptr [rsp + 0x17], 0
0x11dd55:    mov     rax, -2
0x11dd5c:    cmov     rax, qword ptr [rsp]
0x11dd61:    mov     qword ptr [rsp], rax
0x11dd65:    mov     rax, qword ptr [rsp]
0x11dd69:    add     rsp, 0x28
0x11dd6d:    pop     rbx
0x11dd6e:    pop     rbp
0x11dd6f:    pop     r12
0x11dd71:    pop     r13
0x11dd73:    pop     r14
0x11dd75:    pop     r15
0x11dd77:    ret
0x11dd80:    mov     qword ptr [rsp], rbx
0x11dd84:    jmp     0x11dd65
0x11dd37:    cmp     qword ptr [rsp], -1
0x11dd3c:    jne     0x11dcd0
0x11dd3e:    mov     qword ptr [rsp], rbx
0x11dd42:    jmp     0x11dd03
0x11dcd0:    mov     rax, qword ptr [rsp + 0x18]
0x11dcd5:    test    rax, rax
0x11dcd8:    je      0x11dd48
0x11dd48:    mov     byte ptr [rsp + 0x17], 1
0x11dd4d:    jmp     0x11dd03
0x11dcda:    mov     rdi, qword ptr [rsp]
0x11dcde:    mov     rdx, r13
0x11dce1:    mov     rsi, rbp
0x11dce4:    imul    rdi, r13
0x11dce8:    add     rdi, rax
0x11dceb:    call    0x104af0

```

```

0x11dcf0:    mov     ecx, 1
0x11dcf5:    test    eax, eax
0x11dcf7:    movzx   eax, byte ptr [rsp + 0x17]
0x11dcfc:    cmovne  eax, ecx
0x11dcff:    mov     byte ptr [rsp + 0x17], al
0x11dd03:    mov     rax, qword ptr [rsp + 8]
0x11dd08:    add     rbx, 1
0x11dd0c:    add     rbp, r13
0x11dd0f:    mov     r15, qword ptr [rax + rbx*8]
0x11dd13:    test    r15, r15
0x11dd16:    je      0x11dd50
0x11ddcd:    nop     dword ptr [rax]
0x11dd44:    nop     dword ptr [rax]
0x11dd4f:    nop
0x11dd78:    nop     dword ptr [rax + rax]
0x11dd90:    endbr64
0x11dd94:    push    r13
0x11dd96:    cmp     rdx, -1
0x11dd9a:    mov     r13, rsi
0x11dd9d:    lea     rax, [rip + 0x4977]
0x11dda4:    push    r12
0x11dda6:    mov     rsi, rdi
0x11dda9:    lea     r12, [rip + 0x4950]
0x11ddb0:    mov     edi, 1
0x11ddb5:    push    rbx
0x11ddb6:    cmovne  r12, rax
0x11ddba:    call    0x11e000
0x11ddbf:    mov     rdx, r13
0x11ddc2:    mov     esi, 5
0x11ddc7:    xor     edi, edi
0x11ddc9:    mov     rbx, rax
0x11ddcc:    call    0x11ee20
0x11ddd1:    mov     r8, rbx
0x11ddd4:    mov     rdx, r12
0x11ddd7:    pop     rbx
0x11ddd8:    mov     rcx, rax
0x11dddb:    pop     r12
0x11dddd:    xor     esi, esi
0x11dddf:    xor     edi, edi
0x11dde1:    xor     eax, eax
0x11dde3:    pop     r13
0x11dde5:    jmp     0x104d50
0x11ddea:    nop     word ptr [rax + rax]
0x11ddf0:    endbr64
0x11ddf4:    push    r15
0x11ddf6:    push    r14
0x11ddf8:    lea     r14, [rip + 0x494e]
0x11ddff:    push    r13

```

```

0x11de01:    xor     r13d, r13d
0x11de04:    push    r12
0x11de06:    mov     r12, rdx
0x11de09:    mov     edx, 0x14
0x11de0e:    push    rbp
0x11de0f:    mov     rbp, rsi
0x11de12:    mov     esi, 1
0x11de17:    push    rbx
0x11de18:    mov     rbx, rdi
0x11de1b:    sub     rsp, 0x18
0x11de1f:    mov     rcx, qword ptr [rip + 0xca5a]
0x11de26:    mov     qword ptr [rsp + 8], rdi
0x11de2b:    lea     rdi, [rip + 0x4906]
0x11de32:    call    0x104e40
0x11de37:    mov     r15, qword ptr [rbx]
0x11de3a:    xor     ebx, ebx
0x11de3c:    test    r15, r15
0x11de3f:    jne     0x11de75
0x11de41:    jmp     0x11dec0
0x11de75:    mov     r8, qword ptr [rip + 0xca04]
0x11de7c:    test    rbx, rbx
0x11de7f:    je      0x11de48
0x11dec0:    mov     rsi, qword ptr [rip + 0xc9b9]
0x11dec7:    add     rsp, 0x18
0x11decb:    mov     edi, 0xa
0x11ded0:    pop     rbx
0x11ded1:    pop     rbp
0x11ded2:    pop     r12
0x11ded4:    pop     r13
0x11ded6:    pop     r14
0x11ded8:    pop     r15
0x11deda:    jmp     0x104b40
0x11de48:    mov     rcx, r15
0x11de4b:    mov     rdx, r14
0x11de4e:    mov     esi, 1
0x11de53:    mov     rdi, r8
0x11de56:    xor     eax, eax
0x11de58:    mov     r13, rbp
0x11de5b:    call    0x104e60
0x11de81:    mov     rdx, r12
0x11de84:    mov     rsi, rbp
0x11de87:    mov     rdi, r13
0x11de8a:    mov     qword ptr [rsp], r8
0x11de8e:    call    0x104af0
0x11de60:    mov     rax, qword ptr [rsp + 8]
0x11de65:    add     rbx, 1
0x11de69:    add     rbp, r12
0x11de6c:    mov     r15, qword ptr [rax + rbx*8]

```

```

0x11de70:    test    r15, r15
0x11de73:    je      0x11dec0
0x11de93:    mov     r8, qword ptr [rsp]
0x11de97:    test    eax, eax
0x11de99:    jne     0x11de48
0x11de9b:    mov     rcx, r15
0x11de9e:    lea     rdx, [rip + 0x48b2]
0x11dea5:    mov     rdi, r8
0x11dea8:    xor     eax, eax
0x11deaa:    mov     esi, 1
0x11deaf:    call    0x104e60
0x11deb4:    jmp     0x11de60
0x11de43:    nop     dword ptr [rax + rax]
0x11deb6:    nop     word ptr cs:[rax + rax]
0x11dedf:    nop
0x11dee0:    endbr64
0x11dee4:    push    r15
0x11dee6:    mov     r15, rsi
0x11dee9:    push    r14
0x11deeb:    mov     r14, r8
0x11deee:    push    r13
0x11def0:    mov     r13, rdi
0x11def3:    mov     rdi, r15
0x11def6:    push    r12
0x11def8:    mov     r12, rcx
0x11defb:    mov     rcx, r8
0x11defe:    push    rbp
0x11deff:    mov     rbp, rdx
0x11df02:    mov     rdx, r12
0x11df05:    push    rbx
0x11df06:    mov     rsi, rbp
0x11df09:    mov     rbx, r9
0x11df0c:    sub     rsp, 0x18
0x11df10:    call    0x11dc80
0x11df15:    test    rax, rax
0x11df18:    js      0x11df30
0x11df30:    cmp     rax, -1
0x11df34:    lea     r9, [rip + 0x47c5]
0x11df3b:    mov     rsi, r13
0x11df3e:    mov     edi, 1
0x11df43:    lea     rax, [rip + 0x47d1]
0x11df4a:    cmovne  r9, rax
0x11df4e:    mov     qword ptr [rsp + 8], r9
0x11df53:    call    0x11e000
0x11df1a:    add     rsp, 0x18
0x11df1e:    pop     rbx
0x11df1f:    pop     rbp
0x11df20:    pop     r12

```

```

0x11df22:    pop     r13
0x11df24:    pop     r14
0x11df26:    pop     r15
0x11df28:    ret
0x11df58:    mov     rdx, r15
0x11df5b:    mov     esi, 5
0x11df60:    xor     edi, edi
0x11df62:    mov     r13, rax
0x11df65:    call    0x11ee20
0x11df6a:    mov     rdx, qword ptr [rsp + 8]
0x11df6f:    mov     r8, r13
0x11df72:    xor     esi, esi
0x11df74:    mov     rcx, rax
0x11df77:    xor     edi, edi
0x11df79:    xor     eax, eax
0x11df7b:    call    0x104d50
0x11df80:    mov     rdx, r14
0x11df83:    mov     rsi, r12
0x11df86:    mov     rdi, rbp
0x11df89:    call    0x11ddf0
0x11df8e:    call    rbx
0x11df90:    mov     rax, -1
0x11df97:    jmp     0x11df1a
0x11df29:    nop     dword ptr [rax]
0x11df99:    nop     dword ptr [rax]
0x11dfa0:    endbr64
0x11dfa4:    push    r14
0x11dfa6:    push    r13
0x11dfa8:    push    r12
0x11dfaa:    push    rbp
0x11dfab:    push    rbx
0x11dfac:    mov     r14, qword ptr [rsi]
0x11dfaf:    test    r14, r14
0x11dfb2:    je      0x11dfe9
0x11dfe9:    pop     rbx
0x11dfea:    mov     rax, r14
0x11dfed:    pop     rbp
0x11dfee:    pop     r12
0x11dff0:    pop     r13
0x11dff2:    pop     r14
0x11dff4:    ret
0x11dfb4:    mov     r12, rdi
0x11dfb7:    mov     r13, rcx
0x11dfba:    lea     rbx, [rsi + 8]
0x11dfbe:    mov     rbp, rdx
0x11dfc1:    jmp     0x11dfd7
0x11dfd7:    mov     rdx, r13
0x11dfda:    mov     rsi, rbp

```



