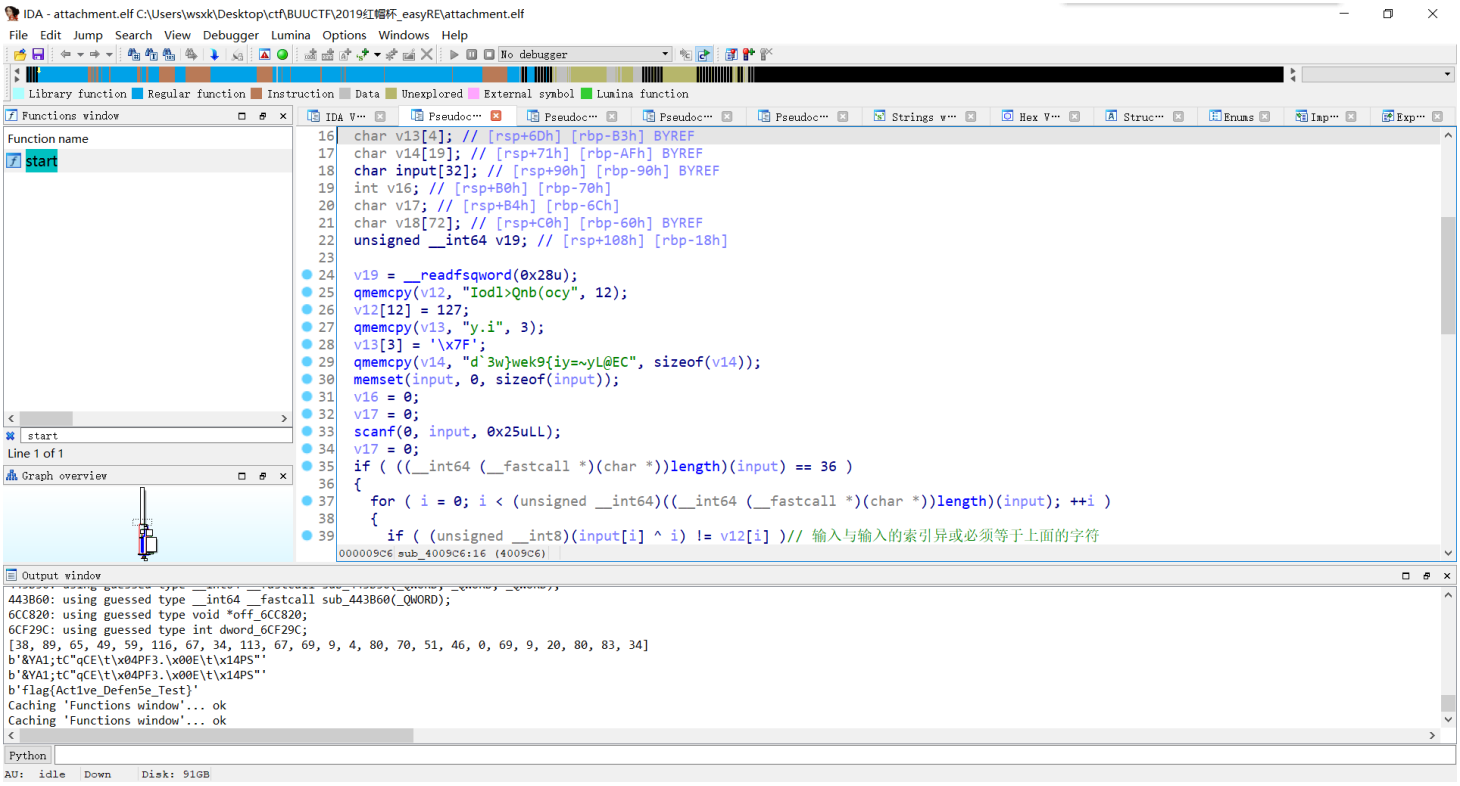
很有意思的ctf题，心理博弈好吧

学到了 fini段和init段的作用

也了解到了心理博弈

https://bbs.pediy.com/thread-254172.htm

ida拖入



顺着这个思路一直下去
得到两个结果

```
b'https://bbs.pediy.com/thread-254172.htm'
b'Info:The first four chars are `flag`'
```

很明显，自己被搞了

这题应该是考了fini_array的知识点

（看了wp才知道）
fini_array
fini段的作用：

此节区包含了可执行的指令，是进程终止代码的一部分。程序正常退出时，系统将安排执行这里的代码。

```
nit_array:00000000006CBEC8 _init_array     segment qword public 'DATA' use64
nit_array:00000000006CBEC8                 assume cs:_init_array
nit_array:00000000006CBEC8                 ;org 6CBEC8h
nit_array:00000000006CBEC8 funcs_4020B8    dq offset sub_400970    ; DATA XREF: sub_402080+2↑o
nit_array:00000000006CBEC8                                         ; sub_402080+A↑o ...
nit_array:00000000006CBEC8                 dq offset sub_4009AE
nit_array:00000000006CBED8 funcs_402130    dq offset sub_4005F0    ; DATA XREF: sub_402110:loc_402130↑r
nit_array:00000000006CBED8 _init_array     ends
nit_array:00000000006CBED8
ini_array:00000000006CBEE0 ; ===========================================================================
ini_array:00000000006CBEE0
ini_array:00000000006CBEE0 ; Segment type: Pure data
ini_array:00000000006CBEE0 ; Segment permissions: Read/Write
ini_array:00000000006CBEE0 _fini_array     segment qword public 'DATA' use64
ini_array:00000000006CBEE0                 assume cs:_fini_array
ini_array:00000000006CBEE0                 ;org 6CBEE0h
ini_array:00000000006CBEE0 off_6CBEE0      dq offset sub_400940    ; DATA XREF: sub_402080:loc_4020C8↑o
ini_array:00000000006CBEE0                                         ; sub_402110+6↑o
ini_array:00000000006CBEE8                 dq offset sub_400D35
ini_array:00000000006CBEF0                 dq offset sub_4005C0
ini_array:00000000006CBEF0 _fini_array     ends
```

可以看到在结束程序输入后，运行了3个函数

其中400d35函数有判断字符输入（其他两个其实都是正常的检查函数）

```
  8  unsigned __int64 v5; // [rsp+28h] [rbp-8h]
  9
 10  v5 = __readfsqword(0x28u);
 11  v1 = sub_43FD20(0LL) - qword_6CEE38;
 12  for ( i = 0; i <= 1233; ++i )
 13  {
 14    sub_40F790(v1);
 15    sub_40FE60();
 16    sub_40FE60();
 17    v1 = sub_40FE60() ^ 0x98765432;
 18  }
 19  v4 = v1;
 20  if ( ((unsigned __int8)v1 ^ byte_6CC0A0[0]) == 'f' && (HIBYTE(v4) ^ (unsigned __int8)byte_6CC0A3) == 'g' )
 21  {
 22    for ( j = 0; j <= 24; ++j )
 23      sub_410E90((unsigned __int8)(byte_6CC0A0[j] ^ *((_BYTE *)&v4 + j % 4)));
 24  }
 25  result = __readfsqword(0x28u) ^ v5;
 26  if ( result )
 27    sub_444020();
 28  return result;
 29}
00000D35 sub_400D35:6 (400D35)
```

可以看到v1和首四个字节异或应该得到flag

那么v1的每个字节为 flag和bcc0a0的前四个字节的异或

写脚本，就能得到flag