a[0] = 1

a[6] 0, 1, 2, 3, 4, 5
(a, b, c, d, e, f)
1 2 4 7 28 33

genap : a[i] = (i-1) + a[i-1]

ganjil : a[i] = (i-1) * a[i-1]

a[2] = 1 + a[1]

```
0x000016fd <+38>:    push   0x3c(%esp)
0x00001701 <+42>:    call   0x2236 <read_six_numbers>
0x00001706 <+47>:    add    $0x10,%esp
0x00001709 <+50>:    cmpl   $0x1,0x4(%esp)
0x0000170e <+55>:    jne    0x171b <Holoro+68>
0x00001710 <+57>:    mov    $0x1,%esi
0x00001715 <+62>:    lea    0x4(%esp),%edi
0x00001719 <+66>:    jmp    0x173b <Holoro+100>
0x0000171b <+68>:    call   0x214e <illegal_move>
0x00001720 <+73>:    jmp    0x1710 <Holoro+57>
0x00001722 <+75>:    mov    %esi,%eax
0x00001724 <+77>:    imul   -0x4(%edi,%esi,4),%eax
0x00001729 <+82>:    cmp    %eax,(%edi,%esi,4)
0x0000172c <+85>:    je     0x1733 <Holoro+92>
0x0000172e <+87>:    call   0x214e <illegal_move>
0x00001733 <+92>:    add    $0x1,%esi
0x00001736 <+95>:    cmp    $0x6,%esi
0x00001739 <+98>:    je     0x1750 <Holoro+121>
0x0000173b <+100>:   test   $0x1,%esi
0x00001741 <+106>:   je     0x1722 <Holoro+75>
0x00001743 <+108>:   mov    %esi,%eax
0x00001745 <+110>:   add    -0x4(%edi,%esi,4),%eax
0x00001749 <+114>:   cmp    %eax,(%edi,%esi,4)
0x0000174c <+117>:   jne    0x172e <Holoro+87>
0x0000174e <+119>:   jmp    0x1733 <Holoro+92>
0x00001750 <+121>:   mov    0x1c(%esp),%eax
0x00001754 <+125>:   xor    %gs:0x14,%eax
0x0000175b <+132>:   jne    0x1764 <Holoro+141>
0x0000175d <+134>:   add    $0x20,%esp
0x00001760 <+137>:   pop    %ebx
```

Handwritten annotations beside the listing:

- → compare a : 1
- → esi = 1
- → edi = alamat a
- → eax = 2
- → eax = 2 * (2) = 4
- → c : 4
- → esi = 2
- → compare 2 : 6
- → test 1 : 1 = 1
- → eax = 1
- → eax = 1 + 1
- → compare b : 2

→ esi : 2
→ compare 2 : 6
je X
→ test 2 : 1 = 0
+75 →

<+92> esi = 3
compare 3 : 6
test 3 : 1 = 1
<+108> eax = 3
eax = 3 + (alamat c)
= 3 + 4
= 7
compare d : 7

<+92> = esi = 4
compare 4 : 6
test 4 : 1 = 0
<+75> eax = 4
eax = 4 * (alamat d)
= 4 * 7
= 28
compare e : 28

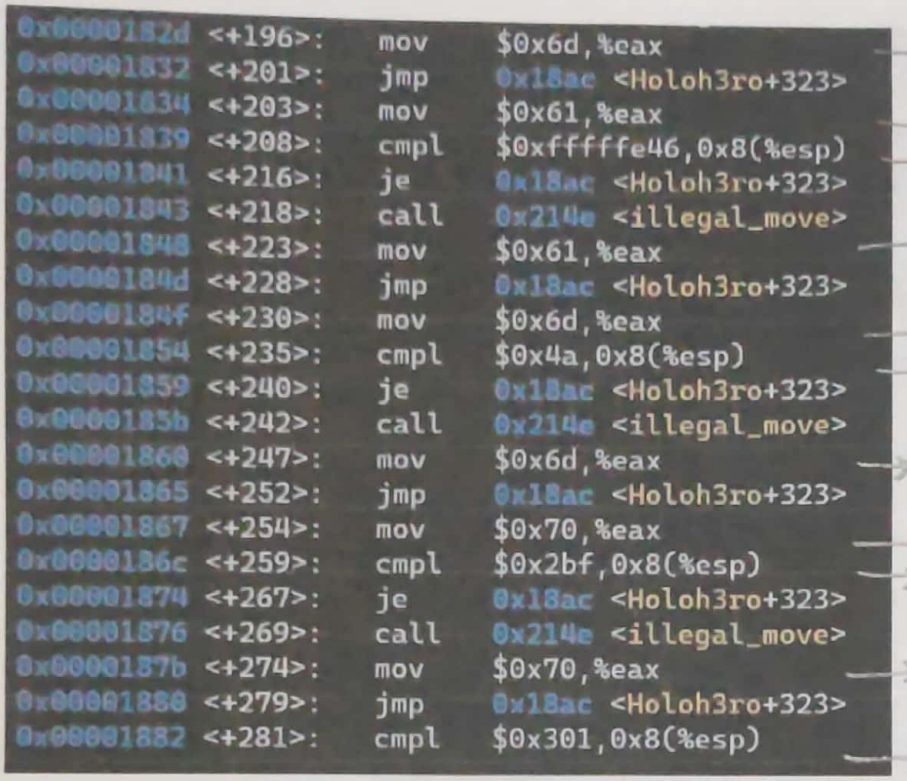
<+92> esi = 5
com 5 : 6
test 5 : 1 = 1
eax = 5