```

0x11dfdd:    mov     rdi, r12
0x11dfe0:    call   0x104af0
0x11dfe5:    test    eax, eax
0x11dfe7:    jne     0x11dfc8
0x11dfc8:    mov     r14, qword ptr [rbx]
0x11dfcb:    add     rbp, r13
0x11dfce:    add     rbx, 8
0x11dfd2:    test    r14, r14
0x11dfd5:    je      0x11dfe9
0x11dfc3:    nop     dword ptr [rax + rax]
0x11dff5:    nop     word ptr cs:[rax + rax]
0x11dfff:    nop
0x11e000:    endbr64
0x11e004:    mov     rdx, rsi
0x11e007:    mov     esi, 5
0x11e00c:    jmp     0x11ee20
0x11e011:    nop     word ptr cs:[rax + rax]
0x11e01c:    nop     dword ptr [rax]
0x11e020:    endbr64
0x11e024:    mov     rdx, rdi
0x11e027:    mov     esi, 5
0x11e02c:    xor     edi, edi
0x11e02e:    jmp     0x11ee20
0x11e033:    nop     word ptr cs:[rax + rax]
0x11e03d:    nop     dword ptr [rax]
0x11e040:    push    r15
0x11e042:    push    r14
0x11e044:    push    r13
0x11e046:    mov     r13, rcx
0x11e049:    push    r12
0x11e04b:    push    rbp
0x11e04c:    push    rbx
0x11e04d:    mov     ebx, r8d
0x11e050:    sub     rsp, 0x88
0x11e057:    mov     qword ptr [rsp + 8], rdi
0x11e05c:    mov     qword ptr [rsp + 0x30], rsi
0x11e061:    mov     qword ptr [rsp + 0x20], rdx
0x11e066:    mov     dword ptr [rsp + 0x10], r8d
0x11e06b:    mov     qword ptr [rsp + 0x18], r9
0x11e070:    mov     rax, qword ptr fs:[0x28]
0x11e079:    mov     qword ptr [rsp + 0x78], rax
0x11e07e:    xor     eax, eax
0x11e080:    call   0x1049a0
0x11e085:    cmp     ebx, 4
0x11e088:    mov     r11, qword ptr [rsp + 0x20]
0x11e08d:    mov     r10, qword ptr [rsp + 0x30]
0x11e092:    mov     qword ptr [rsp + 0x28], rax
0x11e097:    jne     0x11e832

```

```

0x11e09d:    mov     qword ptr [rsp + 0x20], 0
0x11e0a6:    mov     r8d, 1
0x11e0ac:    xor     r9d, r9d
0x11e0af:    nop
0x11e0b0:    mov     rbx, r9
0x11e0b3:    xor     r12d, r12d
0x11e0b6:    nop     word ptr cs:[rax + rax]
0x11e0c0:    cmp     r13, r12
0x11e0c3:    setne   al
0x11e0c6:    cmp     r13, -1
0x11e0ca:    jne     0x11e0d4
0x11e832:    mov     eax, dword ptr [rsp + 0x10]
0x11e836:    jbe     0x11e88a
0x11e0cc:    cmp     byte ptr [r11 + r12], 0
0x11e0d1:    setne   al
0x11e0d4:    test    al, al
0x11e0d6:    je      0x11e7ae
0x11e0d4:    test    al, al
0x11e0d6:    je      0x11e7ae
0x11e88a:    cmp     eax, 2
0x11e88d:    je      0x11e39e
0x11e838:    sub     eax, 5
0x11e83b:    cmp     eax, 1
0x11e83e:    ja      0x11e8c8
0x11e7ae:    test    r12, r12
0x11e7b1:    jne     0x11e7be
0x11e0dc:    lea     rsi, [r11 + r12]
0x11e0e0:    movzx   edx, byte ptr [rsi]
0x11e0e3:    test    r9, r9
0x11e0e6:    je      0x11e4b0
0x11e39e:    lea     rax, [rip + 0x43b7]
0x11e3a5:    mov     dword ptr [rsp + 0x10], 2
0x11e3ad:    mov     r9d, 1
0x11e3b3:    xor     r8d, r8d
0x11e3b6:    mov     qword ptr [rsp + 0x20], rax
0x11e3bb:    test    r10, r10
0x11e3be:    je      0x11e0b0
0x11e893:    cmp     eax, 3
0x11e896:    jne     0x11e8c8
0x11e844:    cmp     dword ptr [rsp + 0x10], 6
0x11e849:    mov     edx, 0x60
0x11e84e:    mov     eax, 0x22
0x11e853:    cmovne  eax, edx
0x11e856:    lea     rcx, [rip + 0x2a3b]
0x11e85d:    lea     rdx, [rip + 0x3ef8]
0x11e864:    cmov    rdx, rcx
0x11e868:    mov     qword ptr [rsp + 0x20], rdx
0x11e86d:    test    r10, r10

```

```

0x11e870:    je      0x11e879
0x11e8c8:    mov     qword ptr [rsp + 0x20], 0
0x11e8d1:    xor     r8d, r8d
0x11e8d4:    xor     r9d, r9d
0x11e8d7:    jmp     0x11e0b0
0x11e7b3:    cmp     dword ptr [rsp + 0x10], 1
0x11e7b8:    je      0x11e380
0x11e7be:    mov     rsi, qword ptr [rsp + 0x20]
0x11e7c3:    mov     rax, rbx
0x11e7c6:    test    rsi, rsi
0x11e7c9:    je      0x11e7f4
0x11e4b0:    mov     ebp, edx
0x11e4b2:    cmp     dl, 0x3f
0x11e4b5:    jg      0x11e6d8
0x11e0ec:    test    r8b, r8b
0x11e0ef:    je      0x11e4b0
0x11e0b0:    mov     rbx, r9
0x11e0b3:    xor     r12d, r12d
0x11e0b6:    nop     word ptr cs:[rax + rax]
0x11e0c0:    cmp     r13, r12
0x11e0c3:    setne   al
0x11e0c6:    cmp     r13, -1
0x11e0ca:    jne     0x11e0d4
0x11e3c4:    mov     rax, qword ptr [rsp + 8]
0x11e3c9:    mov     byte ptr [rax], 0x27
0x11e3cc:    jmp     0x11e0b0
0x11e898:    test    r10, r10
0x11e89b:    je      0x11e8dc
0x11e879:    mov     r8d, 1
0x11e87f:    mov     r9d, 1
0x11e885:    jmp     0x11e0b0
0x11e872:    mov     rsi, qword ptr [rsp + 8]
0x11e877:    mov     byte ptr [rsi], al
0x11e879:    mov     r8d, 1
0x11e87f:    mov     r9d, 1
0x11e885:    jmp     0x11e0b0
0x11e380:    mov     qword ptr [rsp + 0x20], r11
0x11e385:    mov     qword ptr [rsp + 0x10], r10
0x11e38a:    call    0x1049a0
0x11e7f4:    cmp     rax, r10
0x11e7f7:    jae     0x11e802
0x11e7cb:    movzx   edx, byte ptr [rsi]
0x11e7ce:    test    dl, dl
0x11e7d0:    je      0x11e7f4
0x11e4bb:    test    dl, dl
0x11e4bd:    js      0x11e500
0x11e6d8:    cmp     dl, 0x7a
0x11e6db:    jg      0x11e4e0

```

