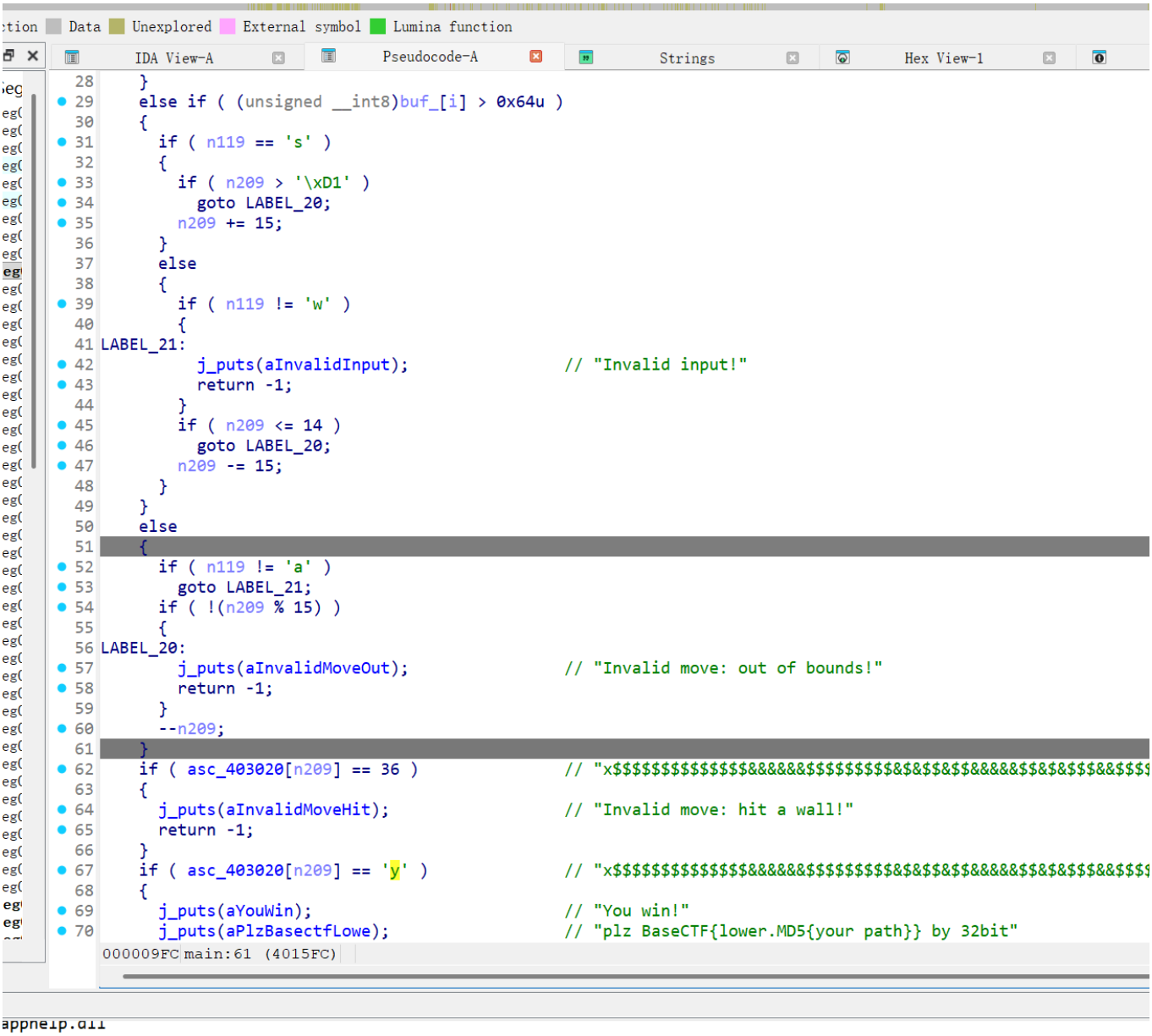
逆向wp-5

逆向题目wp

张程思

一,maze

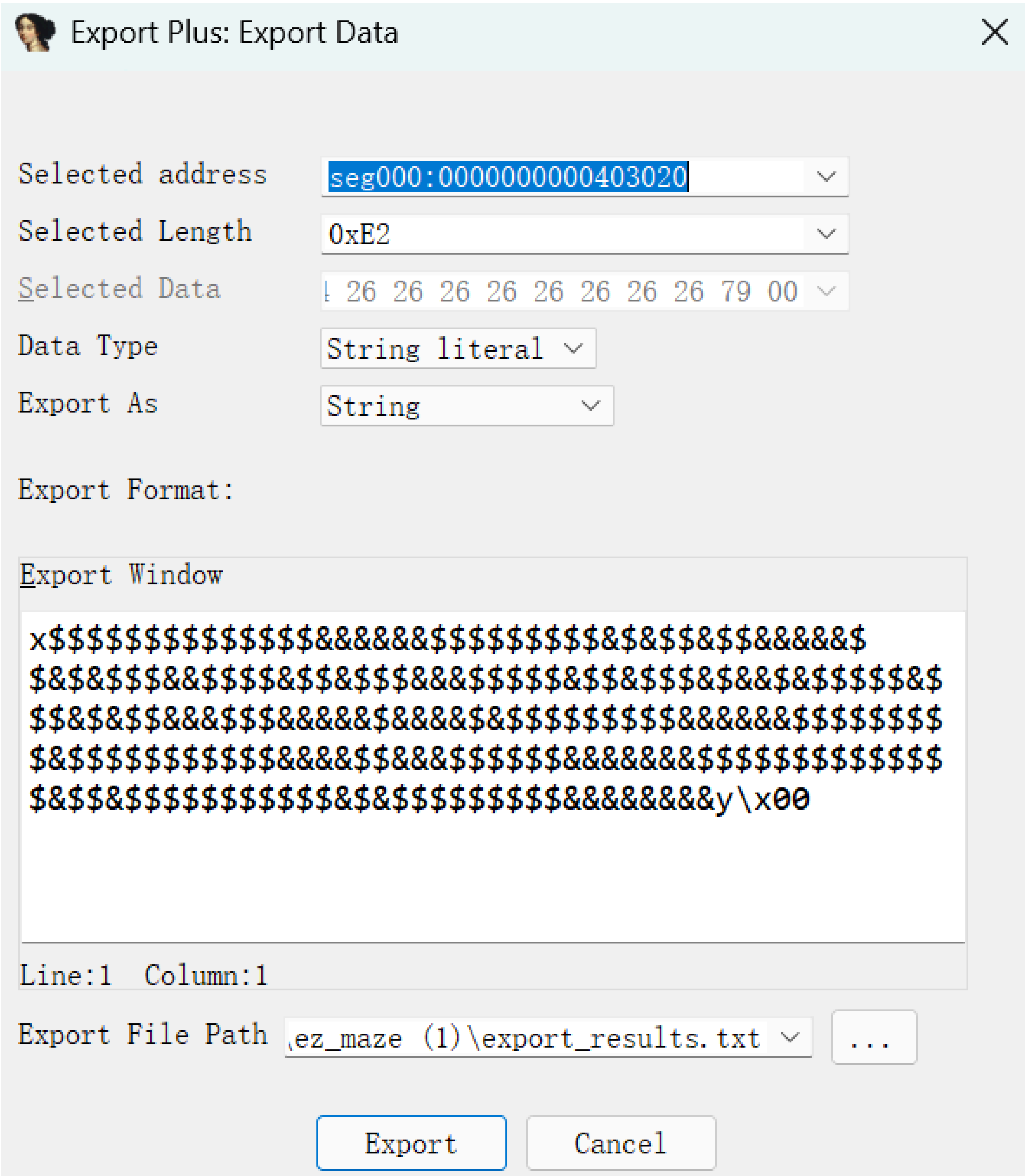
1,根据wasd这四个字母可以判断出来是迷宫问题



2,看一下string这里,找到了迷宫的大致

```
seg000:0000000000402ED8                 dq 0FFFFFFFFFFFFFFFFh
seg000:0000000000402EE0 qword_402EE0    dq 24h dup(0)           ; DATA XREF: seg000:off_403120↓o
seg000:0000000000403000 dword_403000    dd 0Ah                  ; DATA XREF: sub_401180:loc_40130F↑w
seg000:0000000000403004                 align 20h
seg000:0000000000403020 asc_403020      db 'x$$$$$$$$$$$$$$&&&&&&$$$$$$$$$&$&$$&$$&&&&&$$&$&$$$&&$$$$&$$&$$$&'
seg000:0000000000403020                                         ; DATA XREF: main+1BF↑o
seg000:0000000000403020                                         ; main+1E6↑o ...
seg000:0000000000403061                 db '&&$$$$$&$$&$$$&$&&$&$$$$$&$$$&$&$$&&&$$$&&&&&$&&&&$&$$$$$$$$$&&&&'
seg000:00000000004030A2                 db '&&$$$$$$$$$&$$$$$$$$$$$&&&&$$&&&$$$$$$&&&&&&&$$$$$$$$$$$$$$&$$&$$'
seg000:00000000004030E3                 db '$$$$$$$$$&$&$$$$$$$$$&&&&&&&&y',0
seg000:0000000000403102                 align 20h
seg000:0000000000403120 off_403120      dq offset qword_402EE0  ; DATA XREF: sub_401790+4↑r
seg000:0000000000403120                                         ; sub_401790+15↑r ...
seg000:0000000000403128                 align 10h
seg000:0000000000403130                 dq 0FFFFFFFFFFFFFFFFh, 0
seg000:0000000000403140 qword_403140    dq 0FFFFFFFFh, 0        ; DATA XREF: seg000:off_4044F0↓o
seg000:0000000000403150                 dq 0FFh, 0
seg000:0000000000403160 qword_403160    dq 2, 0                 ; DATA XREF: seg000:off_4043E0↓o
seg000:0000000000403170 qword_403170    dq 0FFFFFFFFh, 0        ; DATA XREF: seg000:off_4043F0↓o
```

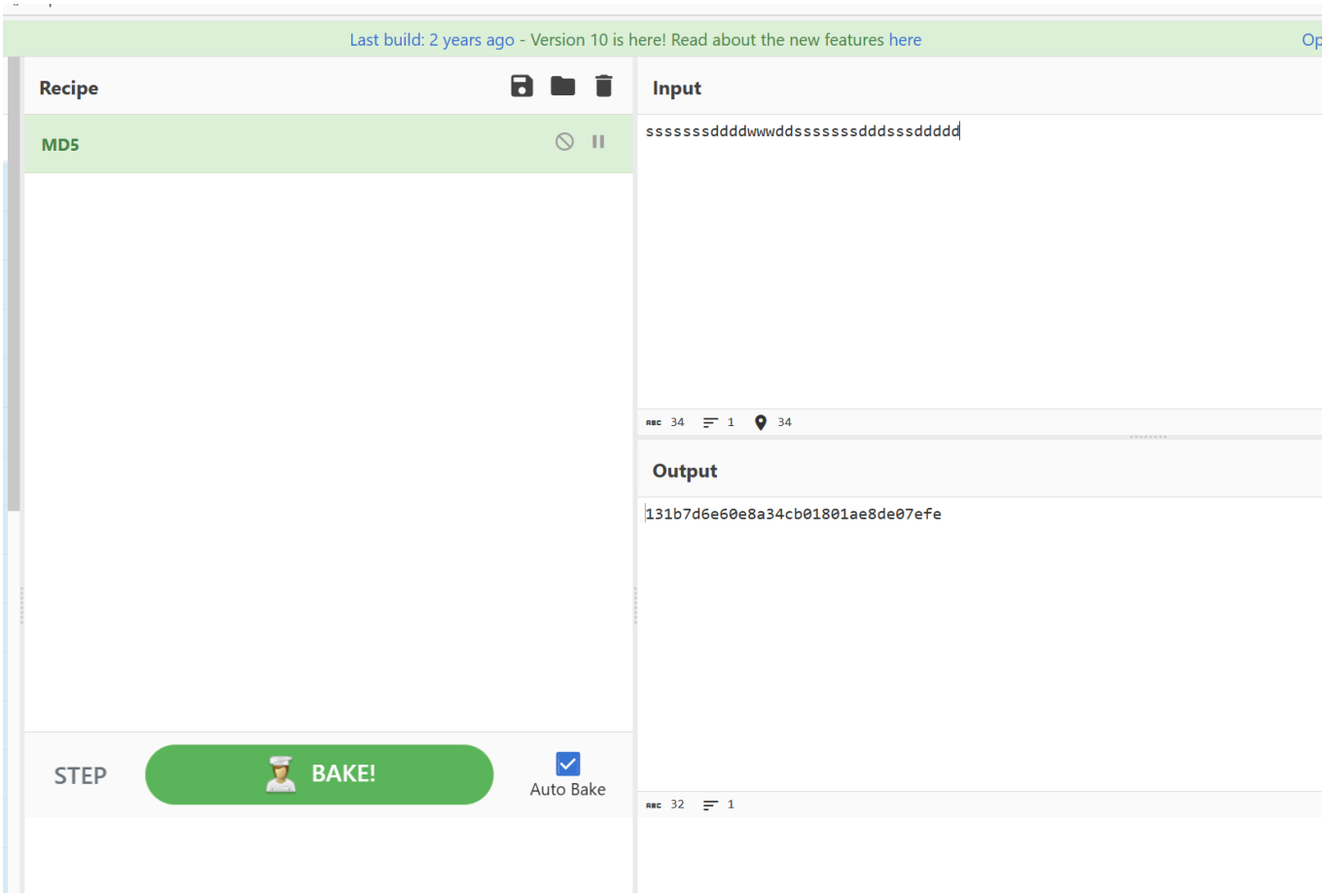
3,提取一下,自己在整理一下



```
'x', '$', '$', '$', '$', '$', '$', '$', '$', '$', '$', '$', '$', '$', '$', '$',
'&', '&', '&', '&', '&', '&', '$', '$', '$', '$', '$', '$', '$', '$', '$', '$',
'&', '$', '&', '$', '$', '&', '$', '$', '&', '&', '&', '&', '&', '$', '$', '$',
'&', '$', '&', '$', '$', '$', '&', '&', '$', '$', '$', '$', '&', '$', '$', '$',
'&', '$', '$', '$', '&', '&', '&', '$', '$', '$', '$', '$', '&', '$', '$', '$',
'&', '$', '$', '$', '&', '$', '&', '&', '$', '&', '$', '$', '$', '$', '$', '$',
'&', '$', '$', '$', '&', '$', '&', '$', '$', '&', '&', '&', '$', '$', '$', '$',
'&', '&', '&', '&', '&', '$', '&', '&', '&', '&', '$', '&', '$', '$', '$', '$',
'$', '$', '$', '$', '$', '$', '&', '&', '&', '&', '&', '&', '$', '$', '$', '$',
'$', '$', '$', '$', '$', '$', '&', '$', '$', '$', '$', '$', '$', '$', '$', '$',
'$', '$', '$', '&', '&', '&', '&', '$', '$', '&', '&', '&', '$', '$', '$', '$',
'$', '$', '$', '&', '&', '&', '&', '&', '&', '&', '$', '$', '$', '$', '$', '$',
'$', '$', '$', '$', '$', '$', '$', '$', '$', '&', '$', '$', '&', '$', '$', '$',
'$', '$', '$', '$', '$', '$', '&', '$', '$', '&', '$', '&', '$', '$', '$', '$',
'$', '$', '$', '$', '$', '$', '&', '&', '&', '&', '&', '&', '&', '&', 'y'
```

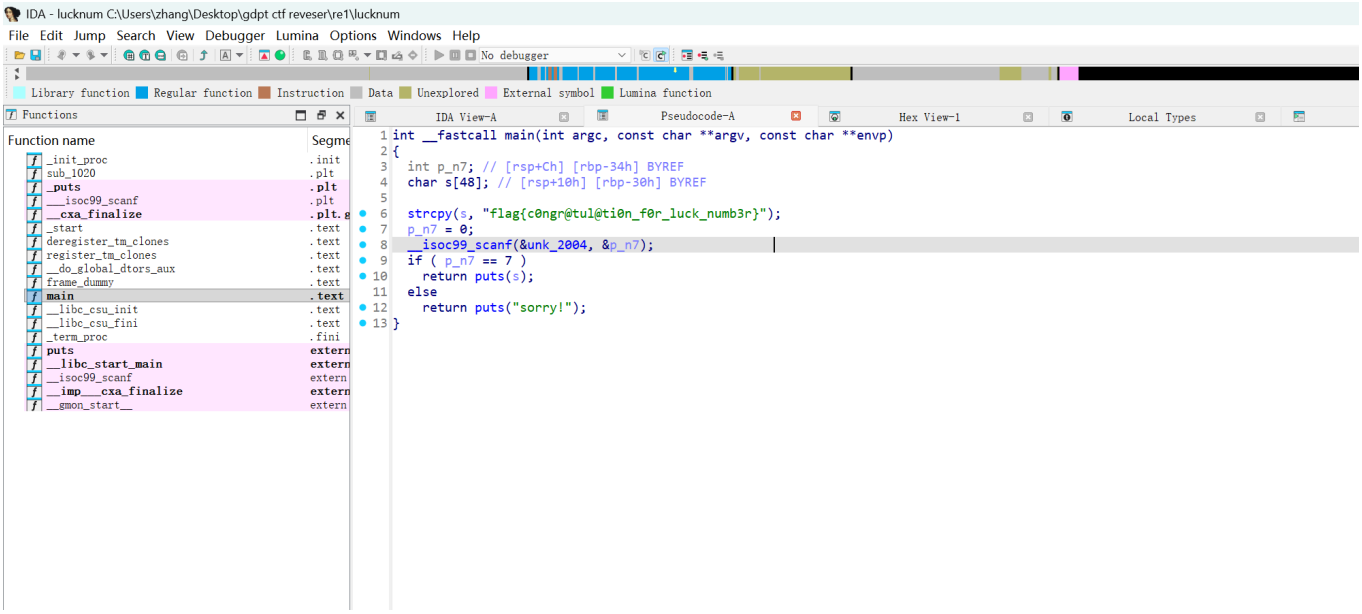
4,这里手动走一下也行，就是最短路径了，sssssssddddwwwddsssssssddddsssddddd

5,md5解一下就出来了



二,re1

1,这次找的题目好像有点水啊,我随便找的比赛



三,re2

1,还原原本的字符串

IDA View-A | Pseudocode-B | Pseudocode-A | Hex View-1 | Local

```
  v10[17] = 26;
  v10[18] = 20;
  v10[19] = 59;
  v10[20] = 4;
  v10[21] = 0;
  printf("plz enter the flag:");
  while ( 1 )
  {
    v7 = getch();
    v12[v11] = v7;
    if ( !v7 || v12[v11] == '\r' )
      break;
    if ( v12[v11] == '\b' )
    {
      printf("\b\b");
      --v11;
    }
    else
    {
      printf("%c", (char)v12[v11++]);
    }
  }
  v9 = 0;
  for ( n17 = 0; n17 < 17; ++n17 )
  {
    if ( (char)v12[n17] != byte_415768[v10[n17]] )
      v9 = 1;
  }
  if ( v12[17] != 49 || v12[18] != 48 || v12[19] != 50 || v12[20] != 52 || v12[21] != 125 )
    v9 = 1;
  v12[v11] = 0;
  printf("\r\n");
  if ( v9 )
  {
    printf("u r wrong\r\n\r\n");
    main(argc_1, argv_1, envp_1);
  }
  else
  {
    printf("u r right!\r\n");
  }
  system("pause");
  return 0;
}
```

000009AA _main_0:59 (4115AA)

```
a:00415767                 align 4
a:00415768 ; char byte_415768[]
a:00415768 byte_415768     db 73h                  ; DATA XREF: _main_0+1E3↑r
a:00415769 aKfxeeftFGyrygt db 'KfxEeft}f{gyrYgthtyhifsjei53UUrrr_t2cdsef66246087138\0087138',0
a:004157A6                 db    0
a:004157A7                 db    0
a:004157A8                 db    0
a:004157A9                 db    0
a:004157AA                 db    0
a:004157AB                 db    0
a:004157AC                 db    0
a:004157AD                 db    0
a:004157AE                 db    0
a:004157AF                 db    0
a:004157B0                 db    0
```

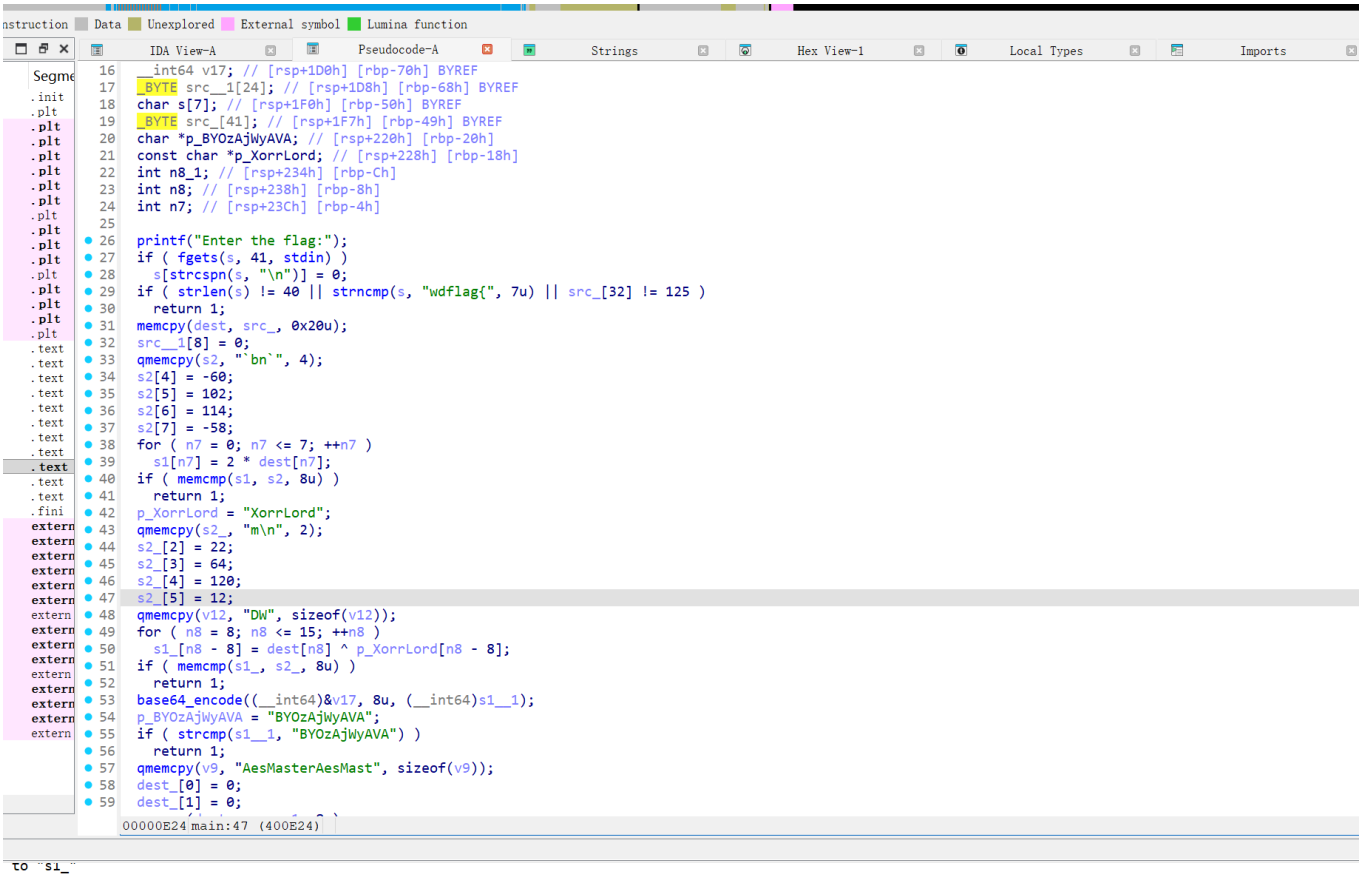
2,

```
data = "eKfxEeft}f{gyrYgthtyhifsjei53UUrrr_t2cdsef66246087138\0087138"

v10 = [1,4,14,10,5,36,23,42,13,19,28,13,27,39,48,41,42]

flag_part = ''.join([data[i] for i in v10])

full_flag = (f"{flag_part}1024}}")
```

```
print(full_flag)
```

四,re3

1,题目给的提示是要闯关四个护卫,看了一下ida,应该就是四个加密,一个乘法,一个异或,一个base64,一个是AES加密,



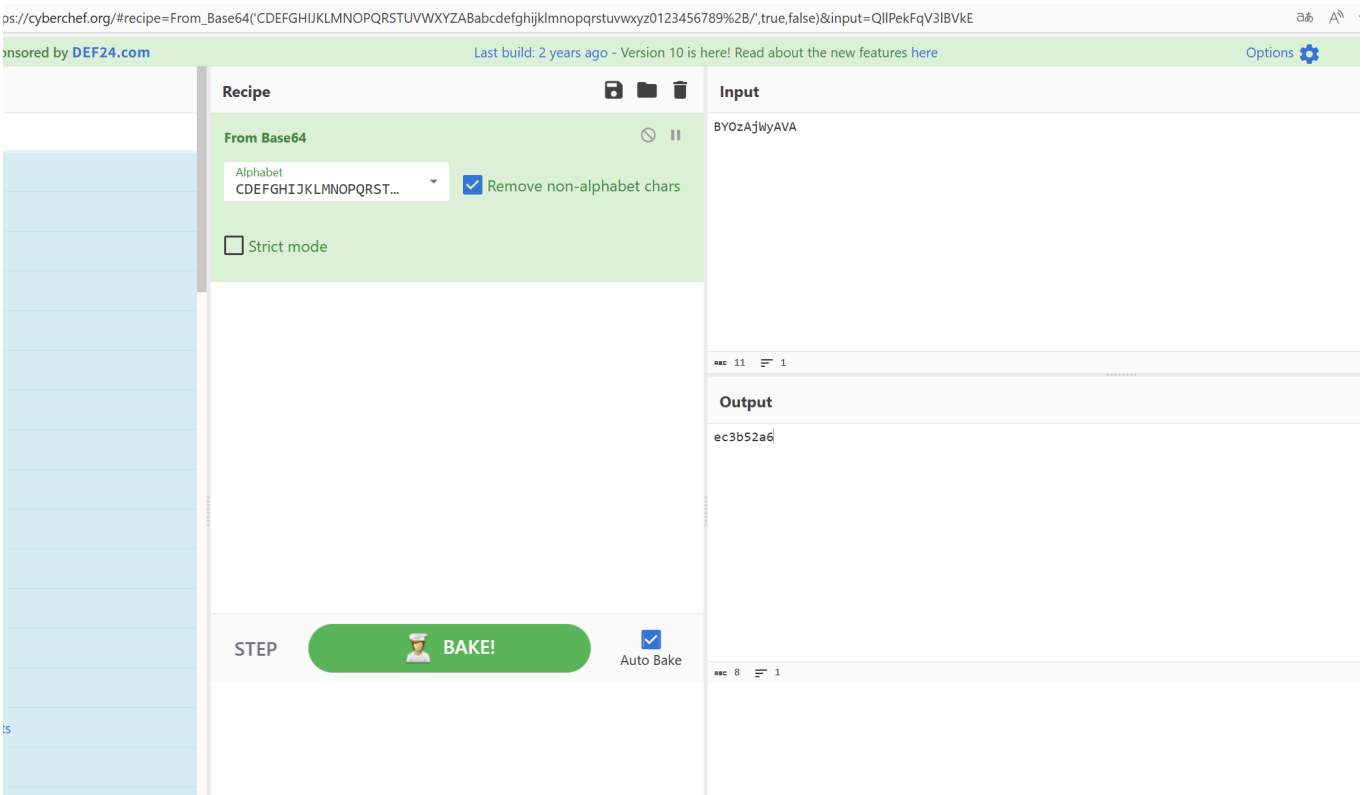
2,第一二步脚本附上

```
memcpy(dest, v51, 0x20uLL);

v49 = 0;

s2 = 96;

v38 = 98;

v39 = 110;

v40 = 96;

v41 = -60;
```

```
v42 = 102;

v43 = 114;

v44 = -58;

for ( i = 0; i <= 7; ++i )

s1[i] = 2 * dest[i];

96

=>

0x60

| 0x60 / 2 = 0x30 = 48 (ASCII '0')

98

=>

0x62

| 0x62 / 2 = 0x31 = 49 (ASCII '1')

110 =>

0x6E

| 0x6E / 2 = 0x37 = 55 (ASCII '7')

96

=>

0x60

| 0x60 / 2 = 0x30 = 48 (ASCII '0')

-60 =>

0xC4

| 0xC4 / 2 = 0x62 = 98
```

```
(ASCII 'b')

102 =>

0x66

| 0x66 / 2 = 0x33 = 51 (ASCII '3')

114 =>

0x72

| 0x72 / 2 = 0x39 = 57 (ASCII '9')

-58 =>

0xC6

| 0xC6 / 2 = 0x63 = 99

(ASCII 'c')
```

3,第三步用赛博厨子



4,第四步还是写脚本

```
qmemcpy(v9, "AesMasterAesMast", sizeof(v9));
```

```
v8[0] = 0LL;

v8[1] = 0LL;

memcpy(v8, v18, 8uLL);

for ( k = 8; k <= 15; ++k )

*((_BYTE *)v8 + k) = 8;

AES_set_encrypt_key(v9, 128LL, v6);

AES_encrypt(v8, v7, v6);

hex_to_string(v7, 16LL, v5);

v4[0] = -10;

v4[1] = -85;

v4[2] = 71;

v4[3] = -66;

v4[4] = 113;

v4[5] = -28;

v4[6] = 1;

v4[7] = -36;

v4[8] = 3;

v4[9] = 48;

v4[10] = -97;

v4[11] = -15;

v4[12] = 67;

v4[13] = -15;

v4[14] = -45;
```

```
v4[15] = 102;

if ( memcmp(v7, v4, 0x10uLL) )

return 1;

puts("Correct Flag!");

return 0;

}
```

然后再exp

```
from Crypto.Cipher import AES

from Crypto.Util.Padding import unpad

# 提供的数据 v4

v4 = bytes([0xF6, 0xAB, 0x47, 0xBE, 0x71, 0xE4, 0x01, 0xDC, 0x03, 0x30,

0x9F, 0xF1, 0x43, 0xF1, 0xD3, 0x66])

# 使用"AesMasterAesMast"作为密钥，编码为字节

key = "AesMasterAesMast".encode()

# 初始化AES解密器，选择ECB模式

cipher = AES.new(key, AES.MODE_ECB)

# 解密数据

decrypted_data = cipher.decrypt(v4)

# 去除填充

try:

unpadded_data = unpad(decrypted_data, AES.block_size)

# 输出解密后的数据，假设是ASCII字符串

print("Decrypted data:", unpadded_data.decode('utf-8',
```

```
errors='ignore'))

except ValueError as e:

print(f"Padding error or decryption issue: {e}")
```

五,re4

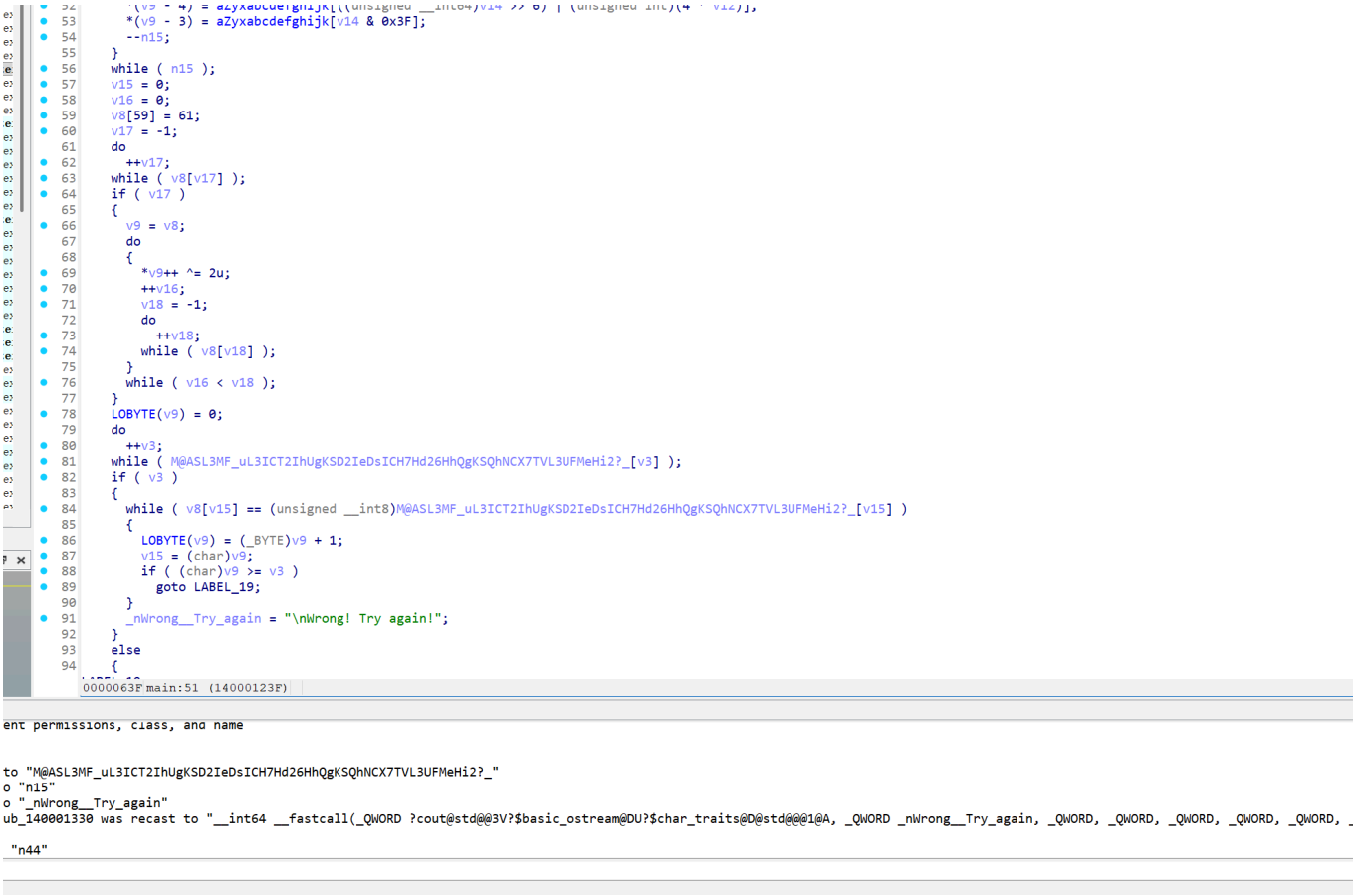
1,先异或再base64换表

IDA View-A | Pseudocode-A | Hex View-1 | Local Types

```
int __fastcall main(int argc, const char **argv, const char **envp)
{
  unsigned __int64 v3; // rbx
  __int64 n44; // rax
  char *v5; // rax
  char *v6; // r10
  __int64 n15; // r11
  char *v8; // r8
  _BYTE *v9; // r9
  unsigned __int8 v10; // dl
  unsigned __int64 v11; // rax
  int v12; // ecx
  __int64 v13; // rax
  unsigned int v14; // edx
  __int64 v15; // rdx
  int v16; // r10d
  __int64 v17; // rax
  unsigned __int64 v18; // rax
  const char *_nWrong__Try_again; // rdx
  _BYTE M@ASL3MF_uL3ICT2IhUgKSD2IeDsICH7Hd26HhQgKSQhNCX7TVL3UFMeHi2?_[65]; // [rsp+20h] [rbp-E0h] BYREF
  char v22; // [rsp+61h] [rbp-9Fh] BYREF

  strcpy(
    M@ASL3MF_uL3ICT2IhUgKSD2IeDsICH7Hd26HhQgKSQhNCX7TVL3UFMeHi2?_,
    "M@ASL3MF`uL3ICT2IhUgKSD2IeDsICH7Hd26HhQgKSQhNCX7TVL3UFMeHi2?");
  sub_140001020("input your flag:");
  sub_140001080("%s");
  v3 = -1;
  n44 = -1;
  do
    ++n44;
  while ( M@ASL3MF_uL3ICT2IhUgKSD2IeDsICH7Hd26HhQgKSQhNCX7TVL3UFMeHi2?_[n44 + 64] );
  if ( n44 == 44 )
  {
    v5 = (char *)malloc(0x3Du);
    v6 = &v22;
    n15 = 15;
    v8 = v5;
    v5[60] = 0;
    v9 = v5 + 2;
    do
    {
      v10 = *(v6 - 1);
      v9 += 4;
```

000004E0 main:1 (1400010E0)

missions, class, and name



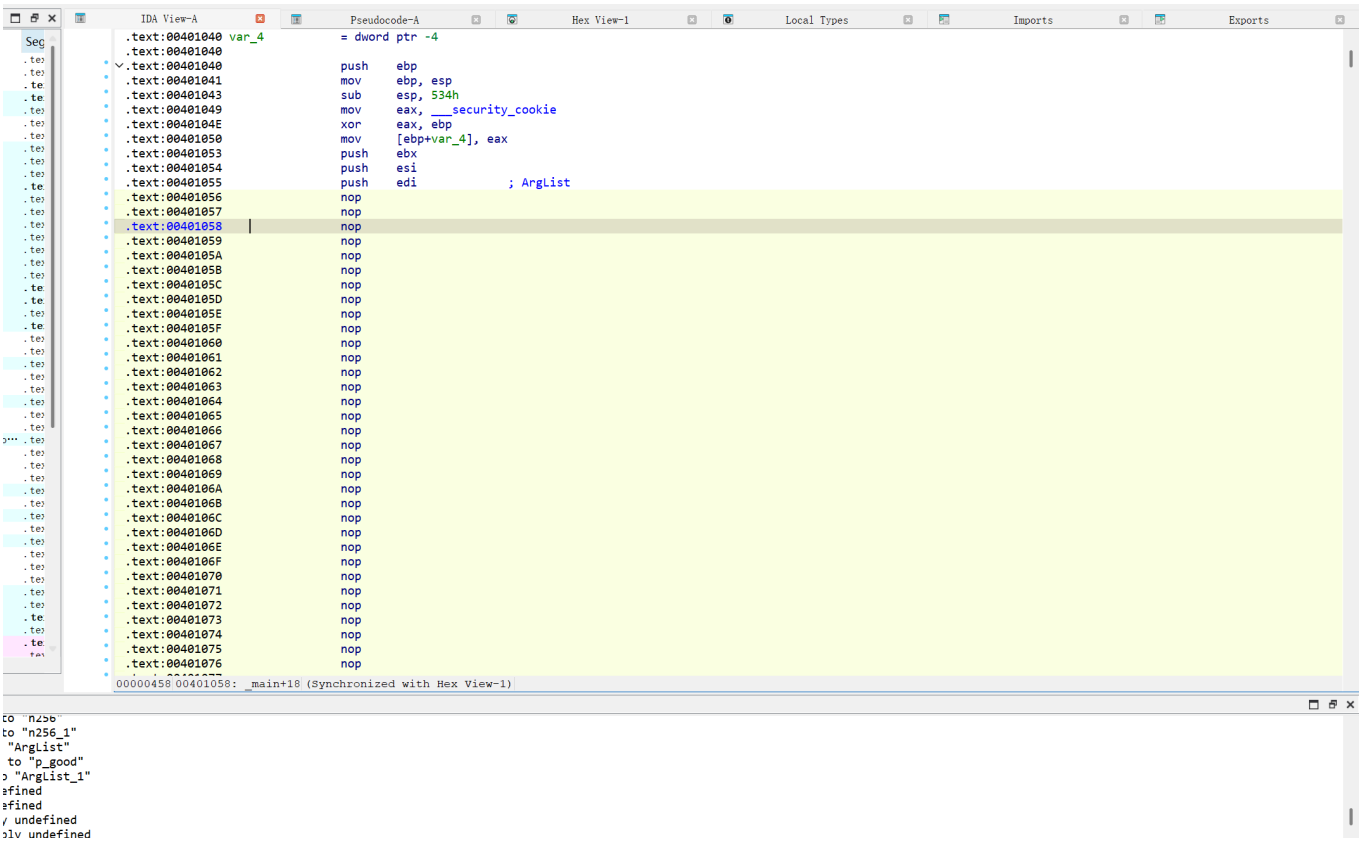
2,脚本如下

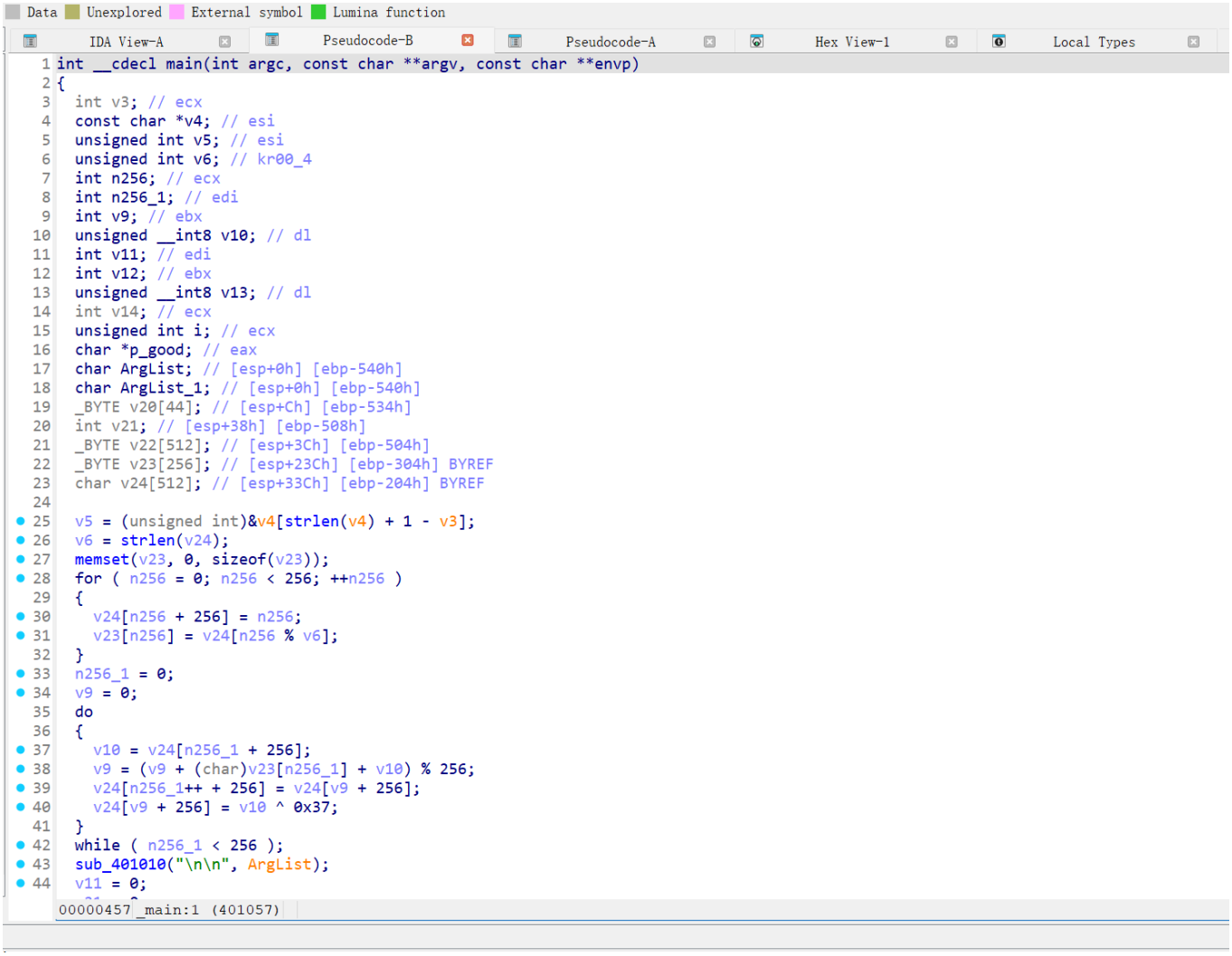
```
import base64

# 自定义 base64 表

custom_table =

"ZYXABCDEFGHIJKLMNOPQRSTUVWzyxabcdefghijklmnopqrstuvw0123456789+/"

standard_table =

"ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789+/"

encoded =

"M@ASL3MF`uL3ICT2IhUgKSD2IeDsICH7Hd26HhQgKSQhNCX7TVL3UFMeHi2?"

xor_result = bytes([ord(c) ^ 2 for c in encoded])

trans = str.maketrans(custom_table, standard_table)

standard_b64 = xor_result.decode().translate(trans)

if len(standard_b64) % 4 != 0:
```

```
standard_b64 += '=' * (4 - len(standard_b64) % 4)

flag = base64.b64decode(standard_b64).decode()

print("Flag:", flag)
```

六,re5

1,初步打开发现无法反编译,我去了一下花就好了





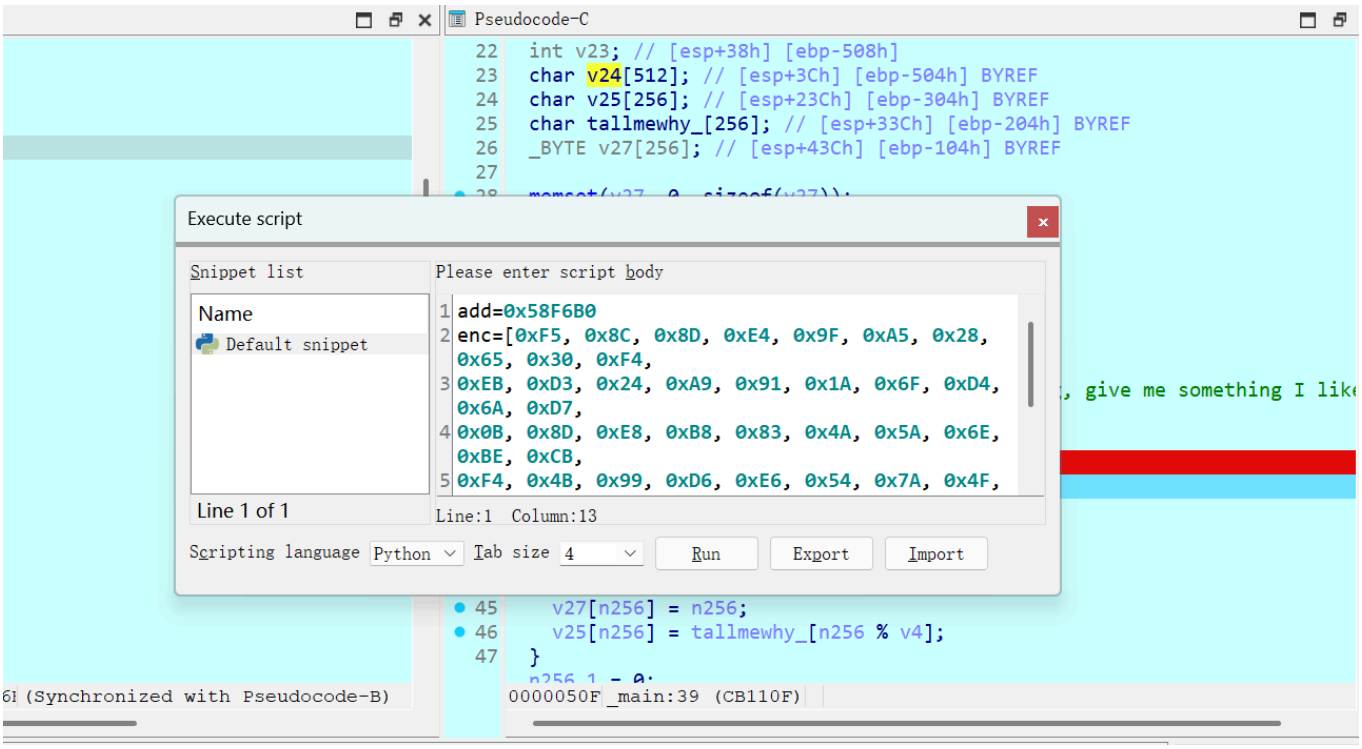

```
int __cdecl main(int argc, const char **argv, const char **envp)
{
  int v3; // ecx
  const char *v4; // esi
  unsigned int v5; // esi
  unsigned int v6; // kr00_4
  int n256; // ecx
  int n256_1; // edi
  int v9; // ebx
  unsigned __int8 v10; // dl
  int v11; // edi
  int v12; // ebx
  unsigned __int8 v13; // dl
  int v14; // ecx
  unsigned int i; // ecx
  char *p_good; // eax
  char ArgList; // [esp+0h] [ebp-540h]
  char ArgList_1; // [esp+0h] [ebp-540h]
  _BYTE v20[44]; // [esp+Ch] [ebp-534h]
  int v21; // [esp+38h] [ebp-508h]
  _BYTE v22[512]; // [esp+3Ch] [ebp-504h]
  _BYTE v23[256]; // [esp+23Ch] [ebp-304h] BYREF
  char v24[512]; // [esp+33Ch] [ebp-204h] BYREF

  v5 = (unsigned int)&v4[strlen(v4) + 1 - v3];
  v6 = strlen(v24);
  memset(v23, 0, sizeof(v23));
  for ( n256 = 0; n256 < 256; ++n256 )
  {
    v24[n256 + 256] = n256;
    v23[n256] = v24[n256 % v6];
  }
  n256_1 = 0;
  v9 = 0;
  do
  {
    v10 = v24[n256_1 + 256];
    v9 = (v9 + (char)v23[n256_1] + v10) % 256;
    v24[n256_1++ + 256] = v24[v9 + 256];
    v24[v9 + 256] = v10 ^ 0x37;
  }
  while ( n256_1 < 256 );
  sub_401010("\n\n", ArgList);
  v11 = 0;
```

_1"
:"
ood"

2,一个rc4,但是中间有异或,应该是魔改rc4

```
  for ( n256 = 0; n256 < 256; ++n256 )
  {
    v24[n256 + 256] = n256;
    v23[n256] = v24[n256 % v6];
  }
  n256_1 = 0;
  v9 = 0;
  do
  {
    v10 = v24[n256_1 + 256];
    v9 = (v9 + (char)v23[n256_1] + v10) % 256;
    v24[n256_1++ + 256] = v24[v9 + 256];
    v24[v9 + 256] = v10 ^ 0x37;
  }
  while ( n256_1 < 256 );
  sub_401010("\n\n", ArgList);
  v11 = 0;
  v21 = 0;
  v12 = 0;
  if ( v5 )
  {
    do
    {
      v11 = (v11 + 1) % 256;
      v13 = v24[v11 + 256];
      v12 = (v13 + v12) % 256;
      v24[v11 + 256] = v24[v12 + 256];
      v24[v12 + 256] = v13;
      v14 = v21;
      v22[v21] ^= v24[(unsigned __int8)(v13 + v24[v11 + 256]) + 256];
      v21 = v14 + 1;
    }
    while ( v14 + 1 < v5 );
    v12 = 0;
  }
  for ( i = 0; i < v5; ++i )
    v12 = v22[i] == v20[i];
  p_good = (char *)&unk_402184;
  if ( v12 == 1 )
    p_good = aGood;
  sub_401010(p_good, ArgList_1);
  return 0;
}
0000054D _main:27 (40114D)
```

3,patch一下



4,最后的循环之前下个断点,找到密文flag就在旁边了