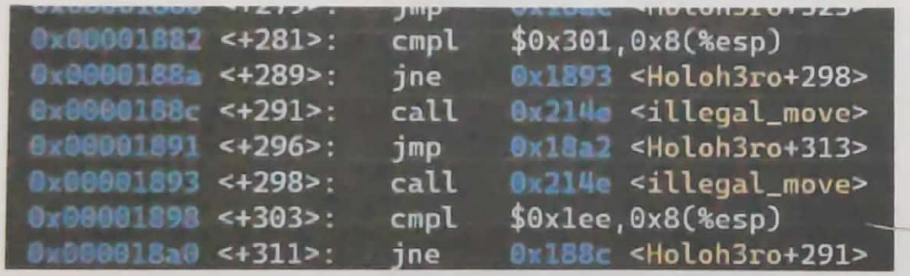
<+110> eax = 5 + 28
= 33
compare f : 33

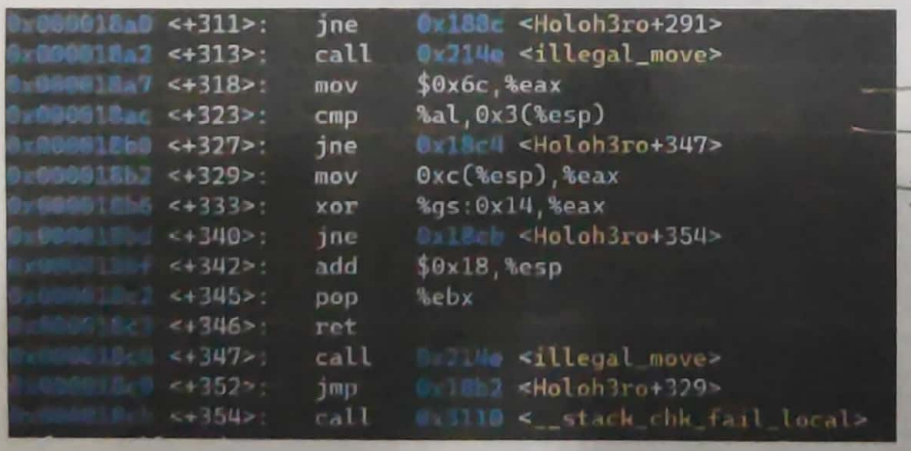
esi = 6
compare 6 : 6
<+121> eax =

```
mp of assembler code for function Holoh3ro:
0x00001769 <+0>:     endbr32
0x0000176d <+4>:     push   %ebx
0x0000176e <+5>:     sub    $0x24,%esp
0x00001771 <+8>:     call   0x13f0 <__x86.get_pc_thunk.bx>
0x00001776 <+13>:    add    $0x57ea,%ebx
0x0000177c <+19>:    mov    %gs:0x14,%eax
0x00001782 <+25>:    mov    %eax,0x18(%esp)
0x00001786 <+29>:    xor    %eax,%eax
0x00001788 <+31>:    lea    0x14(%esp),%eax
0x0000178c <+35>:    push   %eax
0x0000178d <+36>:    lea    0x13(%esp),%eax
0x00001791 <+40>:    push   %eax
0x00001792 <+41>:    lea    0x18(%esp),%eax
0x00001796 <+45>:    push   %eax
0x00001797 <+46>:    lea    -0x2c28(%ebx),%eax
0x0000179d <+52>:    push   %eax
0x0000179e <+53>:    push   0x3c(%esp)
0x000017a2 <+57>:    call   0x12e0 <__isoc99_sscanf@plt>
0x000017a7 <+62>:    add    $0x20,%esp
0x000017aa <+65>:    cmp    $0x2,%eax
0x000017ad <+68>:    jle    0x17ca <Holoh3ro+97>
0x000017af <+70>:    cmpl   $0x7,0x4(%esp)
0x000017b4 <+75>:    ja     0x18a2 <Holoh3ro+313>
0x000017ba <+81>:    mov    0x4(%esp),%eax
0x000017be <+85>:    mov    %ebx,%edx
0x000017c0 <+87>:    add    -0x2c00(%ebx,%eax,4),%edx
0x000017c7 <+94>:    notrack jmp *%edx
0x000017ca <+97>:    call   0x214e <illegal_move>
0x000017cf <+102>:   jmp    0x17af <Holoh3ro+70>
0x000017d1 <+104>:   mov    $0x61,%eax
0x000017d6 <+109>:   cmpl   $0x2ac,0x8(%esp)
0x000017de <+117>:   je     0x18ac <Holoh3ro+323>
0x000017e4 <+123>:   call   0x214e <illegal_move>
```

```
0x000017e4 <+123>:   call   0x214e <illegal_move>
0x000017e9 <+128>:   mov    $0x61,%eax
0x000017ee <+133>:   jmp    0x18ac <Holoh3ro+323>
0x000017f3 <+138>:   mov    $0x73,%eax
0x000017f8 <+143>:   cmpl   $0x3b3,0x8(%esp)
0x00001800 <+151>:   je     0x18ac <Holoh3ro+323>
0x00001806 <+157>:   call   0x214e <illegal_move>
0x0000180b <+162>:   mov    $0x73,%eax
0x00001810 <+167>:   jmp    0x18ac <Holoh3ro+323>
0x00001815 <+172>:   mov    $0x6d,%eax
0x0000181a <+177>:   cmpl   $0xd2,0x8(%esp)
0x00001822 <+185>:   je     0x18ac <Holoh3ro+323>
0x00001828 <+191>:   call   0x214e <illegal_move>
0x0000182d <+196>:   mov    $0x6d,%eax
```