```

0x11e0f5:    lea     r14, [r12 + 1]
0x11e0fa:    mov     r15, r14
0x11e0fd:    cmp     r14, r13
0x11e100:    ja      0x11e120
0x11e8dc:    lea     rax, [rip + 0x29b5]
0x11e8e3:    mov     r8d, 1
0x11e8e9:    mov     r9d, 1
0x11e8ef:    mov     qword ptr [rsp + 0x20], rax
0x11e8f4:    jmp     0x11e0b0
0x11e89d:    mov     rax, qword ptr [rsp + 8]
0x11e8a2:    mov     r8d, 1
0x11e8a8:    mov     r9d, 1
0x11e8ae:    mov     byte ptr [rax], 0x22
0x11e8b1:    lea     rax, [rip + 0x29e0]
0x11e8b8:    mov     qword ptr [rsp + 0x20], rax
0x11e8bd:    jmp     0x11e0b0
0x11e38f:    mov     r10, qword ptr [rsp + 0x10]
0x11e394:    mov     r11, qword ptr [rsp + 0x20]
0x11e399:    mov     qword ptr [rsp + 0x28], rax
0x11e39e:    lea     rax, [rip + 0x43b7]
0x11e3a5:    mov     dword ptr [rsp + 0x10], 2
0x11e3ad:    mov     r9d, 1
0x11e3b3:    xor     r8d, r8d
0x11e3b6:    mov     qword ptr [rsp + 0x20], rax
0x11e3bb:    test    r10, r10
0x11e3be:    je      0x11e0b0
0x11e7f9:    mov     rsi, qword ptr [rsp + 8]
0x11e7fe:    mov     byte ptr [rsi + rax], 0
0x11e802:    mov     rdx, qword ptr [rsp + 0x78]
0x11e807:    sub     rdx, qword ptr fs:[0x28]
0x11e810:    jne     0x11e9ca
0x11e802:    mov     rdx, qword ptr [rsp + 0x78]
0x11e807:    sub     rdx, qword ptr fs:[0x28]
0x11e810:    jne     0x11e9ca
0x11e7d2:    sub     rsi, rbx
0x11e7d5:    mov     rcx, rsi
0x11e7d8:    mov     rsi, qword ptr [rsp + 8]
0x11e7dd:    nop     dword ptr [rax]
0x11e7e0:    cmp     r10, rax
0x11e7e3:    jbe     0x11e7e8
0x11e500:    lea     r14, [r12 + 1]
0x11e505:    cmp     qword ptr [rsp + 0x28], 1
0x11e50b:    mov     byte ptr [rsp + 0x30], dl
0x11e50f:    mov     r15, r14
0x11e512:    je      0x11e250
0x11e4bf:    cmp     dl, 0x3f
0x11e4c2:    ja      0x11e500
0x11e6e1:    cmp     dl, 0x40

```

```

0x11e6e4:    je      0x11e500
0x11e4e0:    cmp     dl, 0x7d
0x11e4e3:    je      0x11e72c
0x11e102:    mov     rax, qword ptr [rsp + 0x20]
0x11e107:    cmp     byte ptr [rax], dl
0x11e109:    jne     0x11e120
0x11e120:    mov     ebp, edx
0x11e122:    cmp     dl, 0x3f
0x11e125:    jg      0x11e458
0x11e816:    add     rsp, 0x88
0x11e81d:    pop     rbx
0x11e81e:    pop     rbp
0x11e81f:    pop     r12
0x11e821:    pop     r13
0x11e823:    pop     r14
0x11e825:    pop     r15
0x11e827:    ret
0x11e9ca:    call    0x1049d0
0x11e7e8:    add     rax, 1
0x11e7ec:    movzx   edx, byte ptr [rcx + rax]
0x11e7f0:    test    dl, dl
0x11e7f2:    jne     0x11e7e0
0x11e7e5:    mov     byte ptr [rsi + rax], dl
0x11e7e8:    add     rax, 1
0x11e7ec:    movzx   edx, byte ptr [rcx + rax]
0x11e7f0:    test    dl, dl
0x11e7f2:    jne     0x11e7e0
0x11e250:    mov     qword ptr [rsp + 0x50], r11
0x11e255:    mov     qword ptr [rsp + 0x48], r10
0x11e25a:    mov     byte ptr [rsp + 0x40], r8b
0x11e25f:    mov     qword ptr [rsp + 0x38], r9
0x11e264:    call    0x104f00
0x11e518:    mov     qword ptr [rsp + 0x70], 0
0x11e521:    lea     rcx, [rsp + 0x70]
0x11e526:    cmp     r13, -1
0x11e52a:    jne     0x11e574
0x11e4c4:    lea     rdi, [rip + 0x43c5]
0x11e4cb:    movzx   eax, dl
0x11e4ce:    movsxd  rax, dword ptr [rdi + rax*4]
0x11e4d2:    add     rax, rdi
0x11e4d5:    jmp     rax
0x11e6ea:    lea     ecx, [rbp - 0x41]
0x11e6ed:    mov     eax, 1
0x11e6f2:    movabs  rdi, 0x3fffffff53ffffff
0x11e6fc:    shl     rax, cl
0x11e6ff:    test    rax, rdi
0x11e702:    jne     0x11e215
0x11e72c:    cmp     r13, 1

```

```

0x11e730:      setne    al
0x11e733:      cmp     r13, -1
0x11e737:      jne     0x11e741
0x11e4e9:      jle     0x11e1f8
0x11e10b:      cmp     rbx, r10
0x11e10e:      jae     0x11e11c
0x11e12b:      test    dl, dl
0x11e12d:      js      0x11e240
0x11e458:      cmp     dl, 0x7a
0x11e45b:      jg      0x11e220
0x11e7e0:      cmp     r10, rax
0x11e7e3:      jbe     0x11e7e8
0x11e269:      movzx   edx, byte ptr [rsp + 0x30]
0x11e26e:      movzx   r8d, byte ptr [rsp + 0x40]
0x11e274:      mov     ecx, 1
0x11e279:      mov     rax, qword ptr [rax]
0x11e27c:      mov     r9, qword ptr [rsp + 0x38]
0x11e281:      mov     r10, qword ptr [rsp + 0x48]
0x11e286:      mov     r11, qword ptr [rsp + 0x50]
0x11e28b:      movzx   eax, word ptr [rax + rdx*2]
0x11e28f:      shr     ax, 0xe
0x11e293:      xor     eax, 1
0x11e296:      and     eax, r8d
0x11e299:      test    al, al
0x11e29b:      je      0x11e1a4
0x11e52c:      mov     rdi, r11
0x11e52f:      mov     qword ptr [rsp + 0x58], r10
0x11e534:      mov     byte ptr [rsp + 0x50], r8b
0x11e539:      mov     qword ptr [rsp + 0x48], r9
0x11e53e:      mov     qword ptr [rsp + 0x40], rsi
0x11e543:      mov     qword ptr [rsp + 0x38], rcx
0x11e548:      mov     qword ptr [rsp + 0x30], r11
0x11e54d:      call    0x1049b0
0x11e574:      mov     rdx, r13
0x11e577:      lea     rdi, [rsp + 0x6c]
0x11e57c:      mov     qword ptr [rsp + 0x48], r11
0x11e581:      sub     rdx, r12
0x11e584:      mov     qword ptr [rsp + 0x40], r10
0x11e589:      mov     byte ptr [rsp + 0x38], r8b
0x11e58e:      mov     qword ptr [rsp + 0x30], r9
0x11e593:      mov     qword ptr [rsp + 0x58], rdx
0x11e598:      mov     qword ptr [rsp + 0x50], rsi
0x11e59d:      call    0x1049f0
0x11e360:      mov     ebp, 0x23
0x11e365:      test    r12, r12
0x11e368:      jne     0x11e215
0x11e320:      mov     eax, dword ptr [rsp + 0x10]
0x11e324:      cmp     eax, 1

```

```

0x11e327:    je      0x11e380
0x11e6a4:    mov     ebp, 8
0x11e6a9:    mov     eax, 0x62
0x11e6ae:    jmp     0x11e650
0x11e668:    mov     ebp, 0xb
0x11e66d:    mov     eax, 0x76
0x11e672:    jmp     0x11e650
0x11e6c8:    mov     ebp, 0xd
0x11e6cd:    mov     eax, 0x72
0x11e6d2:    jmp     0x11e67e
0x11e20a:    cmp     dword ptr [rsp + 0x10], 1
0x11e20f:    je      0x11e380
0x11e690:    add     r12, 1
0x11e694:    test    r8b, r8b
0x11e697:    jne     0x11e3f5
0x11e6b0:    mov     ebp, 0xa
0x11e6b5:    mov     eax, 0x6e
0x11e6ba:    jmp     0x11e67e
0x11e150:    mov     eax, dword ptr [rsp + 0x10]
0x11e154:    cmp     eax, 1
0x11e157:    je      0x11e380
0x11e674:    mov     ebp, 9
0x11e679:    mov     eax, 0x74
0x11e67e:    cmp     dword ptr [rsp + 0x10], 1
0x11e683:    jne     0x11e650
0x11e215:    lea     r15, [r12 + 1]
0x11e21a:    jmp     0x11e1a4
0x11e6bc:    mov     ebp, 7
0x11e6c1:    mov     eax, 0x61
0x11e6c6:    jmp     0x11e650
0x11e63f:    mov     ebp, 0xc
0x11e644:    mov     eax, 0x66
0x11e649:    nop     dword ptr [rax]
0x11e650:    add     r12, 1
0x11e654:    test    r8b, r8b
0x11e657:    je      0x11e1e1
0x11e708:    test    eax, 0xa4000000
0x11e70d:    jne     0x11e20a
0x11e739:    cmp     byte ptr [r11 + 1], 0
0x11e73e:    setne   al
0x11e741:    test    al, al
0x11e743:    jne     0x11e215
0x11e741:    test    al, al
0x11e743:    jne     0x11e215
0x11e1f8:    cmp     dl, 0x7b
0x11e1fb:    je      0x11e72c
0x11e4ef:    cmp     dl, 0x7e
0x11e4f2:    je      0x11e365

```

