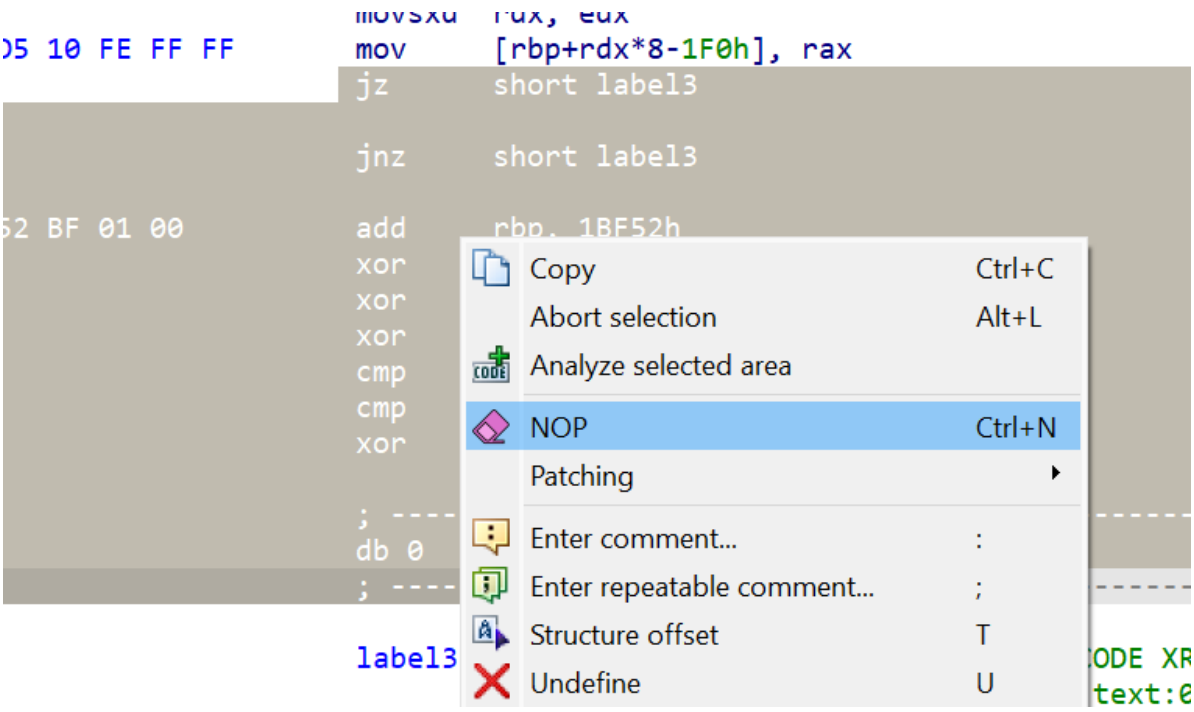
IDA一打开连main函数都没了。

```
.text:000000000000122C                          ; int __cdecl main(int argc, const char **argv, const char **envp)
.text:000000000000122C                          public main
.text:000000000000122C                          main:                                   ; DATA XREF: _start+14↑o
.text:000000000000122C                          ; __unwind {
.text:000000000000122C 55                       push    rbp
.text:000000000000122D 48 89 E5                 mov     rbp, rsp
.text:0000000000001230 48 81 EC F0 01 00 00     sub     rsp, 1F0h
.text:0000000000001237 48 8D 05 4A 0E 00 00     lea     rax, aWatashinoRsaWo            ; "Watashino RSA-wo mitekudasai!'
.text:000000000000123E 48 89 C7                 mov     rdi, rax
.text:0000000000001241 E8 EA FD FF FF           call    _puts
.text:0000000000001241
.text:0000000000001246 48 8D 85 F0 FE FF FF     lea     rax, [rbp-110h]
.text:000000000000124D 48 89 C6                 mov     rsi, rax
.text:0000000000001250 48 8D 05 4F 0E 00 00     lea     rax, aS                         ; "%s"
.text:0000000000001257 48 89 C7                 mov     rdi, rax
.text:000000000000125A B8 00 00 00 00           mov     eax, 0
.text:000000000000125F E8 DC FD FF FF           call    ___isoc99_scanf
.text:000000000000125F
.text:0000000000001264 53                       push    rbx
```

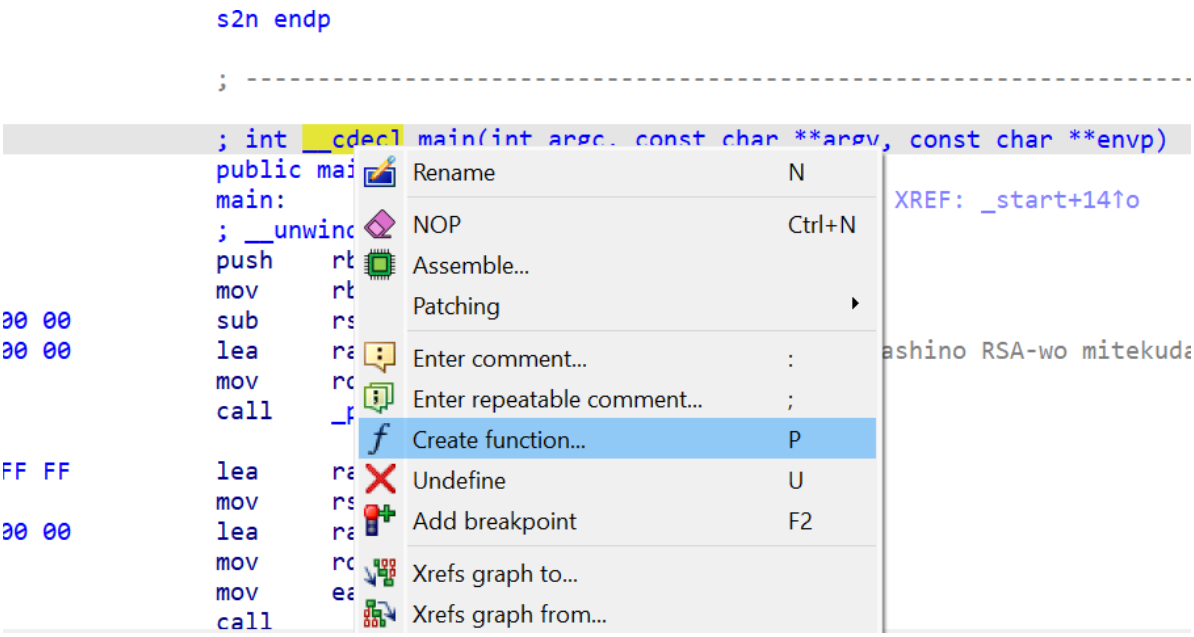
往下翻，有几个地方是必然发生的跳转指令，中间是一坨花指令。全部nop掉。

```
.text:0000000000001264 53                       push    rbx
.text:0000000000001265 48 31 DB                 xor     rbx, rbx
.text:0000000000001268 48 85 DB                 test    rbx, rbx
.text:000000000000126B 75 02                    jnz     short labela
.text:000000000000126B
.text:000000000000126D 74 27                    jz      short labelb
.text:000000000000126D
.text:000000000000126F
.text:000000000000126F                          labela:                                 ; CODE XREF: .text:000000000000126B↑j
.text:000000000000126F 2B 24 25 29 05 53 14     sub     esp, 14530529h
.text:0000000000001276 03 2C 25 11 11 44 14     add     ebp, ds:14441111h
.text:000000000000127D 68 61 68 61 71           push    71616861h
.text:0000000000001282 77 65                    ja      short near ptr loc_12E7+2
.text:0000000000001282
.text:0000000000001284 77 61                    ja      short loc_12E7
.text:0000000000001284
.text:0000000000001284                          ; ---------------------------------------------------------------------------
.text:0000000000001286 64 61 68 6E 64 6F 77 61 6A 6F+aDahndowajojoiw db 'dahndowajojoiwd',0
.text:0000000000001296                          ; ---------------------------------------------------------------------------
.text:0000000000001296
.text:0000000000001296                          labelb:                                 ; CODE XREF: .text:000000000000126D↑j
.text:0000000000001296 5B                       pop     rbx
```

```
.text:00000000000012F5 48 89 84 D5 80 FE FF FF      mov     [rbp+rdx*8-180h], rax
.text:00000000000012FD 74 2C                        jz      short label1
.text:00000000000012FD
.text:00000000000012FF 75 2A                        jnz     short label1
.text:00000000000012FF
.text:0000000000001301 6A 68                        push    68h ; 'h'
.text:0000000000001303 68 2F 2F 2F 73               push    732F2F2Fh
.text:0000000000001308 68 2F 62 69 6E               push    6E69622Fh
.text:000000000000130D 89 E3                        mov     ebx, esp
.text:000000000000130F 68 01 01 01 01               push    1010101h
.text:0000000000001314 14 24                        adc     al, 24h ; '$'
.text:0000000000001316 72 69                        jb      short loc_1381
.text:0000000000001316
.text:0000000000001318 01 11                        add     [rcx], edx
.text:000000000000131A C9                           leave
.text:000000000000131B 51                           push    rcx
.text:000000000000131C 6A 04                        push    4
.text:000000000000131E 59                           pop     rcx
.text:000000000000131F 01 E1                        add     ecx, esp
.text:0000000000001321 51                           push    rcx
.text:0000000000001322 89 11                        mov     [rcx], edx
.text:0000000000001324 D2 6A 0B                     shr     byte ptr [rdx+0Bh], cl
.text:0000000000001327 58                           pop     rax
.text:0000000000001328 CD 80                        int     80h                             ; LINUX -
.text:0000000000001328
.text:0000000000001328                              ; ---------------------------------------------------------------------------
.text:000000000000132A 00                           db 0
.text:000000000000132B                              ; ---------------------------------------------------------------------------
.text:000000000000132B
.text:000000000000132B                              label1:                                 ; CODE XREF: .t
```



然后在main应该开始的地方create function



就能f5了。

```
puts("Watashino RSA-wo mitekudasai!");
__isoc99_scanf("%s", v7);
if ( (unsigned int)len(v7) == 52 )
{
  for ( i = 0; i <= 12; ++i )
  {
    v4 = s2n(&v7[4 * i], 4LL);
    v6[i + 14] = v4;
    v5 = qpow(v6[i + 14], 0721LL, 3162244531LL);
    v6[i] = v5;
    if ( v6[i] != c[i] )
    {
      puts("ls -al $HOME/.rustup");
      return 1;
    }
  }
  puts("yes");
  return 0;
}
else
{
  puts("ls -al $HOME/.rustup");
  return 1;
}
```

发现是RSA加密。把密文挑出来。

```
.rodata:0000000000002020                                ; _QWORD c[13]
.rodata:0000000000002020 C5 2E 3C 75 00 00 00 00 36 C7+c dq 753C2EC5h, 8D90C736h, 81282CB0h, 7EECC470h, 944E15D3h, 2C7AC726h, 717E8070h, 30CBE439h, 0B1D95A9Ch
.rodata:0000000000002020 90 8D 00 00 00 00 B0 2C 28 81+                                 ; DATA XREF: main+15F↑o
.rodata:0000000000002020 00 00 00 00 70 C4 EC 7E 00 00+dq 6DB667BBh, 1240463Ch, 77CBFE64h, 11D8BE59h
.rodata:0000000000002088                                ; const char s[]
.rodata:0000000000002088 57 61 74 61 73 68 69 6E 6F 20+s db 'Watashino RSA-wo mitekudasai!',0  ; DATA XREF: main+B↑o
.rodata:00000000000020A6 25 73 00                       aS db '%s',0                    ; DATA XREF: main+24↑o
.rodata:00000000000020A9                                ; const char aLsAlHomeRustup[]
```

公钥很小，随便分解。

```
C:\Users\Laffey>Downloads\yafu-1.34\yafu-Win32.exe
factor(3162244531)


fac: factoring 3162244531
fac: using pretesting plan: normal
fac: no tune info: using qs/gnfs crossover of 95 digits
div: primes less than 10000
rho: x^2 + 3, starting 1000 iterations on C10
rho: x^2 + 2, starting 1000 iterations on C10
Total factoring time = 0.0035 seconds


***factors found***

P5 = 56099
P5 = 56369

ans = 1
```

```python
from Crypto.Util.number import*
e=0o721
p=56099
q=56369
n=p*q
phi=(p-1)*(q-1)
d=pow(e,-1,phi)
ca=[0x00000000753C2EC5, 0x000000008D90C736, 0x0000000081282CB0,
0x000000007EECC470, 0x00000000944E15D3, 0x000000002C7AC726, 0x00000000717E8070,
0x0000000030CBE439, 0x00000000B1D95A9C, 0x000000006DB667BB, 0x0000000001240463C,
0x0000000077CBFE64, 0x0000000011D8BE59]
flag=b''

for c in ca:
    m=pow(c,d,n)
    flag+=long_to_bytes(m)

print(flag)
```

```
>>> print(flag)
b'flag{reverse_is_N0T_@lways_jusT_RE_myy_H@bIb1!!!!!!}'
>>>
```

flag{reverse_is_N0T_@lways_jusT_RE_myy_H@bIb1!!!!!!}