c

0x634d2d9 0x00000001

bc

d

100 → 73

's'

```
Dump of assembler code for function Secret_Society:
   0x00001914 <+0>:     endbr32
   0x00001918 <+4>:     push   %edi
   0x00001919 <+5>:     push   %esi
   0x0000191a <+6>:     push   %ebx
   0x0000191b <+7>:     sub    $0x10,%esp
   0x0000191e <+10>:    call   0x1dce <__x86.get_pc_thunk.si>
   0x00001923 <+15>:    add    $0x563d,%esi
   0x00001929 <+21>:    mov    %gs:0x14,%eax
   0x0000192f <+27>:    mov    %eax,0xc(%esp)
   0x00001933 <+31>:    xor    %eax,%eax
   0x00001935 <+33>:    lea    0x8(%esp),%eax
   0x00001939 <+37>:    push   %eax
   0x0000193a <+38>:    lea    0x8(%esp),%eax
   0x0000193e <+42>:    push   %eax
   0x0000193f <+43>:    lea    -0x249d(%esi),%eax
   0x00001945 <+49>:    push   %eax
   0x00001946 <+50>:    push   0x2c(%esp)
   0x0000194a <+54>:    mov    %esi,%ebx
   0x0000194c <+56>:    call   0x12e0 <__isoc99_sscanf@plt>
   0x00001951 <+61>:    add    $0x10,%esp
   0x00001954 <+64>:    cmp    $0x2,%eax
   0x00001957 <+67>:    jne    0x1971 <Secret_Society+93>
   0x00001959 <+69>:    mov    0x4(%esp),%eax
   0x0000195d <+73>:    sub    $0x2,%eax
   0x00001960 <+76>:    cmp    $0x15,%eax
   0x00001963 <+79>:    ja     0x1971 <Secret_Society+93>
   0x00001965 <+81>:    mov    0x8(%esp),%eax
   0x00001969 <+85>:    sub    $0x2,%eax
   0x0000196c <+88>:    cmp    $0x15,%eax
   0x0000196f <+91>:    jbe    0x1978 <Secret_Society+100>
   0x00001971 <+93>:    mov    %esi,%ebx
   0x00001973 <+95>:    call   0x214e <illegal_move>
   0x00001978 <+100>:   mov    0x8(%esp),%ebx
   0x0000197c <+104>:   mov    0x4(%esp),%edi
```

```
   0x0000197c <+104>:   mov    0x4(%esp),%edi
   0x00001980 <+108>:   sub    $0x8,%esp
   0x00001983 <+111>:   push   %ebx
   0x00001984 <+112>:   push   %edi
   0x00001985 <+113>:   call   0x18d0 <func4>
   0x0000198a <+118>:   add    $0x10,%esp
   0x0000198d <+121>:   mov    %eax,%ecx
   0x0000198f <+123>:   mov    %ebx,%eax
   0x00001991 <+125>:   imul   %edi,%eax
   0x00001994 <+128>:   cltd
   0x00001995 <+129>:   idiv   %ecx
   0x00001997 <+131>:   cmp    $0x18,%eax
   [illegible] <+134>:  jne    [illegible] <Secret_Society+156>
   [illegible] <+136>:  mov    0xc(%esp),%eax
   [illegible] <+140>:  xor    %gs:0x14,%eax
   [illegible] <+147>:  jne    [illegible] <Secret_Society+165>
   [illegible] <+149>:  add    $0x10,%esp
   [illegible] <+152>:  pop    %ebx
   [illegible] <+153>:  pop    %esi
   [illegible] <+154>:  pop    %edi
   [illegible] <+155>:  ret
   [illegible] <+156>:  mov    %esi,%ebx
   [illegible] <+158>:  call   [illegible] <illegal_move>
   [illegible] <+163>:  jmp    [illegible] <Secret_Society+136>
   [illegible] <+165>:  call   [illegible] <__stack_chk_fail_local>
```