```

0x11e110:    mov     rax, qword ptr [rsp + 8]
0x11e115:    mov     byte ptr [rax + rbx], 0x5c
0x11e119:    movzx   edx, byte ptr [rsi]
0x11e11c:    add     rbx, 1
0x11e120:    mov     ebp, edx
0x11e122:    cmp     dl, 0x3f
0x11e125:    jg      0x11e458
0x11e11c:    add     rbx, 1
0x11e120:    mov     ebp, edx
0x11e122:    cmp     dl, 0x3f
0x11e125:    jg      0x11e458
0x11e240:    cmp     qword ptr [rsp + 0x28], 1
0x11e246:    mov     byte ptr [rsp + 0x30], dl
0x11e24a:    jne     0x11e518
0x11e133:    cmp     dl, 0x3f
0x11e136:    ja      0x11e240
0x11e461:    cmp     dl, 0x40
0x11e464:    je      0x11e240
0x11e220:    cmp     dl, 0x7d
0x11e223:    je      0x11e72c
0x11e1a4:    test    r8b, r8b
0x11e1a7:    je      0x11e9cf
0x11e2a1:    mov     eax, r8d
0x11e2a4:    mov     rsi, qword ptr [rsp + 8]
0x11e2a9:    add     rcx, r12
0x11e2ac:    jmp     0x11e2c6
0x11e552:    mov     r10, qword ptr [rsp + 0x58]
0x11e557:    movzx   r8d, byte ptr [rsp + 0x50]
0x11e55d:    mov     r9, qword ptr [rsp + 0x48]
0x11e562:    mov     rsi, qword ptr [rsp + 0x40]
0x11e567:    mov     r13, rax
0x11e56a:    mov     rcx, qword ptr [rsp + 0x38]
0x11e56f:    mov     r11, qword ptr [rsp + 0x30]
0x11e574:    mov     rdx, r13
0x11e577:    lea     rdi, [rsp + 0x6c]
0x11e57c:    mov     qword ptr [rsp + 0x48], r11
0x11e581:    sub     rdx, r12
0x11e584:    mov     qword ptr [rsp + 0x40], r10
0x11e589:    mov     byte ptr [rsp + 0x38], r8b
0x11e58e:    mov     qword ptr [rsp + 0x30], r9
0x11e593:    mov     qword ptr [rsp + 0x58], rdx
0x11e598:    mov     qword ptr [rsp + 0x50], rsi
0x11e59d:    call    0x1049f0
0x11e5a2:    mov     r9, qword ptr [rsp + 0x30]
0x11e5a7:    movzx   r8d, byte ptr [rsp + 0x38]
0x11e5ad:    test    rax, rax
0x11e5b0:    mov     r10, qword ptr [rsp + 0x40]
0x11e5b5:    mov     r11, qword ptr [rsp + 0x48]

```



```

0x11e5ba:    mov     rcx, rax
0x11e5bd:    je      0x11e1a4
0x11e36e:    cmp     dword ptr [rsp + 0x10], 1
0x11e373:    jne     0x11e215
0x11e329:    cmp     eax, 3
0x11e32c:    je      0x11e760
0x11e650:    add     r12, 1
0x11e654:    test    r8b, r8b
0x11e657:    je      0x11e1e1
0x11e67e:    cmp     dword ptr [rsp + 0x10], 1
0x11e683:    jne     0x11e650
0x11e69d:    xor     ebp, ebp
0x11e69f:    jmp     0x11e1e1
0x11e3f5:    cmp     r10, rbx
0x11e3f8:    jbe     0x11e403
0x11e15d:    cmp     eax, 2
0x11e160:    jne     0x11e19a
0x11e685:    jmp     0x11e380
0x11e1e1:    cmp     rbx, r10
0x11e1e4:    jae     0x11e1ef
0x11e65d:    mov     ebp, eax
0x11e65f:    jmp     0x11e1cf
0x11e713:    cmp     bpl, 0x5c
0x11e717:    jne     0x11e500
0x11e749:    jmp     0x11e365
0x11e201:    cmp     dl, 0x7c
0x11e204:    jne     0x11e500
0x11e365:    test    r12, r12
0x11e368:    jne     0x11e215
0x11e4f8:    nop     dword ptr [rax + rax]
0x11e13c:    lea     rcx, [rip + 0x464d]
0x11e143:    movzx   eax, dl
0x11e146:    movsxd  rax, dword ptr [rcx + rax*4]
0x11e14a:    add     rax, rcx
0x11e14d:    jmp     rax
0x11e46a:    lea     ecx, [rbp - 0x41]
0x11e46d:    mov     eax, 1
0x11e472:    movabs  rdi, 0x3fffffff53ffffff
0x11e47c:    shl     rax, cl
0x11e47f:    test    rax, rdi
0x11e482:    jne     0x11e1b0
0x11e229:    jle     0x11e727
0x11e9cf:    mov     r12, r15
0x11e9d2:    jmp     0x11e1e1
0x11e1ad:    nop     dword ptr [rax]
0x11e2c6:    test    al, al
0x11e2c8:    je      0x11e30a
0x11e5c3:    cmp     rax, -1

```

```

0x11e5c7:    je      0x11e828
0x11e379:    nop     dword ptr [rax]
0x11e760:    lea     rax, [r12 + 2]
0x11e765:    lea     r15, [r12 + 1]
0x11e76a:    mov     ebp, 0x3f
0x11e76f:    cmp     rax, r13
0x11e772:    jae     0x11e1a4
0x11e332:    lea     r15, [r12 + 1]
0x11e337:    mov     ebp, 0x3f
0x11e33c:    jmp     0x11e1a4
0x11e403:    lea     rax, [rbx + 1]
0x11e407:    cmp     r10, rax
0x11e40a:    jbe     0x11e416
0x11e3fa:    mov     rax, qword ptr [rsp + 8]
0x11e3ff:    mov     byte ptr [rax + rbx], 0x5c
0x11e403:    lea     rax, [rbx + 1]
0x11e407:    cmp     r10, rax
0x11e40a:    jbe     0x11e416
0x11e162:    cmp     r10, rbx
0x11e165:    jbe     0x11e170
0x11e19a:    lea     r15, [r12 + 1]
0x11e19f:    mov     ebp, 0x27
0x11e1a4:    test    r8b, r8b
0x11e1a7:    je      0x11e9cf
0x11e1e6:    mov     rax, qword ptr [rsp + 8]
0x11e1eb:    mov     byte ptr [rax + rbx], bpl
0x11e1ef:    add     rbx, 1
0x11e1f3:    jmp     0x11e0c0
0x11e1ef:    add     rbx, 1
0x11e1f3:    jmp     0x11e0c0
0x11e1cf:    cmp     rbx, r10
0x11e1d2:    jae     0x11e1dd
0x11e71d:    mov     eax, 0x5c
0x11e722:    jmp     0x11e67e
0x11e441:    mov     ebp, 0x61
0x11e446:    jmp     0x11e34d
0x11e341:    mov     ebp, 0x72
0x11e346:    cmp     dword ptr [rsp + 0x10], 1
0x11e34b:    je      0x11e380
0x11e3e5:    mov     r12, r14
0x11e3e8:    mov     ebp, 0x76
0x11e3ed:    jmp     0x11e1cf
0x11e44b:    mov     ebp, 0x6e
0x11e450:    jmp     0x11e346
0x11e3d1:    mov     ebp, 0x74
0x11e3d6:    jmp     0x11e346
0x11e3f2:    mov     r12, r14
0x11e3f5:    cmp     r10, rbx

```

```

0x11e3f8:    jbe     0x11e403
0x11e355:    mov     ebp, 0x62
0x11e35a:    jmp     0x11e34d
0x11e3db:    mov     ebp, 0x66
0x11e3e0:    jmp     0x11e34d
0x11e488:    test    eax, 0xa4000000
0x11e48d:    jne     0x11e20a
0x11e1b0:    mov     edx, ebp
0x11e1b2:    mov     eax, 1
0x11e1b7:    mov     ecx, ebp
0x11e1b9:    mov     r12, r15
0x11e1bc:    shr     dl, 5
0x11e1bf:    shl     eax, cl
0x11e1c1:    movzx   edx, dl
0x11e1c4:    mov     rsi, qword ptr [rsp + 0x18]
0x11e1c9:    and     eax, dword ptr [rsi + rdx*4 + 4]
0x11e1cd:    je      0x11e1e1
0x11e727:    cmp     dl, 0x7b
0x11e72a:    jne     0x11e750
0x11e22f:    cmp     dl, 0x7e
0x11e232:    je      0x11e365
0x11e30a:    mov     r12, r14
0x11e30d:    cmp     r14, rcx
0x11e310:    jb      0x11e2b0
0x11e2ca:    cmp     rbx, r10
0x11e2cd:    jae     0x11e2d3
0x11e828:    mov     eax, r8d
0x11e82b:    xor     ecx, ecx
0x11e82d:    jmp     0x11e299
0x11e5cd:    cmp     rax, -2
0x11e5d1:    mov     rsi, qword ptr [rsp + 0x50]
0x11e5d6:    mov     rdx, qword ptr [rsp + 0x58]
0x11e5db:    je      0x11e93b
0x11e778:    cmp     byte ptr [r11 + r12 + 1], 0x3f
0x11e77e:    jne     0x11e1a4
0x11e416:    lea     rax, [rbx + 2]
0x11e41a:    cmp     r10, rax
0x11e41d:    jbe     0x11e429
0x11e40c:    mov     rax, qword ptr [rsp + 8]
0x11e411:    mov     byte ptr [rax + rbx + 1], 0x30
0x11e416:    lea     rax, [rbx + 2]
0x11e41a:    cmp     r10, rax
0x11e41d:    jbe     0x11e429
0x11e170:    lea     rax, [rbx + 1]
0x11e174:    cmp     r10, rax
0x11e177:    jbe     0x11e183
0x11e167:    mov     rax, qword ptr [rsp + 8]
0x11e16c:    mov     byte ptr [rax + rbx], 0x27

```

```

0x11e170:    lea     rax, [rbx + 1]
0x11e174:    cmp     r10, rax
0x11e177:    jbe     0x11e183
0x11e0c0:    cmp     r13, r12
0x11e0c3:    setne   al
0x11e0c6:    cmp     r13, -1
0x11e0ca:    jne     0x11e0d4
0x11e1d4:    mov     rax, qword ptr [rsp + 8]
0x11e1d9:    mov     byte ptr [rax + rbx], 0x5c
0x11e1dd:    add     rbx, 1
0x11e1e1:    cmp     rbx, r10
0x11e1e4:    jae     0x11e1ef
0x11e1dd:    add     rbx, 1
0x11e1e1:    cmp     rbx, r10
0x11e1e4:    jae     0x11e1ef
0x11e34d:    mov     r12, r14
0x11e350:    jmp     0x11e1cf
0x11e346:    cmp     dword ptr [rsp + 0x10], 1
0x11e34b:    je      0x11e380
0x11e493:    cmp     bpl, 0x5c
0x11e497:    jne     0x11e240
0x11e750:    cmp     dl, 0x7c
0x11e753:    je      0x11e20a
0x11e238:    nop     dword ptr [rax + rax]
0x11e2b0:    cmp     rbx, r10
0x11e2b3:    jae     0x11e2b9
0x11e312:    jmp     0x11e1e1
0x11e2cf:    mov     byte ptr [rsi + rbx], 0x5c
0x11e2d3:    lea     rdx, [rbx + 1]
0x11e2d7:    cmp     r10, rdx
0x11e2da:    jbe     0x11e2e8
0x11e2d3:    lea     rdx, [rbx + 1]
0x11e2d7:    cmp     r10, rdx
0x11e2da:    jbe     0x11e2e8
0x11e299:    test    al, al
0x11e29b:    je      0x11e1a4
0x11e93b:    xor     ecx, ecx
0x11e93d:    cmp     r13, r12
0x11e940:    ja      0x11e951
0x11e5e1:    cmp     dword ptr [rsp + 0x10], 1
0x11e5e6:    je      0x11e95f
0x11e784:    movzx   ebp, byte ptr [r11 + r12 + 2]
0x11e78a:    cmp     bpl, 0x3e
0x11e78e:    ja      0x11e7a4
0x11e429:    add     rbx, 3
0x11e42d:    mov     eax, 0x10000
0x11e432:    mov     edx, 1
0x11e437:    mov     ebp, 0x30

```

```

0x11e43c:      jmp      0x11e1c4
0x11e41f:      mov     rax, qword ptr [rsp + 8]
0x11e424:      mov     byte ptr [rax + rbx + 2], 0x30
0x11e429:      add     rbx, 3
0x11e42d:      mov     eax, 0x10000
0x11e432:      mov     edx, 1
0x11e437:      mov     ebp, 0x30
0x11e43c:      jmp     0x11e1c4
0x11e183:      lea     rax, [rbx + 2]
0x11e187:      cmp     r10, rax
0x11e18a:      jbe     0x11e196
0x11e179:      mov     rax, qword ptr [rsp + 8]
0x11e17e:      mov     byte ptr [rax + rbx + 1], 0x5c
0x11e183:      lea     rax, [rbx + 2]
0x11e187:      cmp     r10, rax
0x11e18a:      jbe     0x11e196
0x11e49d:      cmp     dword ptr [rsp + 0x10], 1
0x11e4a2:      je      0x11e380
0x11e759:      jmp     0x11e240
0x11e2b5:      mov     byte ptr [rsi + rbx], bpl
0x11e2b9:      movzx   ebp, byte ptr [r11 + r14]
0x11e2be:      add     rbx, 1
0x11e2c2:      add     r14, 1
0x11e2c6:      test    al, al
0x11e2c8:      je      0x11e30a
0x11e2b9:      movzx   ebp, byte ptr [r11 + r14]
0x11e2be:      add     rbx, 1
0x11e2c2:      add     r14, 1
0x11e2c6:      test    al, al
0x11e2c8:      je      0x11e30a
0x11e2e8:      lea     rdx, [rbx + 2]
0x11e2ec:      cmp     r10, rdx
0x11e2ef:      jbe     0x11e300
0x11e2dc:      mov     edx, ebp
0x11e2de:      shr     dl, 6
0x11e2e1:      add     edx, 0x30
0x11e2e4:      mov     byte ptr [rsi + rbx + 1], dl
0x11e2e8:      lea     rdx, [rbx + 2]
0x11e2ec:      cmp     r10, rdx
0x11e2ef:      jbe     0x11e300
0x11e942:      jmp     0x11e9c2
0x11e951:      cmp     byte ptr [rsi + rcx], 0
0x11e955:      jne     0x11e948
0x11e95f:      cmp     rax, 1
0x11e963:      je      0x11e5ec
0x11e5ec:      mov     edi, dword ptr [rsp + 0x6c]
0x11e5f0:      mov     qword ptr [rsp + 0x50], r11
0x11e5f5:      mov     qword ptr [rsp + 0x48], r10

```

```

0x11e5fa:    mov     byte ptr [rsp + 0x40], r8b
0x11e5ff:    mov     qword ptr [rsp + 0x38], r9
0x11e604:    mov     qword ptr [rsp + 0x30], rcx
0x11e609:    call    0x104e80
0x11e790:    movabs  rax, 0x7000a38200000000
0x11e79a:    bt      rax, rbp
0x11e79e:    jb      0x11e8f9
0x11e7a4:    mov     ebp, 0x3f
0x11e7a9:    jmp     0x11e1a4
0x11e1c4:    mov     rsi, qword ptr [rsp + 0x18]
0x11e1c9:    and     eax, dword ptr [rsi + rdx*4 + 4]
0x11e1cd:    je      0x11e1e1
0x11e196:    add     rbx, 3
0x11e19a:    lea     r15, [r12 + 1]
0x11e19f:    mov     ebp, 0x27
0x11e1a4:    test    r8b, r8b
0x11e1a7:    je      0x11e9cf
0x11e18c:    mov     rax, qword ptr [rsp + 8]
0x11e191:    mov     byte ptr [rax + rbx + 2], 0x27
0x11e196:    add     rbx, 3
0x11e19a:    lea     r15, [r12 + 1]
0x11e19f:    mov     ebp, 0x27
0x11e1a4:    test    r8b, r8b
0x11e1a7:    je      0x11e9cf
0x11e4a8:    jmp     0x11e34d
0x11e300:    and     ebp, 7
0x11e303:    add     rbx, 3
0x11e307:    add     ebp, 0x30
0x11e30a:    mov     r12, r14
0x11e30d:    cmp     r14, rcx
0x11e310:    jb      0x11e2b0
0x11e2f1:    mov     edx, ebp
0x11e2f3:    shr     dl, 3
0x11e2f6:    and     edx, 7
0x11e2f9:    add     edx, 0x30
0x11e2fc:    mov     byte ptr [rsi + rbx + 2], dl
0x11e300:    and     ebp, 7
0x11e303:    add     rbx, 3
0x11e307:    add     ebp, 0x30
0x11e30a:    mov     r12, r14
0x11e30d:    cmp     r14, rcx
0x11e310:    jb      0x11e2b0
0x11e9c2:    mov     eax, r8d
0x11e9c5:    jmp     0x11e299
0x11e957:    mov     eax, r8d
0x11e95a:    jmp     0x11e630
0x11e948:    add     rcx, 1
0x11e94c:    cmp     rdx, rcx

```