edi
esi
ebx

b 8
a 4
esp

→ "%d %d"

→ compr eax : 2
≠
→ eax = a → 23
→ eax = a - 2 → 21
compr a - 2 ≤ 21 $a \leq 23$
21 : 21

→ eax = b
→ b - 2 $b \leq 23$
→

→ ebx = b
→ edi = a
→
→ esp - 8
→ ecx = eax (hasil gcd)
→ eax = ebx (a)
→ eax = eax · edi (
a · a · b

→ $\frac{eax}{ecx} = 24 = \frac{a \cdot b}{gcd(a,b)}$

1 24 $a \leq 23$
2 12 $b \leq 23$
3 8   a = 3
4 6   b = 8
6 4

gcd(a,b) = gcd(b, a%b)

gcd(9,2)

gcd(7,0)

```
Dump of assembler code for function func4:
   0x000018d0 <+0>:     endbr32
   0x000018d4 <+4>:     sub    $0xc,%esp
   0x000018d7 <+7>:     mov    0x10(%esp),%ecx
   0x000018db <+11>:    mov    0x14(%esp),%eax
   0x000018df <+15>:    cmp    %eax,%ecx
   0x000018e1 <+17>:    jg     0x18ef <func4+31>
   0x000018e3 <+19>:    mov    %eax,%edx
   0x000018e5 <+21>:    test   %ecx,%ecx
   0x000018e7 <+23>:    jne    0x1900 <func4+48>
   0x000018e9 <+25>:    mov    %edx,%eax
   0x000018eb <+27>:    add    $0xc,%esp
   0x000018ee <+30>:    ret
   0x000018ef <+31>:    sub    $0x8,%esp
   0x000018f2 <+34>:    push   %ecx
   0x000018f3 <+35>:    push   %eax
   0x000018f4 <+36>:    call   0x18d0 <func4>
   0x000018f9 <+41>:    add    $0x10,%esp
   0x000018fc <+44>:    mov    %eax,%edx
   0x000018fe <+46>:    jmp    0x18e9 <func4+25>
   0x00001900 <+48>:    sub    $0x8,%esp
   0x00001903 <+51>:    push   %ecx
   0x00001904 <+52>:    cltd
   0x00001905 <+53>:    idiv   %ecx
   0x00001907 <+55>:    push   %edx
   0x00001908 <+56>:    call   0x18d0 <func4>
   0x0000190d <+61>:    add    $0x10,%esp
   0x00001910 <+64>:    mov    %eax,%edx
   0x00001912 <+66>:    jmp    0x18e9 <func4+25>
```