```

0x11e94f:    je      0x11e957
0x11e969:    lea     rdx, [r11 + r14]
0x11e96d:    add     rsi, rax
0x11e970:    jmp     0x11e985
0x11e60e:    movzx   r8d, byte ptr [rsp + 0x40]
0x11e614:    mov     rcx, qword ptr [rsp + 0x30]
0x11e619:    test    eax, eax
0x11e61b:    mov     r9, qword ptr [rsp + 0x38]
0x11e620:    mov     r10, qword ptr [rsp + 0x48]
0x11e625:    sete    al
0x11e628:    mov     r11, qword ptr [rsp + 0x50]
0x11e62d:    and     eax, r8d
0x11e630:    cmp     rcx, 1
0x11e634:    ja      0x11e2a4
0x11e8f9:    cmp     r10, rbx
0x11e8fc:    jbe     0x11e907
0x11e630:    cmp     rcx, 1
0x11e634:    ja      0x11e2a4
0x11e985:    movzx   eax, byte ptr [rdx]
0x11e988:    sub     eax, 0x5b
0x11e98b:    cmp     al, 0x21
0x11e98d:    ja      0x11e978
0x11e63a:    jmp     0x11e299
0x11e2a4:    mov     rsi, qword ptr [rsp + 8]
0x11e2a9:    add     rcx, r12
0x11e2ac:    jmp     0x11e2c6
0x11e907:    lea     rax, [rbx + 1]
0x11e90b:    cmp     r10, rax
0x11e90e:    jbe     0x11e91a
0x11e8fe:    mov     rax, qword ptr [rsp + 8]
0x11e903:    mov     byte ptr [rax + rbx], 0x3f
0x11e907:    lea     rax, [rbx + 1]
0x11e90b:    cmp     r10, rax
0x11e90e:    jbe     0x11e91a
0x11e98f:    movabs  rdi, 0x20000002b
0x11e999:    bt      rdi, rax
0x11e99d:    jae     0x11e978
0x11e978:    add     rdx, 1
0x11e97c:    cmp     rsi, rdx
0x11e97f:    je      0x11e5ec
0x11e91a:    lea     rax, [rbx + 2]
0x11e91e:    cmp     r10, rax
0x11e921:    jbe     0x11e92d
0x11e910:    mov     rax, qword ptr [rsp + 8]
0x11e915:    mov     byte ptr [rax + rbx + 1], 0x5c
0x11e91a:    lea     rax, [rbx + 2]
0x11e91e:    cmp     r10, rax
0x11e921:    jbe     0x11e92d

```

```

0x11e99f:    mov     qword ptr [rsp + 0x20], r10
0x11e9a4:    mov     qword ptr [rsp + 0x10], r11
0x11e9a9:    call    0x1049a0
0x11e92d:    add     rbx, 3
0x11e931:    lea     r15, [r12 + 3]
0x11e936:    jmp     0x11e1a4
0x11e923:    mov     rax, qword ptr [rsp + 8]
0x11e928:    mov     byte ptr [rax + rbx + 2], 0x3f
0x11e92d:    add     rbx, 3
0x11e931:    lea     r15, [r12 + 3]
0x11e936:    jmp     0x11e1a4
0x11e9ae:    mov     r11, qword ptr [rsp + 0x10]
0x11e9b3:    mov     r10, qword ptr [rsp + 0x20]
0x11e9b8:    mov     qword ptr [rsp + 0x28], rax
0x11e9bd:    jmp     0x11e39e
0x11e21c:    nop     dword ptr [rax]
0x11e2ae:    nop
0x11e317:    nop     word ptr [rax + rax]
0x11e35c:    nop     dword ptr [rax]
0x11e455:    nop     dword ptr [rax]
0x11e4ad:    nop     dword ptr [rax]
0x11e4d8:    nop     dword ptr [rax + rax]
0x11e664:    nop     dword ptr [rax]
0x11e68a:    nop     word ptr [rax + rax]
0x11e6d4:    nop     dword ptr [rax]
0x11e74e:    nop
0x11e75e:    nop
0x11e8c2:    nop     word ptr [rax + rax]
0x11e944:    nop     dword ptr [rax]
0x11e972:    nop     word ptr [rax + rax]
0x11e9d7:    nop     word ptr [rax + rax]
0x11e9e0:    push    r15
0x11e9e2:    mov     r15d, edi
0x11e9e5:    push    r14
0x11e9e7:    push    r13
0x11e9e9:    mov     r13, rdx
0x11e9ec:    push    r12
0x11e9ee:    mov     r12, rsi
0x11e9f1:    push    rbp
0x11e9f2:    mov     rbp, rcx
0x11e9f5:    push    rbx
0x11e9f6:    sub     rsp, 0x28
0x11e9fa:    call    0x104860
0x11e9ff:    mov     r14, qword ptr [rip + 0xbe0a]
0x11ea06:    mov     rbx, rax
0x11ea09:    mov     eax, dword ptr [rax]
0x11ea0b:    mov     dword ptr [rsp + 0xc], eax
0x11ea0f:    test    r15d, r15d

```



```

0x11ea12:      js      0x104f2a
0x11ea18:      cmp     dword ptr [rip + 0xbde5], r15d
0x11ea1f:      mov     r10d, dword ptr [rsp + 0xc]
0x11ea24:      ja      0x11ea89
0x11ea26:      lea     edx, [r15 + 1]
0x11ea2a:      lea     rax, [rip + 0xbdef]
0x11ea31:      mov     rsi, rdx
0x11ea34:      mov     dword ptr [rsp + 0x10], edx
0x11ea38:      shl     rsi, 4
0x11ea3c:      cmp     r14, rax
0x11ea3f:      je      0x11eb40
0x11ea89:      movsxd   r8, r15d
0x11ea8c:      mov     r9, rbp
0x11ea8f:      mov     rcx, r13
0x11ea92:      mov     rdx, r12
0x11ea95:      shl     r8, 4
0x11ea99:      mov     dword ptr [rsp + 0x18], r10d
0x11ea9e:      lea     r15, [r14 + r8]
0x11eaa2:      mov     r8d, dword ptr [rbp]
0x11eaa6:      mov     r14, qword ptr [r15 + 8]
0x11eaaa:      mov     rsi, qword ptr [r15]
0x11eaad:      mov     rdi, r14
0x11eab0:      mov     qword ptr [rsp + 0x10], rsi
0x11eab5:      call    0x11e040
0x11eb40:      xor     edi, edi
0x11eb42:      mov     qword ptr [rsp + 0x18], rdx
0x11eb47:      call    0x11d690
0x11ea45:      mov     rdi, r14
0x11ea48:      mov     qword ptr [rsp + 0x18], rdx
0x11ea4d:      call    0x11d690
0x11eaba:      mov     r10d, dword ptr [rsp + 0x18]
0x11eabf:      mov     rsi, qword ptr [rsp + 0x10]
0x11eac4:      mov     dword ptr [rbx], r10d
0x11eac7:      cmp     rax, rsi
0x11eaca:      jb      0x11eb1f
0x11eb4c:      movdqa   xmm0, xmmword ptr [rip + 0xbccc]
0x11eb54:      mov     rdx, qword ptr [rsp + 0x18]
0x11eb59:      mov     qword ptr [rip + 0xbcb0], rax
0x11eb60:      mov     r14, rax
0x11eb63:      movups   xmmword ptr [rax], xmm0
0x11eb66:      jmp     0x11ea61
0x11ea52:      mov     rdx, qword ptr [rsp + 0x18]
0x11ea57:      mov     qword ptr [rip + 0xbdb2], rax
0x11ea5e:      mov     r14, rax
0x11ea61:      mov     edi, dword ptr [rip + 0xbd9d]
0x11ea67:      xor     esi, esi
0x11ea69:      sub     rdx, rdi
0x11ea6c:      shl     rdi, 4

```

```

0x11ea70:    shl     rdx, 4
0x11ea74:    add     rdi, r14
0x11ea77:    call    0x104a50
0x11eb1f:    mov     eax, dword ptr [rsp + 0xc]
0x11eb23:    mov     dword ptr [rbx], eax
0x11eb25:    add     rsp, 0x28
0x11eb29:    mov     rax, r14
0x11eb2c:    pop     rbx
0x11eb2d:    pop     rbp
0x11eb2e:    pop     r12
0x11eb30:    pop     r13
0x11eb32:    pop     r14
0x11eb34:    pop     r15
0x11eb36:    ret
0x11eacc:    lea     rsi, [rax + 1]
0x11ead0:    lea     rax, [rip + 0xd089]
0x11ead7:    mov     qword ptr [r15], rsi
0x11eada:    cmp     r14, rax
0x11eadd:    je      0x11eaf1
0x11ea61:    mov     edi, dword ptr [rip + 0xbd9d]
0x11ea67:    xor     esi, esi
0x11ea69:    sub     rdx, rdi
0x11ea6c:    shl     rdi, 4
0x11ea70:    shl     rdx, 4
0x11ea74:    add     rdi, r14
0x11ea77:    call    0x104a50
0x11ea7c:    mov     eax, dword ptr [rsp + 0x10]
0x11ea80:    mov     r10d, dword ptr [rbx]
0x11ea83:    mov     dword ptr [rip + 0xbd7b], eax
0x11ea89:    movsxd  r8, r15d
0x11ea8c:    mov     r9, rbp
0x11ea8f:    mov     rcx, r13
0x11ea92:    mov     rdx, r12
0x11ea95:    shl     r8, 4
0x11ea99:    mov     dword ptr [rsp + 0x18], r10d
0x11ea9e:    lea     r15, [r14 + r8]
0x11eaa2:    mov     r8d, dword ptr [rbp]
0x11eaa6:    mov     r14, qword ptr [r15 + 8]
0x11eaaa:    mov     rsi, qword ptr [r15]
0x11eaad:    mov     rdi, r14
0x11eab0:    mov     qword ptr [rsp + 0x10], rsi
0x11eab5:    call    0x11e040
0x11eaf1:    mov     rdi, rsi
0x11eaf4:    mov     qword ptr [rsp + 0x10], rsi
0x11eaf9:    call    0x11d670
0x11eadf:    mov     rdi, r14
0x11eae2:    mov     qword ptr [rsp + 0x10], rsi
0x11eae7:    call    0x104800

```

```

0x11eafe:    mov     r8d, dword ptr [rbp]
0x11eb02:    mov     r9, rbp
0x11eb05:    mov     rcx, r13
0x11eb08:    mov     qword ptr [r15 + 8], rax
0x11eb0c:    mov     rsi, qword ptr [rsp + 0x10]
0x11eb11:    mov     rdx, r12
0x11eb14:    mov     rdi, rax
0x11eb17:    mov     r14, rax
0x11eb1a:    call    0x11e040
0x11eaec:    mov     rsi, qword ptr [rsp + 0x10]
0x11eaf1:    mov     rdi, rsi
0x11eaf4:    mov     qword ptr [rsp + 0x10], rsi
0x11eaf9:    call    0x11d670
0x11eb37:    nop     word ptr [rax + rax]
0x11eb6b:    nop     dword ptr [rax + rax]
0x11eb70:    endbr64
0x11eb74:    push    r12
0x11eb76:    push    rbp
0x11eb77:    mov     rbp, rdi
0x11eb7a:    push    rbx
0x11eb7b:    call    0x104860
0x11eb80:    test    rbp, rbp
0x11eb83:    mov     esi, 0x24
0x11eb88:    mov     r12d, dword ptr [rax]
0x11eb8b:    mov     rbx, rax
0x11eb8e:    lea     rax, [rip + 0xd0cb]
0x11eb95:    cmove   rbp, rax
0x11eb99:    mov     rdi, rbp
0x11eb9c:    call    0x11d760
0x11eba1:    mov     dword ptr [rbx], r12d
0x11eba4:    pop     rbx
0x11eba5:    pop     rbp
0x11eba6:    pop     r12
0x11eba8:    ret
0x11eba9:    nop     dword ptr [rax]
0x11ebb0:    endbr64
0x11ebb4:    test    rdi, rdi
0x11ebb7:    lea     rax, [rip + 0xd0a2]
0x11ebb8:    cmove   rdi, rax
0x11ebbc:    mov     eax, dword ptr [rdi]
0x11ebc4:    ret
0x11ebc5:    nop     word ptr cs:[rax + rax]
0x11ebd0:    endbr64
0x11ebd4:    test    rdi, rdi
0x11ebd7:    lea     rax, [rip + 0xd082]
0x11ebde:    cmove   rdi, rax
0x11ebe2:    mov     dword ptr [rdi], esi
0x11ebe4:    ret

```

```

0x11ebe5:      nop      word ptr cs:[rax + rax]
0x11ebf0:      endbr64
0x11ebf4:      test     rdi, rdi
0x11ebf7:      lea      rax, [rip + 0xd062]
0x11ebfe:      mov      ecx, esi
0x11ec00:      cmov     rdi, rax
0x11ec04:      mov      eax, esi
0x11ec06:      and      ecx, 0x1f
0x11ec09:      shr      al, 5
0x11ec0c:      movzx    eax, al
0x11ec0f:      lea      rsi, [rdi + rax*4 + 4]
0x11ec14:      mov      edi, dword ptr [rsi]
0x11ec16:      mov      eax, edi
0x11ec18:      shr      eax, cl
0x11ec1a:      xor      edx, eax
0x11ec1c:      and      eax, 1
0x11ec1f:      and      edx, 1
0x11ec22:      shl      edx, cl
0x11ec24:      xor      edx, edi
0x11ec26:      mov      dword ptr [rsi], edx
0x11ec28:      ret
0x11ec29:      nop      dword ptr [rax]
0x11ec30:      endbr64
0x11ec34:      push     r15
0x11ec36:      lea      rax, [rip + 0xd023]
0x11ec3d:      push     r14
0x11ec3f:      mov      r14, rdx
0x11ec42:      push     r13
0x11ec44:      mov      r13, rsi
0x11ec47:      push     r12
0x11ec49:      mov      r12, rdi
0x11ec4c:      push     rbp
0x11ec4d:      push     rbx
0x11ec4e:      mov      rbx, r8
0x11ec51:      sub      rsp, 0x18
0x11ec55:      test     r8, r8
0x11ec58:      cmov     rbx, rax
0x11ec5c:      mov      qword ptr [rsp + 8], rcx
0x11ec61:      call     0x104860
0x11ec66:      mov      rcx, qword ptr [rsp + 8]
0x11ec6b:      mov      rdx, r14
0x11ec6e:      mov      rsi, r13
0x11ec71:      mov      r15d, dword ptr [rax]
0x11ec74:      mov      r8d, dword ptr [rbx]
0x11ec77:      mov      rbp, rax
0x11ec7a:      mov      r9, rbx
0x11ec7d:      mov      rdi, r12
0x11ec80:      call     0x11e040

```

```

0x11ec85:    mov     dword ptr [rbp], r15d
0x11ec89:    add     rsp, 0x18
0x11ec8d:    pop     rbx
0x11ec8e:    pop     rbp
0x11ec8f:    pop     r12
0x11ec91:    pop     r13
0x11ec93:    pop     r14
0x11ec95:    pop     r15
0x11ec97:    ret
0x11ec98:    nop     dword ptr [rax + rax]
0x11eca0:    endbr64
0x11eca4:    push    r15
0x11eca6:    mov     r15, rdi
0x11eca9:    push    r14
0x11ecab:    mov     r14, rsi
0x11ecae:    push    r13
0x11ecb0:    push    r12
0x11ecb2:    push    rbp
0x11ecb3:    push    rbx
0x11ecb4:    mov     rbx, rdx
0x11ecb7:    sub     rsp, 0x18
0x11ecbb:    call    0x104860
0x11ecc0:    test    rbx, rbx
0x11ecc3:    mov     rcx, r14
0x11ecc6:    mov     rdx, r15
0x11ecc9:    mov     r12d, dword ptr [rax]
0x11eccc:    mov     rbp, rax
0x11eccf:    lea     rax, [rip + 0xcf8a]
0x11ecd6:    cmovbe  rbx, rax
0x11ecda:    xor     esi, esi
0x11ecdc:    xor     edi, edi
0x11ecde:    mov     r8d, dword ptr [rbx]
0x11ece1:    mov     r9, rbx
0x11ece4:    call    0x11e040
0x11ece9:    mov     dword ptr [rbp], r12d
0x11eced:    lea     r13, [rax + 1]
0x11ecf1:    mov     dword ptr [rsp + 0xc], r12d
0x11ecf6:    mov     rdi, r13
0x11ecf9:    call    0x11d670
0x11ecfe:    mov     r8d, dword ptr [rbx]
0x11ed01:    mov     r9, rbx
0x11ed04:    mov     rcx, r14
0x11ed07:    mov     rdx, r15
0x11ed0a:    mov     rsi, r13
0x11ed0d:    mov     rdi, rax
0x11ed10:    mov     r12, rax
0x11ed13:    call    0x11e040
0x11ed18:    mov     eax, dword ptr [rsp + 0xc]

```

```

0x11ed1c:    mov     dword ptr [rbp], eax
0x11ed1f:    add     rsp, 0x18
0x11ed23:    mov     rax, r12
0x11ed26:    pop     rbx
0x11ed27:    pop     rbp
0x11ed28:    pop     r12
0x11ed2a:    pop     r13
0x11ed2c:    pop     r14
0x11ed2e:    pop     r15
0x11ed30:    ret
0x11ed31:    nop     word ptr cs:[rax + rax]
0x11ed3c:    nop     dword ptr [rax]
0x11ed40:    endbr64
0x11ed44:    mov     eax, dword ptr [rip + 0xbaba]
0x11ed4a:    push    r12
0x11ed4c:    mov     r12, qword ptr [rip + 0xbabd]
0x11ed53:    push    rbp
0x11ed54:    push    rbx
0x11ed55:    cmp     eax, 1
0x11ed58:    jbe     0x11ed81
0x11ed81:    mov     rdi, qword ptr [r12 + 8]
0x11ed86:    lea     rbx, [rip + 0xcdd3]
0x11ed8d:    cmp     rdi, rbx
0x11ed90:    je      0x11eda9
0x11ed5a:    sub     eax, 2
0x11ed5d:    lea     rbx, [r12 + 0x18]
0x11ed62:    shl     rax, 4
0x11ed66:    lea     rbp, [r12 + rax + 0x28]
0x11ed6b:    nop     dword ptr [rax + rax]
0x11ed70:    mov     rdi, qword ptr [rbx]
0x11ed73:    add     rbx, 0x10
0x11ed77:    call    0x104800
0x11eda9:    lea     rbx, [rip + 0xba70]
0x11edb0:    cmp     r12, rbx
0x11edb3:    je      0x11edc4
0x11ed92:    call    0x104800
0x11ed7c:    cmp     rbx, rbp
0x11ed7f:    jne     0x11ed70
0x11edc4:    mov     dword ptr [rip + 0xba36], 1
0x11edce:    pop     rbx
0x11edcf:    pop     rbp
0x11edd0:    pop     r12
0x11edd2:    ret
0x11edb5:    mov     rdi, r12
0x11edb8:    call    0x104800
0x11ed97:    mov     qword ptr [rip + 0xba8a], rbx
0x11ed9e:    mov     qword ptr [rip + 0xba77], 0x100
0x11eda9:    lea     rbx, [rip + 0xba70]

```