→ a
→ b
→ Compare ecx : eax
a : b
→ edx = b
→ test ecx : ecx
a a
kalo 0 return
→ eax = b
→ gcd
→ a
→ a div b
eax ↓ 5 div 2 = 2
edx = 1
→ gcd(2,1)

gcd(7,2)
5

idiv return ⇒ eax (hasil bagi)
↳ edx (sisa bagi)

[(12), (10), (4), (1), (13), (15), (2), (11), (9), (6), (7), (5), (14), (0), (8), (3)]

16 elemen

```
0x000019be <+0>:    endbr32
0x000019c2 <+4>:    push   %esi
0x000019c3 <+5>:    push   %ebx
0x000019c4 <+6>:    sub    $0x14,%esp
0x000019c7 <+9>:    call   0x13f0 <__x86.get_pc_thunk.bx>
0x000019cc <+14>:   add    $0x5594,%ebx
0x000019d2 <+20>:   mov    %gs:0x14,%eax
0x000019d8 <+26>:   mov    %eax,0xc(%esp)
0x000019dc <+30>:   xor    %eax,%eax
0x000019de <+32>:   lea    0x8(%esp),%eax
0x000019e2 <+36>:   push   %eax
0x000019e3 <+37>:   lea    0x8(%esp),%eax
0x000019e7 <+41>:   push   %eax
0x000019e8 <+42>:   lea    -0x249d(%ebx),%eax
0x000019ee <+48>:   push   %eax
0x000019ef <+49>:   push   0x2c(%esp)
0x000019f3 <+53>:   call   0x12e0 <__isoc99_sscanf@plt>
0x000019f8 <+58>:   add    $0x10,%esp
0x000019fb <+61>:   cmp    $0x1,%eax
0x000019fe <+64>:   jle    0x1a58 <Area15+154>
0x00001a00 <+66>:   mov    0x4(%esp),%eax
0x00001a04 <+70>:   and    $0xf,%eax
0x00001a07 <+73>:   mov    %eax,0x4(%esp)
0x00001a0b <+77>:   cmp    $0xf,%eax
0x00001a0e <+80>:   je     0x1a40 <Area15+130>
0x00001a10 <+82>:   mov    $0x0,%ecx
0x00001a15 <+87>:   mov    $0x0,%edx
0x00001a1a <+92>:   lea    -0x2be0(%ebx),%esi
0x00001a20 <+98>:   add    $0x1,%edx
0x00001a23 <+101>:  mov    (%esi,%eax,4),%eax
0x00001a26 <+104>:  add    %eax,%ecx
0x00001a28 <+106>:  cmp    $0xf,%eax
0x00001a2b <+109>:  jne    0x1a20 <Area15+98>
0x00001a2d <+111>:  movl   $0xf,0x4(%esp)
0x00001a35 <+119>:  cmp    $0xf,%edx
0x00001a38 <+122>:  jne    0x1a40 <Area15+130>
```

```
0x00001a38 <+122>:  jne    0x1a40 <Area15+130>
0x00001a3a <+124>:  cmp    %ecx,0x8(%esp)
0x00001a3e <+128>:  je     0x1a45 <Area15+135>
0x00001a40 <+130>:  call   0x214e <illegal_move>
0x00001a45 <+135>:  mov    0xc(%esp),%eax
0x00001a49 <+139>:  xor    %gs:0x14,%eax
0x00001a50 <+146>:  jne    0x1a5f <Area15+161>
0x00001a52 <+148>:  add    $0x14,%esp
0x00001a55 <+151>:  pop    %ebx
0x00001a56 <+152>:  pop    %esi
0x00001a57 <+153>:  ret
0x00001a58 <+154>:  call   0x214e <illegal_move>
0x00001a5d <+159>:  jmp    0x1a00 <Area15+66>
0x00001a5f <+161>:  call   0x1110 <__stack_chk_fail_local>
```

Handwritten notes (right):

"%d %d"
a b

eax = a
eax: a and 0xf (a%16)
1111
a = (a%16)
Compare (a%16) : 15
gabolih kelipatan 15
ecx = 0
edx = 0
esi = ebx - 0x2be0
edx = 0+1 = 1
eax = 0x56559380 + 4 * 2
ecx = 0x56559388
Compare 0x56559388 : 15
edx = 2
eax = 0x56559380 + 4 * 0x56559388
ecx = 0x56559388 +

Handwritten notes (left):

2 180
a = (a%15)
ecx = 0
edx = 0
esi =
edx: 1
eax = 4
ecx = 4

3 147
eax: 3
esp+4: 3
ecx: 0
edx = 0
esi: ...
edx = 1
eax: esi + 12 = 2

edx = 2, eax = esi + 4

esi = 0x56559380

Jawaban : 3 (120-3)
3 117

a b c d e f
1 2 3 4 5 6

```
Dump of assembler code for function Green_Necromancer:
   0x00001a64 <+0>:     endbr32
   0x00001a68 <+4>:     push   %ebp
   0x00001a69 <+5>:     push   %edi
   0x00001a6a <+6>:     push   %esi
   0x00001a6b <+7>:     push   %ebx
   0x00001a6c <+8>:     sub    $0x64,%esp
   0x00001a6f <+11>:    call   0x13f0 <__x86.get_pc_thunk.bx>
   0x00001a74 <+16>:    add    $0x54ec,%ebx
   0x00001a7a <+22>:    mov    %gs:0x14,%eax
   0x00001a80 <+28>:    mov    %eax,0x54(%esp)
   0x00001a84 <+32>:    xor    %eax,%eax
   0x00001a86 <+34>:    lea    0x24(%esp),%eax
   0x00001a8a <+38>:    push   %eax
   0x00001a8b <+39>:    push   0x7c(%esp)
   0x00001a8f <+43>:    call   0x2236 <read_six_numbers>
   0x00001a94 <+48>:    lea    0x30(%esp),%ebp
   0x00001a98 <+52>:    add    $0x10,%esp
   0x00001a9b <+55>:    movl   $0x0,0xc(%esp)
   0x00001aa3 <+63>:    lea    0x34(%esp),%eax
   0x00001aa7 <+67>:    mov    %eax,0x8(%esp)
   0x00001aab <+71>:    jmp    0x1ace <Green_Necromancer+106>
   0x00001aad <+73>:    call   0x214e <illegal_move>
   0x00001ab2 <+78>:    jmp    0x1adb <Green_Necromancer+119>
   0x00001ab4 <+80>:    add    $0x4,%esi
   0x00001ab7 <+83>:    cmp    %esi,0x8(%esp)
   0x00001abb <+87>:    je     0x1acb <Green_Necromancer+103>
   0x00001abd <+89>:    mov    (%esi),%eax
   0x00001abf <+91>:    cmp    %eax,-0x4(%edi)
   0x00001ac2 <+94>:    jne    0x1ab4 <Green_Necromancer+80>
   0x00001ac4 <+96>:    call   0x214e <illegal_move>
   0x00001ac9 <+101>:   jmp    0x1ab4 <Green_Necromancer+80>
   0x00001acb <+103>:   add    $0x4,%ebp
   0x00001ace <+106>:   mov    %ebp,%edi
```

```
   0x00001ad0 <+108>:   mov    -0x4(%ebp),%eax
   0x00001ad3 <+111>:   sub    $0x1,%eax
   0x00001ad6 <+114>:   cmp    $0x5,%eax
   0x00001ad9 <+117>:   ja     0x1aad <Green_Necromancer+73>
   0x00001adb <+119>:   addl   $0x1,0xc(%esp)
   0x00001ae0 <+124>:   mov    0xc(%esp),%eax
   0x00001ae4 <+128>:   cmp    $0x5,%eax
   0x00001ae7 <+131>:   jg     0x1aed <Green_Necromancer+137>
   0x00001ae9 <+133>:   mov    %ebp,%esi
   0x00001aeb <+135>:   jmp    0x1abd <Green_Necromancer+89>
   0x00001aed <+137>:   mov    $0x0,%esi
   0x00001af2 <+142>:   mov    %esi,%edi
   0x00001af4 <+144>:   mov    0x1c(%esp,%esi,4),%ecx
   0x00001af8 <+148>:   mov    $0x1,%eax
   0x00001afd <+153>:   lea    0x598(%ebx),%edx
   0x00001b03 <+159>:   cmp    $0x1,%ecx
   0x00001b06 <+162>:   jle    0x1b12 <Green_Necromancer+174>
   0x00001b08 <+164>:   mov    0x8(%edx),%edx
```

→ ebp = alamat b
→ set = 0
→ eax = alamat f + 4 (g)
→ esp8 = alamat f+g
esi = alamat c
compare alamat g : alamat
→ eax = 2
→ compare 1 : 2
→ edi = alamat b
→ eax = 1
→ eax = 0
→ compare 0 : 5
→ set = 1
→ eax = 1
→ compare 1 : 5
→ esi = alamat b
→ esi = 0
→ edi = 0
→ ecx = esp + 0 + 28 (1) → 4
→ eax = 1
→ edx = ebx + 0x598
0x565556160 (head)
edx = 0x5655c978
→ compare ecx : 1
2 : 1

edi : 1
ecx : 2
eax = 1
edx = head
compare ecx : 1
2 : 1

esi : 0
edi : 0

ecx = 2

```
0x00001b08 <+164>:   mov    0x8(%edx),%edx
0x00001b0b <+167>:   add    $0x1,%eax
0x00001b0e <+170>:   cmp    %ecx,%eax
0x00001b10 <+172>:   jne    0x1b08 <Green_Necromancer+164>
0x00001b12 <+174>:   mov    %edx,0x34(%esp,%edi,4)
0x00001b16 <+178>:   add    $0x1,%esi
0x00001b19 <+181>:   cmp    $0x6,%esi
0x00001b1c <+184>:   jne    0x1af2 <Green_Necromancer+142>
0x00001b1e <+186>:   mov    0x34(%esp),%esi
0x00001b22 <+190>:   mov    0x38(%esp),%eax
0x00001b26 <+194>:   mov    %eax,0x8(%esi)
0x00001b29 <+197>:   mov    0x3c(%esp),%edx
0x00001b2d <+201>:   mov    %edx,0x8(%eax)
0x00001b30 <+204>:   mov    0x40(%esp),%eax
0x00001b34 <+208>:   mov    %eax,0x8(%edx)
0x00001b37 <+211>:   mov    0x44(%esp),%edx
```

edx = value (edx+8)
eax = 2
compare 2 : 2
→ g = head
→ esi = 1
→ compare 1 : 6
→ 0x5655c4f8
→ 0x5655c509
→ 8esi = 0x5655c509
→ edx =

```
0x00001b37 <+211>:   mov    0x44(%esp),%edx
0x00001b3b <+215>:   mov    %edx,0x8(%eax)
0x00001b3e <+218>:   mov    0x48(%esp),%eax
0x00001b42 <+222>:   mov    %eax,0x8(%edx)
0x00001b45 <+225>:   movl   $0x0,0x8(%eax)
0x00001b4c <+232>:   mov    $0x5,%edi
0x00001b51 <+237>:   jmp    0x1b5b <Green_Necromancer+247>
0x00001b53 <+239>:   mov    0x8(%esi),%esi
0x00001b56 <+242>:   sub    $0x1,%edi
0x00001b59 <+245>:   je     0x1b6b <Green_Necromancer+263>
0x00001b5b <+247>:   mov    0x8(%esi),%eax
0x00001b5e <+250>:   mov    (%eax),%eax
0x00001b60 <+252>:   cmp    %eax,(%esi)
0x00001b62 <+254>:   jle    0x1b53 <Green_Necromancer+239>
0x00001b64 <+256>:   call   0x214e <illegal_move>
0x00001b69 <+261>:   jmp    0x1b53 <Green_Necromancer+239>
0x00001b6b <+263>:   mov    0x4c(%esp),%eax
0x00001b6f <+267>:   xor    %gs:0x14,%eax
0x00001b76 <+274>:   jne    0x1b80 <Green_Necromancer+284>
0x00001b78 <+276>:   add    $0x5c,%esp
0x00001b7b <+279>:   pop    %ebx
0x00001b7c <+280>:   pop    %esi
0x00001b7d <+281>:   pop    %edi
0x00001b7e <+282>:   pop    %ebp
0x00001b7f <+283>:   ret
0x00001b80 <+284>:   call   0x3116 <__stack_chk_fail_local>
```

→ eax = 0x5655c509
→ eax = 326
→ Compare 326 : 336

Compare 517 : 960

edx 0x5655c4f8

1. 0x000001a6 = 326
2. alamat : 0x5655c509, val : 366
3. 0x5655c510, val : 321
4. 0x5655c51c, val : 517
5. 0x5655c528, val : 960
6. 0x5655c030, val : 479

3, 1, 2, 6, 4, 5

test :

Compare 321 : 326
Compare 326 : 366
Compare 366 : 479
Compare 479 : 517

```
Dump of assembler code for function Tanigox_secret_lair:
   0x00001bda <+0>:     endbr32
   0x00001bde <+4>:     push   %esi
   0x00001bdf <+5>:     push   %ebx
   0x00001be0 <+6>:     sub    $0x4,%esp
   0x00001be3 <+9>:     call   0x13f0 <__x86.get_pc_thunk.bx>
   0x00001be8 <+14>:    add    $0x5378,%ebx
   0x00001bee <+20>:    call   0x2284 <read_line>
   0x00001bf3 <+25>:    sub    $0x4,%esp
   0x00001bf6 <+28>:    push   $0xa
   0x00001bf8 <+30>:    push   $0x0
   0x00001bfa <+32>:    push   %eax
   0x00001bfb <+33>:    call   0x1360 <strtol@plt>
   0x00001c00 <+38>:    mov    %eax,%esi
   0x00001c02 <+40>:    lea    -0x67(%eax),%eax
   0x00001c05 <+43>:    add    $0x10,%esp
   0x00001c08 <+46>:    cmp    $0x7eb,%eax
   0x00001c0d <+51>:    ja     0x1dba <Tanigox_secret_lair+480>
   0x00001c13 <+57>:    sub    $0x8,%esp
   0x00001c16 <+60>:    push   %esi
   0x00001c17 <+61>:    lea    0x544(%ebx),%eax
   0x00001c1d <+67>:    push   %eax
   0x00001c1e <+68>:    call   0x1b85 <fun7>
   0x00001c23 <+73>:    add    $0x10,%esp
   0x00001c26 <+76>:    cmp    $0x5,%eax
   0x00001c29 <+79>:    jne    0x1dc4 <Tanigox_secret_lair+490>
   0x00001c2f <+85>:    sub    $0xc,%esp
   0x00001c32 <+88>:    lea    -0x2ba0(%ebx),%eax
   0x00001c38 <+94>:    push   %eax
   0x00001c39 <+95>:    call   0x1290 <puts@plt>
   0x00001c3e <+100>:   add    $0x8,%esp
   0x00001c41 <+103>:   lea    -0x2b74(%ebx),%eax
   0x00001c47 <+109>:   push   %eax
   0x00001c48 <+110>:   push   $0x1
   0x00001c4a <+112>:   call   0x1320 <__printf_chk@plt>
   0x00001c4f <+117>:   add    $0x8,%esp
```

```
Dump of assembler code for function fun7:
   0x56556b85 <+0>:     endbr32
   0x56556b89 <+4>:     push   %ebx
   0x56556b8a <+5>:     sub    $0x8,%esp
   0x56556b8d <+8>:     mov    0x10(%esp),%edx
   0x56556b91 <+12>:    mov    0x14(%esp),%ecx
   0x56556b95 <+16>:    test   %edx,%edx
   0x56556b97 <+18>:    je     0x56556bd3 <fun7+78>
   0x56556b99 <+20>:    mov    (%edx),%ebx
   0x56556b9b <+22>:    cmp    %ecx,%ebx
   0x56556b9d <+24>:    jg     0x56556bab <fun7+38>
   0x56556b9f <+26>:    mov    $0x0,%eax
   0x56556ba4 <+31>:    jne    0x56556bbe <fun7+57>
   0x56556ba6 <+33>:    add    $0x8,%esp
   0x56556ba9 <+36>:    pop    %ebx
   0x56556baa <+37>:    ret
   0x56556bab <+38>:    sub    $0x8,%esp
   0x56556bae <+41>:    push   %ecx
   0x56556baf <+42>:    push   0x4(%edx)
   0x56556bb2 <+45>:    call   0x56556b85 <fun7>
   0x56556bb7 <+50>:    add    $0x10,%esp
   0x56556bba <+53>:    add    %eax,%eax
   0x56556bbc <+55>:    jmp    0x56556ba6 <fun7+33>
   0x56556bbe <+57>:    sub    $0x8,%esp
   0x56556bc1 <+60>:    push   %ecx
   0x56556bc2 <+61>:    push   0x8(%edx)
   0x56556bc5 <+64>:    call   0x56556b85 <fun7>
   0x56556bca <+69>:    add    $0x10,%esp
   0x56556bcd <+72>:    lea    0x1(%eax,%eax,1),%eax
   0x56556bd1 <+76>:    jmp    0x56556ba6 <fun7+33>
   0x56556bd3 <+78>:    mov    $0xffffffff,%eax
   0x56556bd8 <+83>:    jmp    0x56556ba6 <fun7+33>
End of assembler dump.
```