```

0x11edb0:    cmp     r12, rbx
0x11edb3:    je      0x11edc4
0x11ed70:    mov     rdi, qword ptr [rbx]
0x11ed73:    add     rbx, 0x10
0x11ed77:    call    0x104800
0x11edbd:    mov     qword ptr [rip + 0xba4c], rbx
0x11edc4:    mov     dword ptr [rip + 0xba36], 1
0x11edce:    pop     rbx
0x11edcf:    pop     rbp
0x11edd0:    pop     r12
0x11edd2:    ret
0x11edd3:    nop     word ptr cs:[rax + rax]
0x11edde:    nop
0x11ede0:    endbr64
0x11ede4:    lea     rcx, [rip + 0xce75]
0x11edeb:    mov     rdx, -1
0x11edf2:    jmp     0x11e9e0
0x11edf7:    nop     word ptr [rax + rax]
0x11ee00:    endbr64
0x11ee04:    mov     rsi, rdi
0x11ee07:    lea     rcx, [rip + 0xce52]
0x11ee0e:    mov     rdx, -1
0x11ee15:    xor     edi, edi
0x11ee17:    jmp     0x11e9e0
0x11ee1c:    nop     dword ptr [rax]
0x11ee20:    endbr64
0x11ee24:    sub     rsp, 0x68
0x11ee28:    mov     r8d, esi
0x11ee2b:    pxor    xmm0, xmm0
0x11ee2f:    mov     rsi, rdx
0x11ee32:    mov     rax, qword ptr fs:[0x28]
0x11ee3b:    mov     qword ptr [rsp + 0x58], rax
0x11ee40:    xor     eax, eax
0x11ee42:    movups  xmmword ptr [rsp + 0x34], xmm0
0x11ee47:    mov     rdx, -1
0x11ee4e:    mov     rcx, rsp
0x11ee51:    movups  xmmword ptr [rsp + 0x44], xmm0
0x11ee56:    movdqa  xmm2, xmmword ptr [rsp + 0x40]
0x11ee5c:    mov     eax, dword ptr [rsp + 0x50]
0x11ee60:    mov     dword ptr [rsp + 0x30], r8d
0x11ee65:    movdqa  xmm1, xmmword ptr [rsp + 0x30]
0x11ee6b:    mov     dword ptr [rsp + 0x20], eax
0x11ee6f:    movaps  xmmword ptr [rsp], xmm1
0x11ee73:    movaps  xmmword ptr [rsp + 0x10], xmm2
0x11ee78:    call    0x11e9e0
0x11ee7d:    mov     rdx, qword ptr [rsp + 0x58]
0x11ee82:    sub     rdx, qword ptr fs:[0x28]
0x11ee8b:    jne     0x11ee92

```

```

0x11ee8d:    add     rsp, 0x68
0x11ee91:    ret
0x11ee92:    call   0x1049d0
0x11ee97:    nop     word ptr [rax + rax]
0x11eea0:    endbr64
0x11eea4:    sub     rsp, 0x68
0x11eea8:    mov     r8d, esi
0x11eeab:    pxor    xmm0, xmm0
0x11eeaf:    mov     rsi, rdx
0x11eeb2:    mov     rax, qword ptr fs:[0x28]
0x11eebb:    mov     qword ptr [rsp + 0x58], rax
0x11eec0:    xor     eax, eax
0x11eec2:    movups  xmmword ptr [rsp + 0x34], xmm0
0x11eec7:    mov     rdx, rcx
0x11eeca:    mov     rcx, rsp
0x11eecd:    movups  xmmword ptr [rsp + 0x44], xmm0
0x11eed2:    movdqa  xmm2, xmmword ptr [rsp + 0x40]
0x11eed8:    mov     eax, dword ptr [rsp + 0x50]
0x11eedc:    mov     dword ptr [rsp + 0x30], r8d
0x11eee1:    movdqa  xmm1, xmmword ptr [rsp + 0x30]
0x11eee7:    mov     dword ptr [rsp + 0x20], eax
0x11eeeb:    movaps  xmmword ptr [rsp], xmm1
0x11eeef:    movaps  xmmword ptr [rsp + 0x10], xmm2
0x11eef4:    call   0x11e9e0
0x11eef9:    mov     rdx, qword ptr [rsp + 0x58]
0x11eefe:    sub     rdx, qword ptr fs:[0x28]
0x11ef07:    jne     0x11ef0e
0x11ef09:    add     rsp, 0x68
0x11ef0d:    ret
0x11ef0e:    call   0x1049d0
0x11ef13:    nop     word ptr cs:[rax + rax]
0x11ef1e:    nop
0x11ef20:    endbr64
0x11ef24:    sub     rsp, 0x68
0x11ef28:    pxor    xmm0, xmm0
0x11ef2c:    mov     rdx, -1
0x11ef33:    mov     rax, qword ptr fs:[0x28]
0x11ef3c:    mov     qword ptr [rsp + 0x58], rax
0x11ef41:    xor     eax, eax
0x11ef43:    mov     dword ptr [rsp + 0x30], edi
0x11ef47:    mov     rcx, rsp
0x11ef4a:    xor     edi, edi
0x11ef4c:    movups  xmmword ptr [rsp + 0x34], xmm0
0x11ef51:    movdqa  xmm1, xmmword ptr [rsp + 0x30]
0x11ef57:    movups  xmmword ptr [rsp + 0x44], xmm0
0x11ef5c:    movdqa  xmm2, xmmword ptr [rsp + 0x40]
0x11ef62:    mov     eax, dword ptr [rsp + 0x50]
0x11ef66:    movaps  xmmword ptr [rsp], xmm1

```



```

0x11ef6a:    mov     dword ptr [rsp + 0x20], eax
0x11ef6e:    movaps  xmmword ptr [rsp + 0x10], xmm2
0x11ef73:    call    0x11e9e0
0x11ef78:    mov     rdx, qword ptr [rsp + 0x58]
0x11ef7d:    sub     rdx, qword ptr fs:[0x28]
0x11ef86:    jne     0x11ef8d
0x11ef88:    add     rsp, 0x68
0x11ef8c:    ret
0x11ef8d:    call    0x1049d0
0x11ef92:    nop     word ptr cs:[rax + rax]
0x11ef9d:    nop     dword ptr [rax]
0x11efa0:    endbr64
0x11efa4:    sub     rsp, 0x38
0x11efa8:    movdqa  xmm0, xmmword ptr [rip + 0xccb0]
0x11efb0:    movdqa  xmm1, xmmword ptr [rip + 0xccb8]
0x11efb8:    mov     ecx, esi
0x11efba:    mov     rax, qword ptr fs:[0x28]
0x11efc3:    mov     qword ptr [rsp + 0x28], rax
0x11efc8:    xor     eax, eax
0x11efca:    mov     eax, dword ptr [rip + 0xccb0]
0x11efd0:    and     ecx, 0x1f
0x11efd3:    mov     r9, rsp
0x11efd6:    movaps  xmmword ptr [rsp], xmm0
0x11efda:    mov     dword ptr [rsp + 0x20], eax
0x11efde:    mov     eax, esi
0x11efe0:    shr     al, 5
0x11efe3:    movaps  xmmword ptr [rsp + 0x10], xmm1
0x11efe8:    movzx   eax, al
0x11efeb:    lea     rdx, [rsp + rax*4 + 4]
0x11eff0:    mov     esi, dword ptr [rdx]
0x11eff2:    mov     eax, esi
0x11eff4:    shr     eax, cl
0x11eff6:    not     eax
0x11eff8:    and     eax, 1
0x11effb:    shl     eax, cl
0x11effd:    mov     rcx, r9
0x11f000:    xor     eax, esi
0x11f002:    mov     rsi, rdi
0x11f005:    xor     edi, edi
0x11f007:    mov     dword ptr [rdx], eax
0x11f009:    mov     rdx, -1
0x11f010:    call    0x11e9e0
0x11f015:    mov     rdx, qword ptr [rsp + 0x28]
0x11f01a:    sub     rdx, qword ptr fs:[0x28]
0x11f023:    jne     0x11f02a
0x11f025:    add     rsp, 0x38
0x11f029:    ret
0x11f02a:    call    0x1049d0

```

```
0x11f02f:    nop
0x11f030:    endbr64
0x11f034:    mov     esi, 0x3a
0x11f039:    jmp     0x11efa0
0x11f040:    endbr64
0x11f044:    sub     rsp, 8
0x11f048:    add     rsp, 8
0x11f04c:    ret
```

[]: