### 北京邮电大学—期末线上机试答题报告

|  |  |  |
| --- | --- | --- |
|  |  |  |
| 机 试 总 成 绩  (满分 100`) |  | |

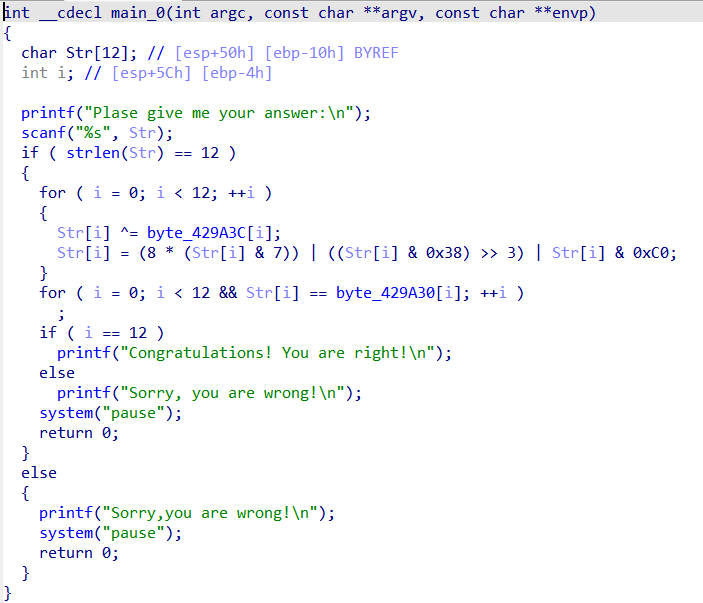
|  |  |
| --- | --- |
| **学院** | 网络空间安全学院 |
| **专业** | 信息安全 |
| **班级** | 2023211801班 |
| **姓名** | **詹冲** |
| **学号** | 2023211616 |
| **座位号** | 30 |
| **考核科目** | 《汇编语言与逆向工程》 |
| **考核方式** | 机试、开卷 |

**2025年06月29日（上午） 10:00—12:00**

**（10:00—12:00期间，机试电脑需全程断网）**

1. CPP1.exe (本题20`)

使用IDA打开第一个CPP1.exe文件，点击进入其主程序如下。



输入明文后，首先对明文长度进行判断，长度必须等于12，否则则会输出错误提示。



其次对于输入的明文需要与00429A3C处数组中的12字节进行异或。



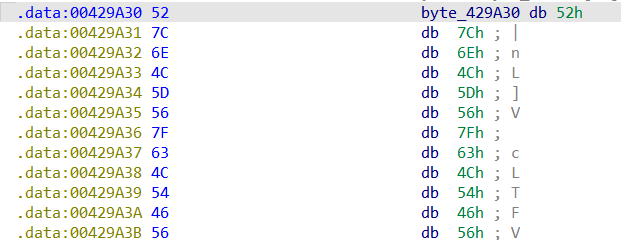
然后对明文每字节中的比特进行移位。保持最高两个比特不变，0-2比特与3-5比特调换位置。



最后得到的密文与00429A30处的数组进行比较，如果最终i等于12则正确，否则输出错误信息。

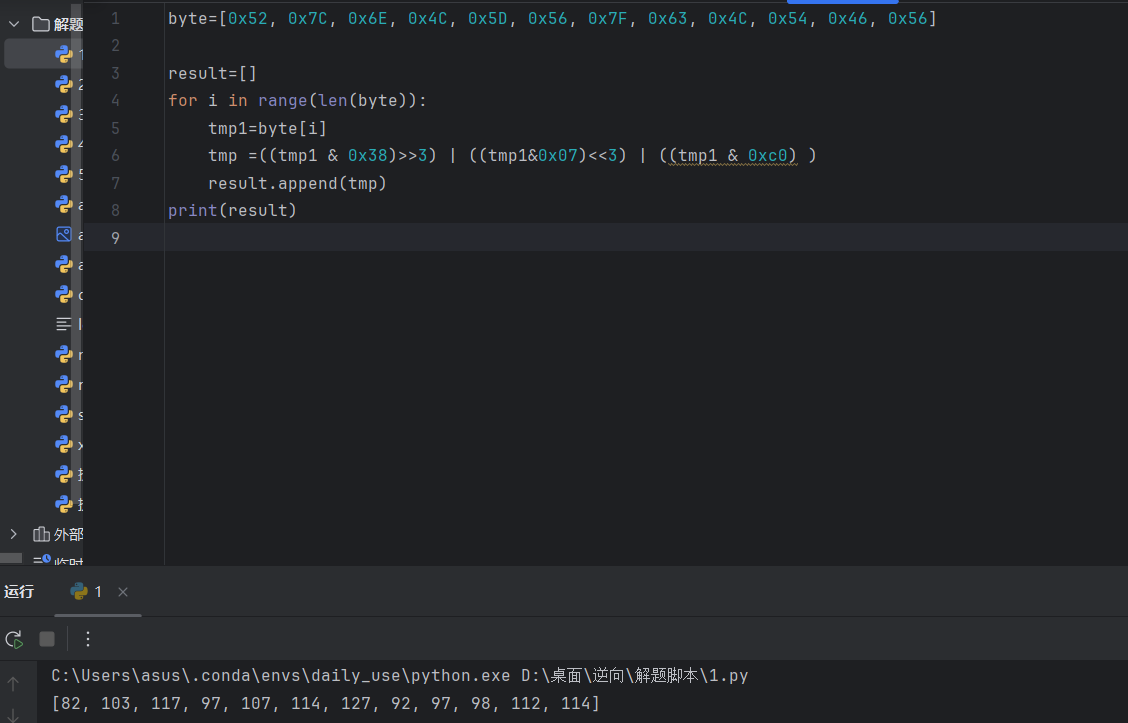


整理密文为[0x52, 0x7C, 0x6E, 0x4C, 0x5D, 0x56, 0x7F, 0x63, 0x4C, 0x54, 0x46, 0x56]

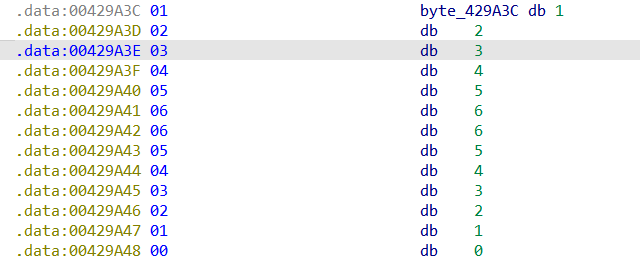


使用解密脚本进行解密，首先对密文字符进行比特移位操作。

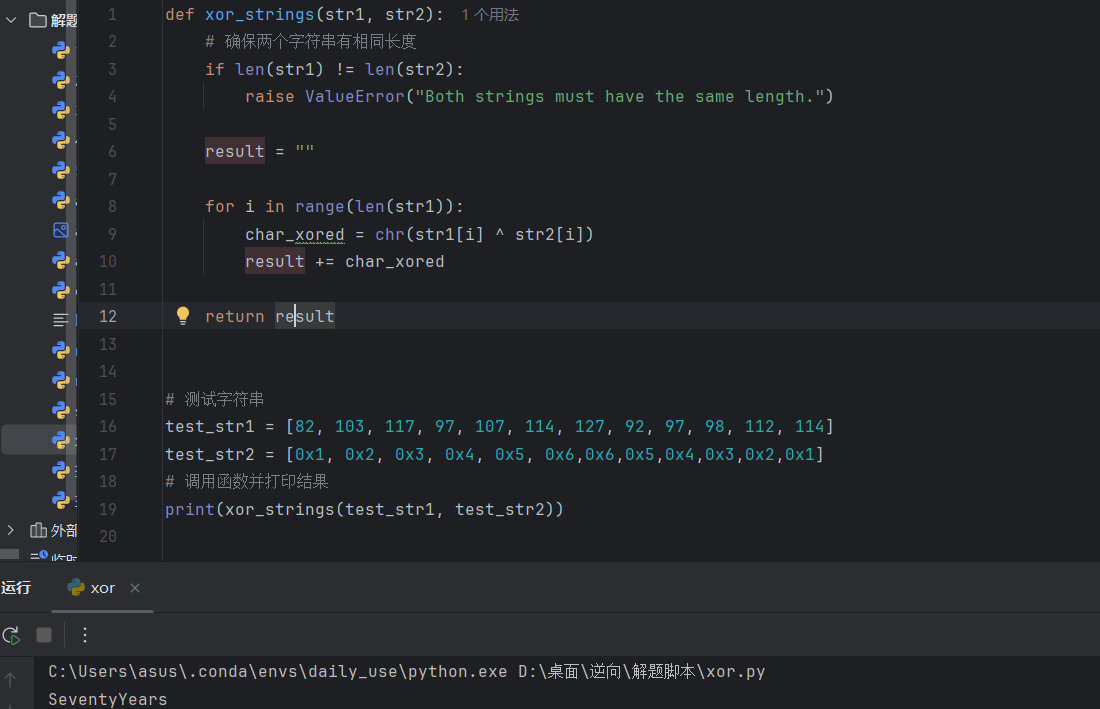
得到[82, 103, 117, 97, 107, 114, 127, 92, 97, 98, 112, 114]。



异或加密的密钥为[0x1, 0x2, 0x3, 0x4, 0x5, 0x6,0x6,0x5,0x4,0x3,0x2,0x1]



然后使用异或解密脚本对上述数组进行异或解密，得到flag为SeventyYears

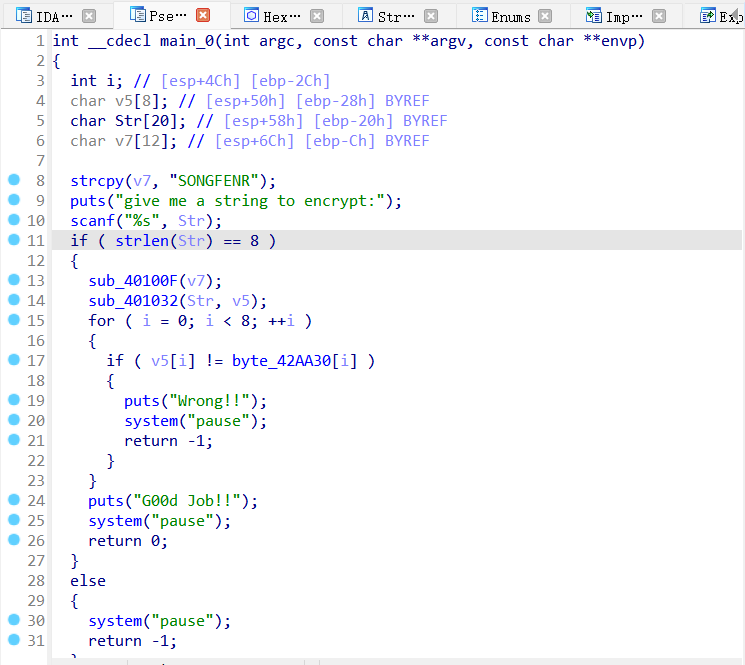


验证flag=SeventyYears发现程序通过了。



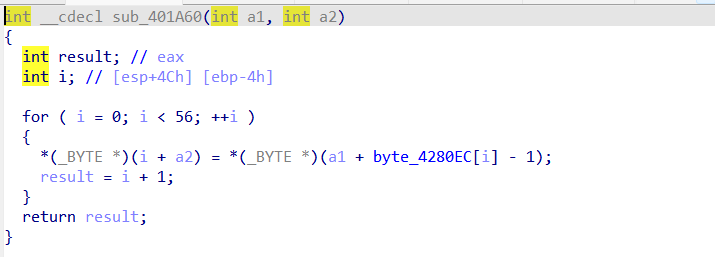
1. CPP2.exe (本题20`)

使用IDA打开第一个CPP2.exe文件，点击进入其主程序如下。

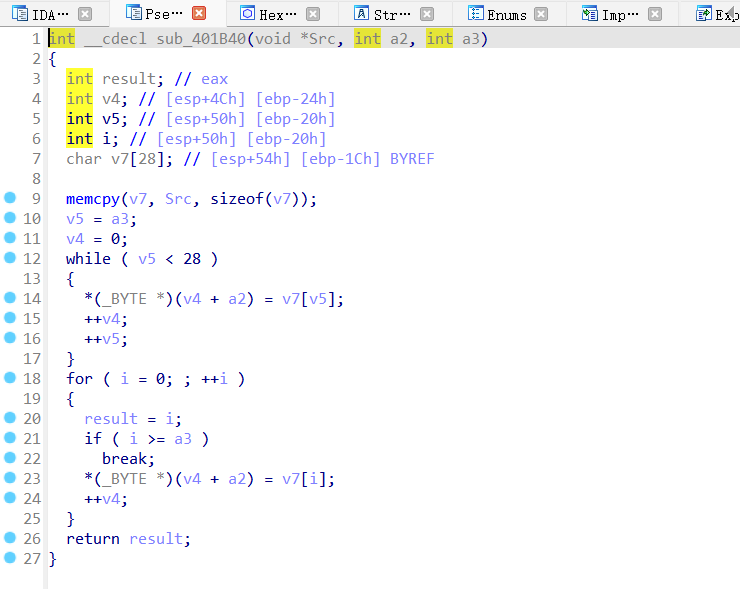


发现函数sub\_40100F是密钥编排函数，点击进入查看其详情

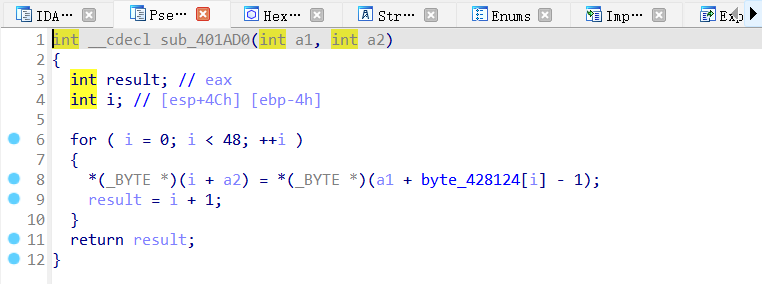
发现函数sub\_401A60是PC-1置换



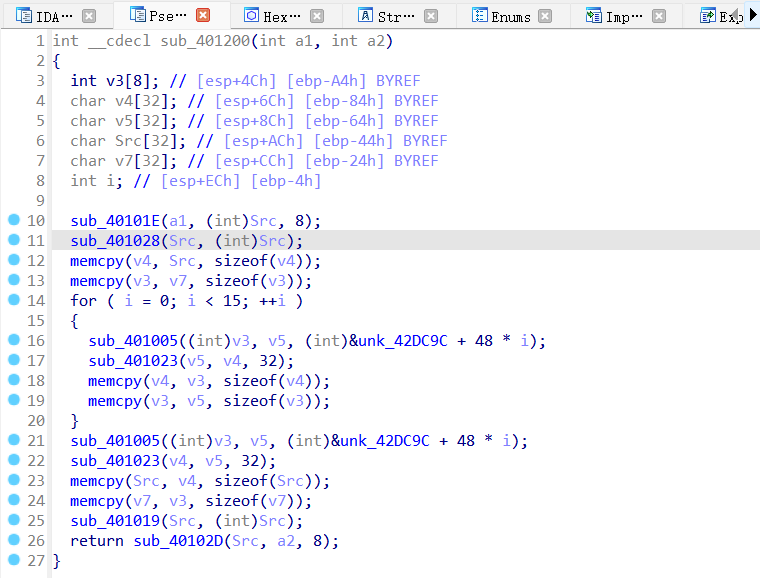
发现函数sub\_101b40是对56位密钥中的左右两个子密钥进行移位



发现函数sub\_401ad0是PC-2置换。



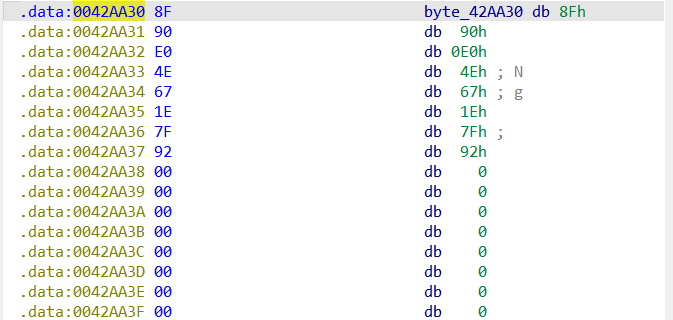
进入函数sub\_401200()发现是十六轮迭代。



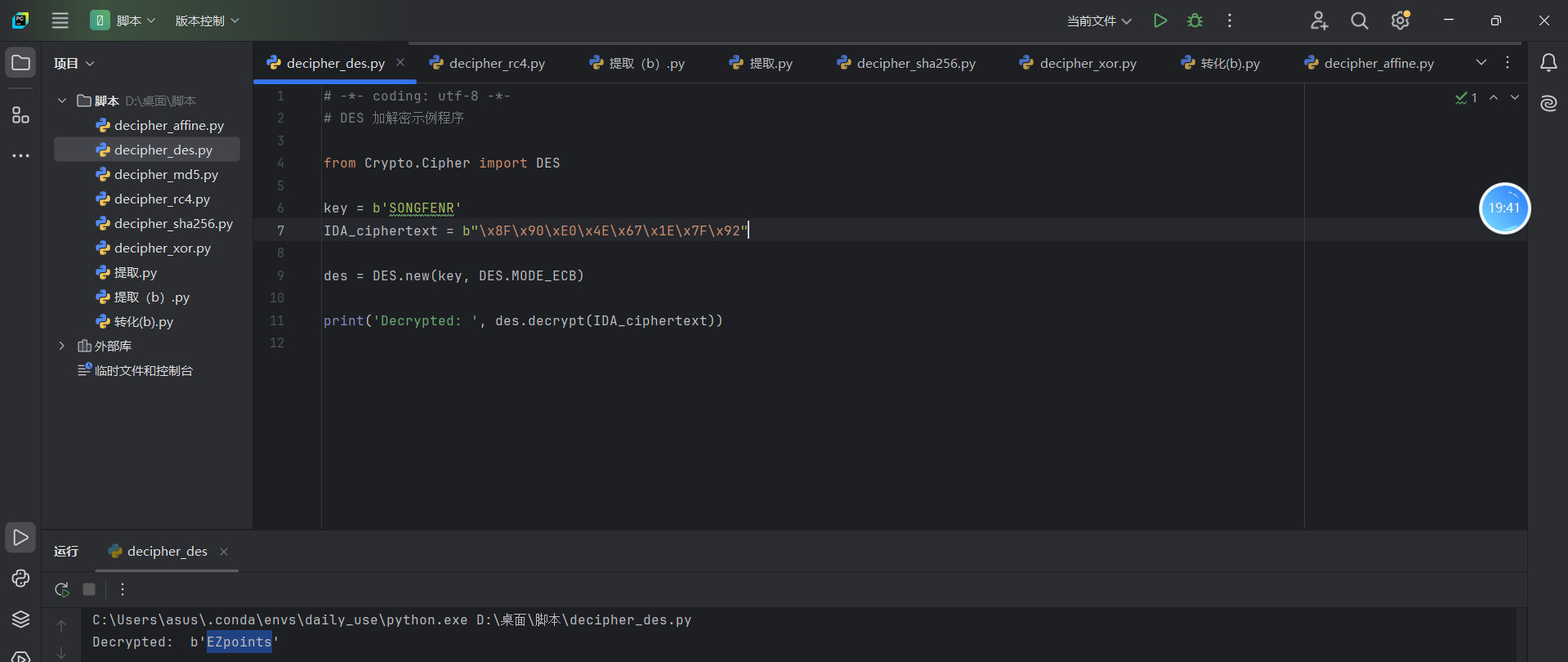
发现F函数即sub401005中有E扩展，密钥加，S盒代换等等

据此推断出加密算法为DES

从程序中可以得知DES的初始密钥为SONGFENR，最终的密文为在0042AA30处，具体见下。



使用DES解密程序进行解密，得到flag为EZpoints。

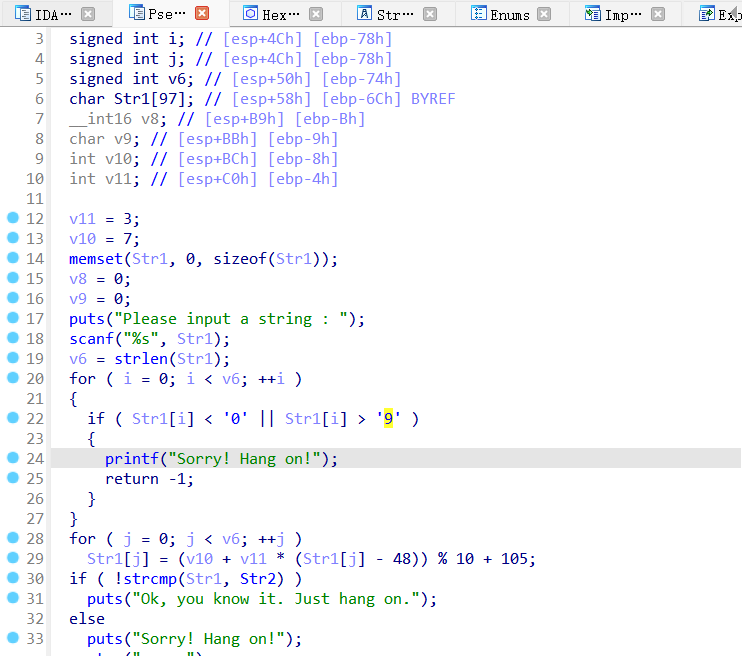


验证flag=EZpoints发现程序通过了。

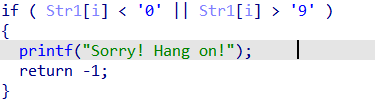


1. CPP3.exe (本题20`)

使用IDA打开第一个CPP3.exe文件，点击进入其主程序如下。



首先发现其对输入进行判断，输入必须在0-9之间，若不在则输出报错信息。



随后对输入进行异种仿射变换Str1[j] = (v10 + v11 \* (Str1[j] - 48)) % 10 + 105

（其中v10=7,v11=3）。



最后得到的密文与Str2进行比对。



Str2具体内容为ippio

编写脚本对仿射变换进行求解，得到flag为10013



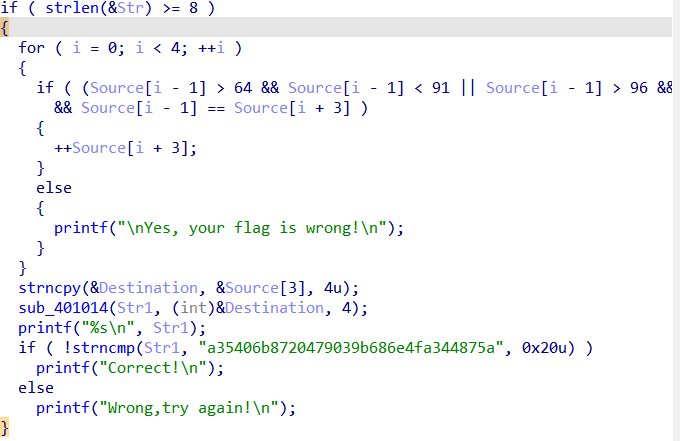
验证flag=10013发现程序通过了。



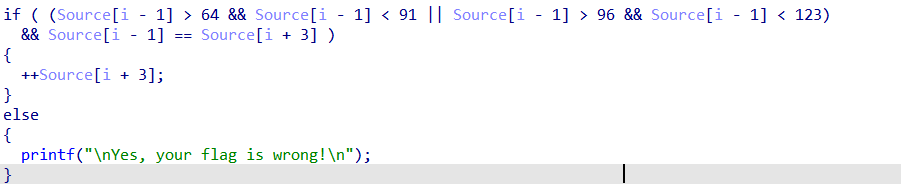
1. CPP4.exe (本题20`)

使用IDA打开CPP4.exe文件，输入明文之后，首先判断输入的明文长度是不是要大于8。

如果大于8则继续，如果小于则输出报错信息。

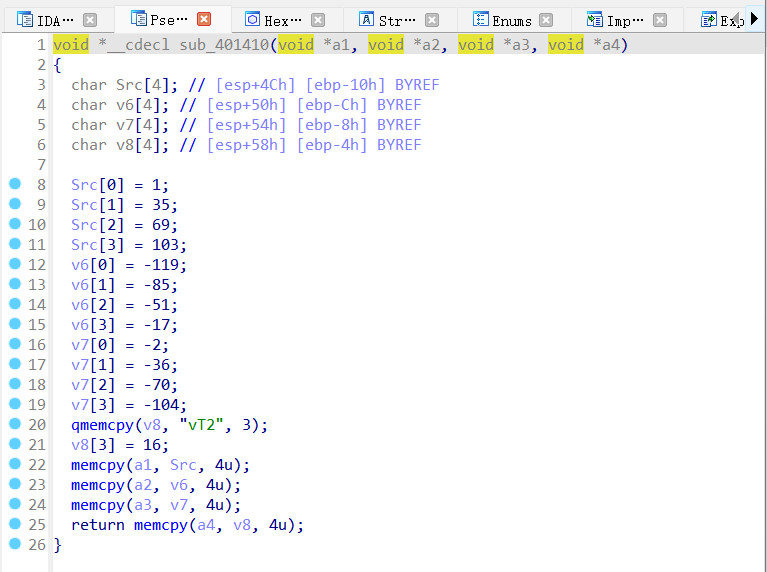


接着对输入的内容进行判断，判断输入是否是大写字母或者小写字母，并且比较第一位和第五位，第二位和第六位，第三位和第七位，第四位和第八位是否相等，如果相等那么后半部分将会加1，否则输出报错信息。

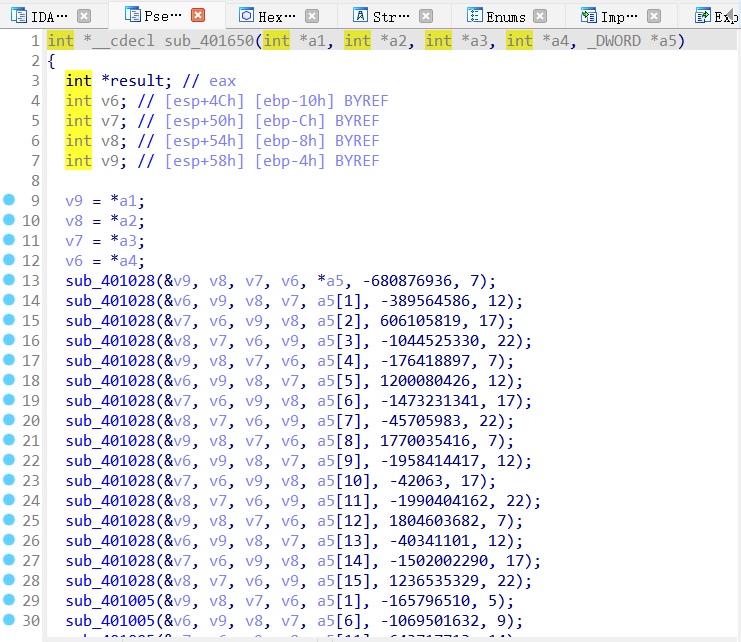


接着取从第五位开始到第八位的字节放入Destination，对Destination进行加密。猜测加密为MD5，原因如下

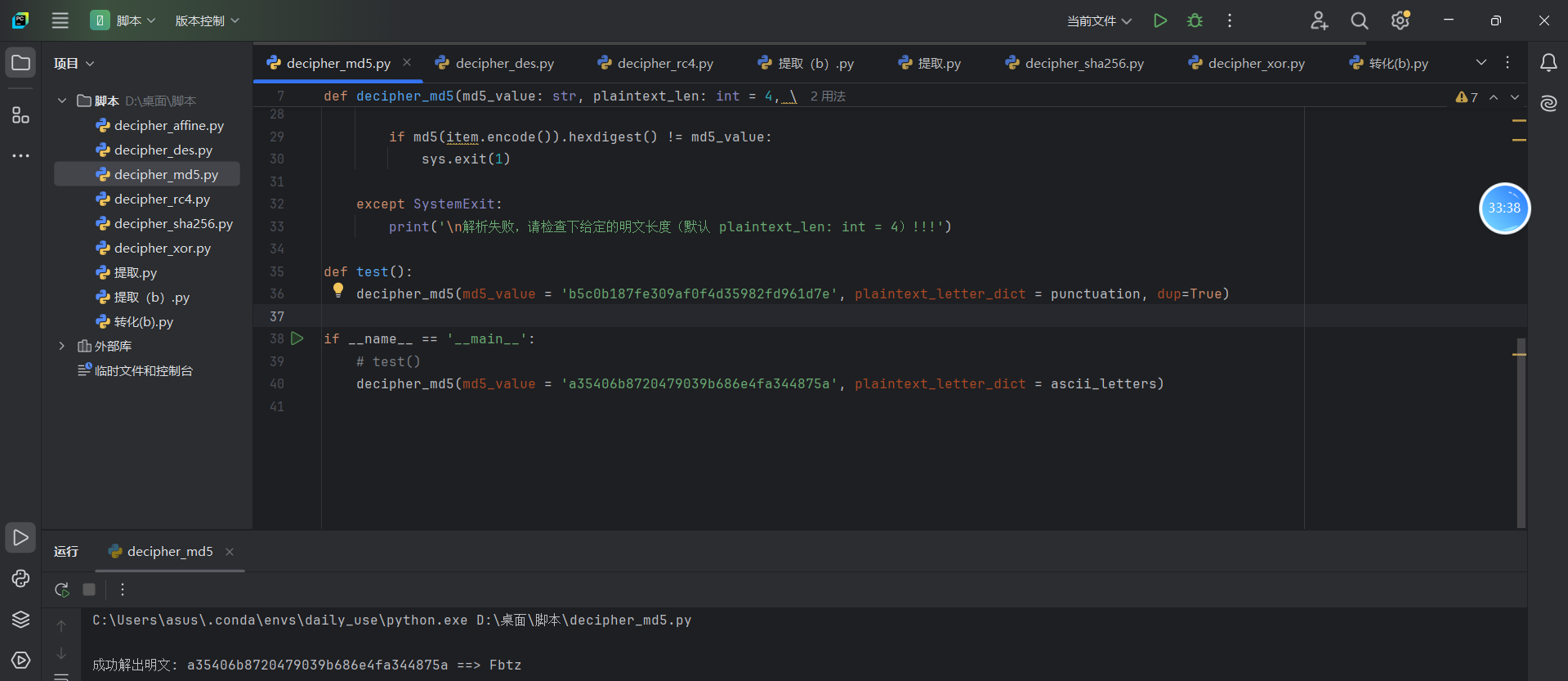
发现函数sub\_401410是对四个变量进行初始化，可以发现4个变量的取值，与MD5中ABCD四个初始变量一致。



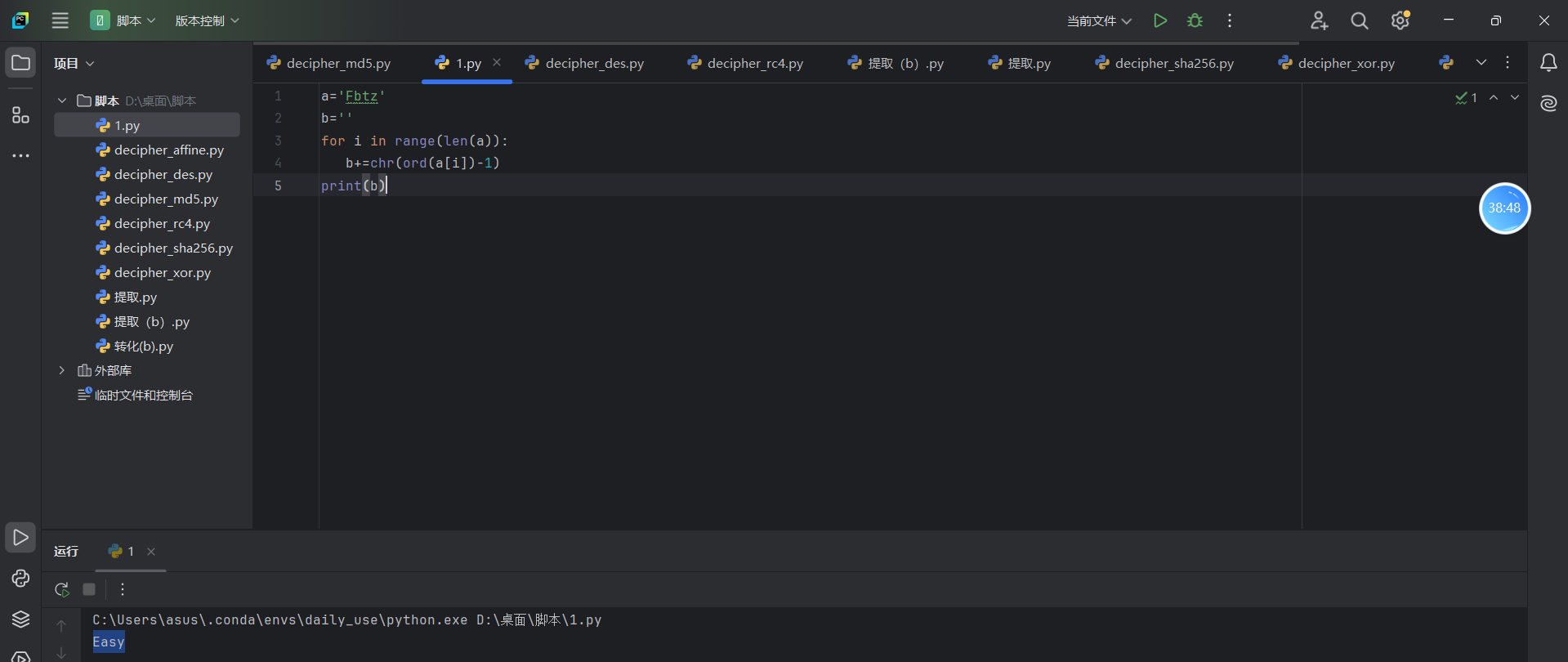
发现函数sub\_401650()是对输入的四个变量进行64轮的轮函数，经过64轮的轮函数后，四个变量分别与初始值相加，函数返回。



且哈希值为a35406b8720479039b686e4fa344875a长度为128比特，据此推断出加密算法为MD5，使用MD5解密程序进行解密，解密得到结果为Fbtz。



对上述Fbtz后四位均进行减1操作得到前四位为Easy。

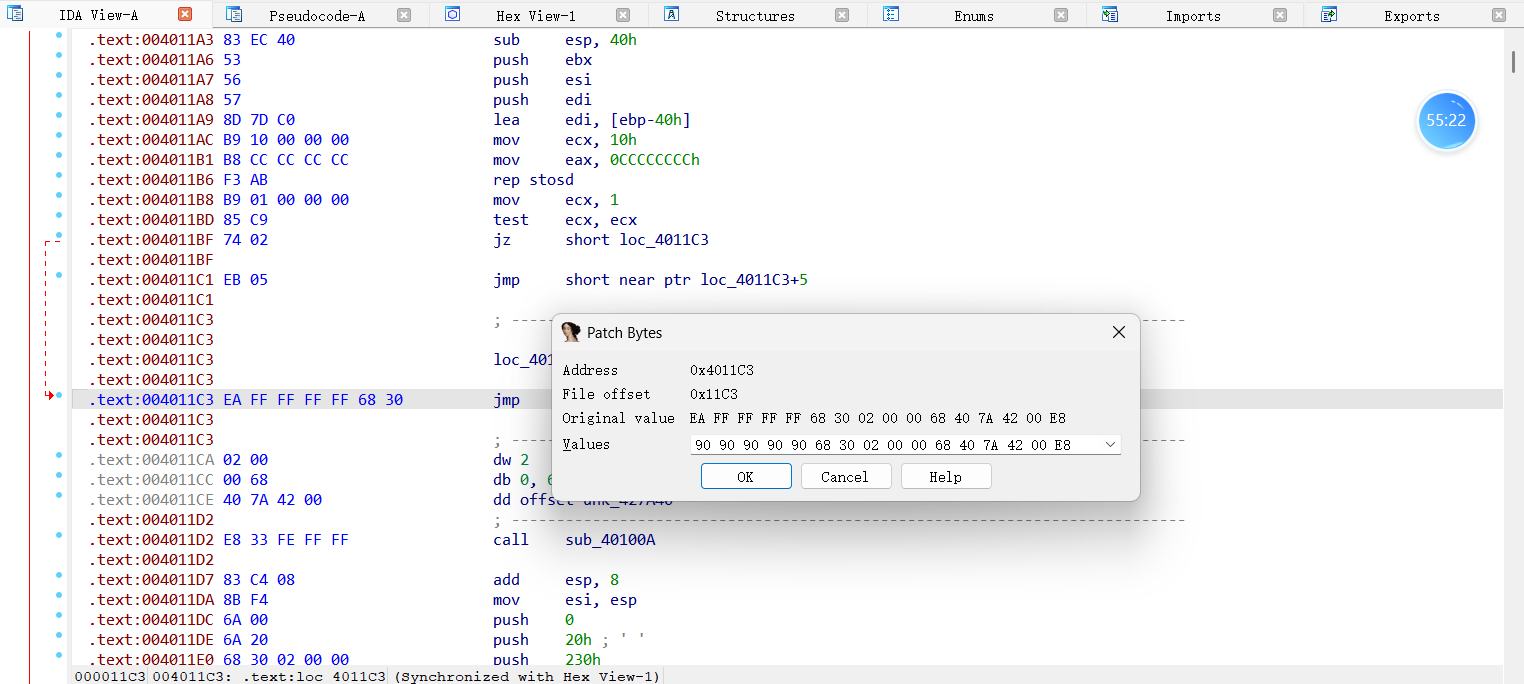


验证flag=EasyEasy发现程序通过了。

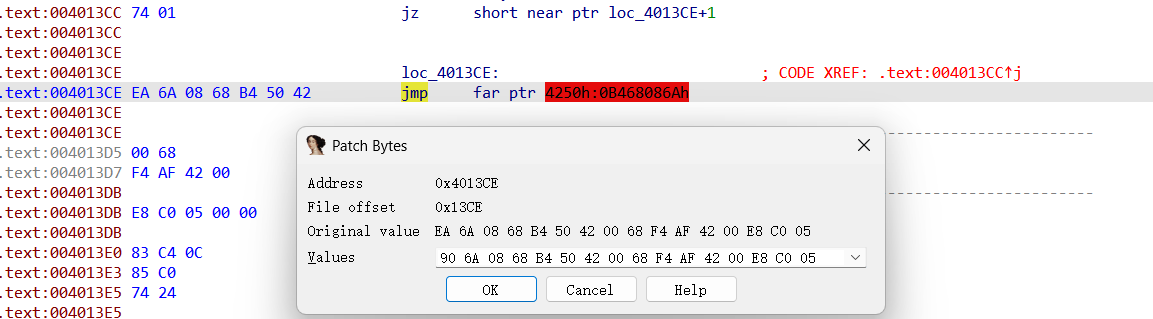


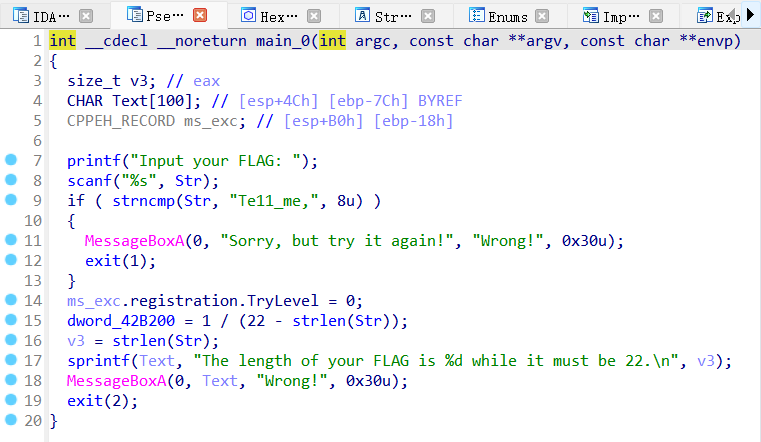
1. CPP5.exe (本题20`)

使用IDA打开CPP5.exe文件，发现反汇编失效，发现第一处花指令为4011C3处，并且根据提示发现4011C1是直接跳转至4011C3+5的，所以中间五个字节都需要nop掉

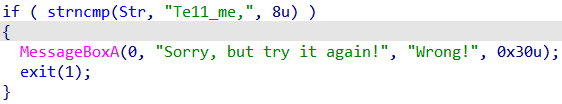


在4013CE处发现第二处花指令，对花指令进行修改，并且根据提示发现4013CE是直接跳转至4013CE+1的，所以只需要nop掉中间一个字节即可。

对程序应用更改，之后重新打开IDA，发现程序恢复正常。



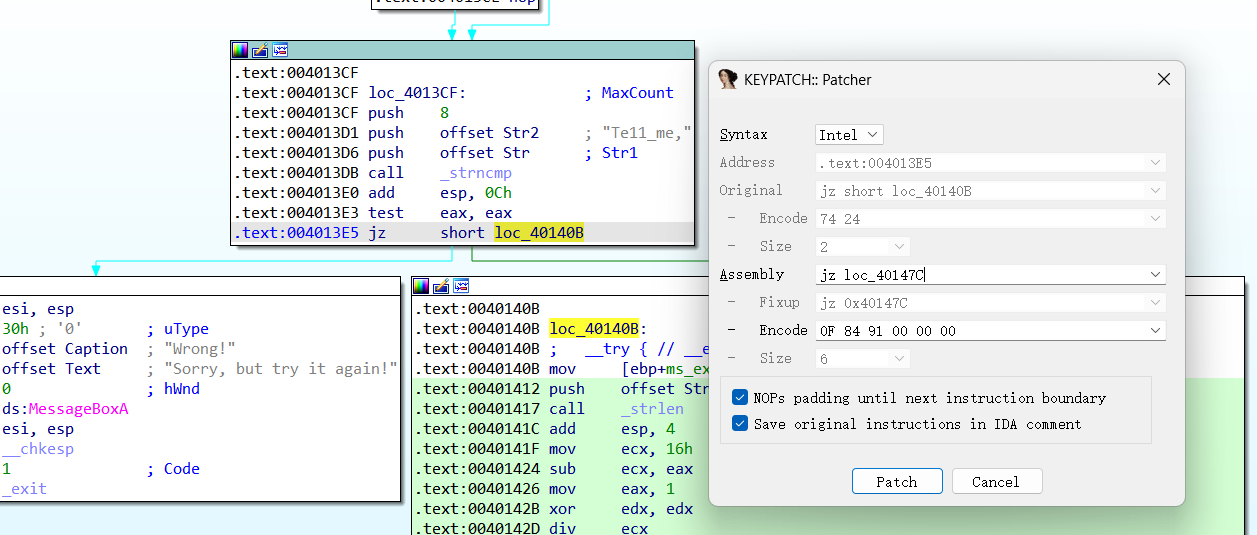
首先程序与Te11\_me,进行比较,若不相等则输出错误信息。



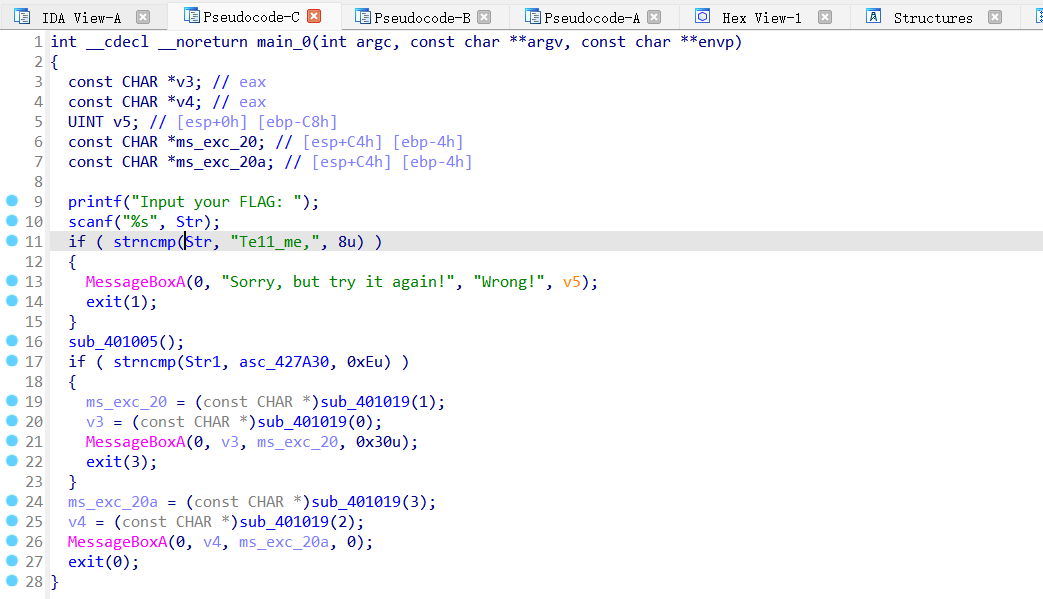
后面对长度进行判断，发现输入明文长度为22时，会出现除零异常，所以可以推断出flag的长度为22。



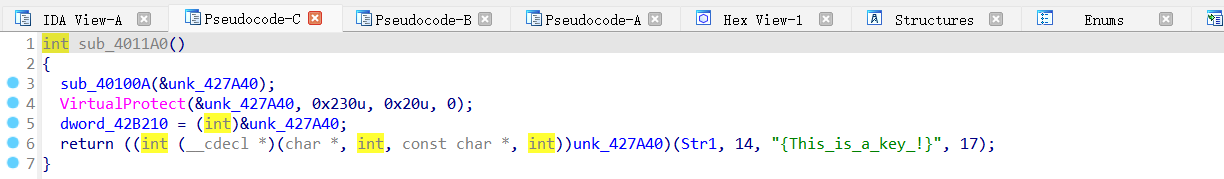
发现异常处理程序在40147C处，使用keypatch修改4013E5处的跳转语句，修改为jz loc\_40147C。



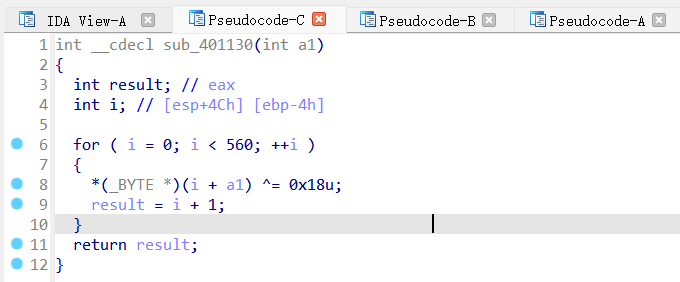
应用keypatch之后查看伪代码发现异常处理程序。



点击查看sub\_401005()。



点击sub\_40100A()->sub\_401130(()，发现该函数实现了让00427A40处的数组与0x18进行异或，长度为560个字节



VirtualProtect函数使00427A40处的560个字节的数组变为可执行属性。



之后调用00427A40处的可执行代码，传入Str1，长度为14，密钥{This\_is\_a\_key\_!}，长度为17。



经过以上分析，发现该程序经过了SMC混淆，所以需要相应脚本进行解密。

#include <idc.idc>

static xor\_setp1(){

auto addr = 0x00427a40; //这里填入要解密字节串的起始地址

auto i = 0;

for(i=0; i<0x230; i++) //循环结束的条件为字节串的结束地址

{

PatchByte(addr+i, Byte(addr+i)^0x18); //异或的数字根据情况修改

}

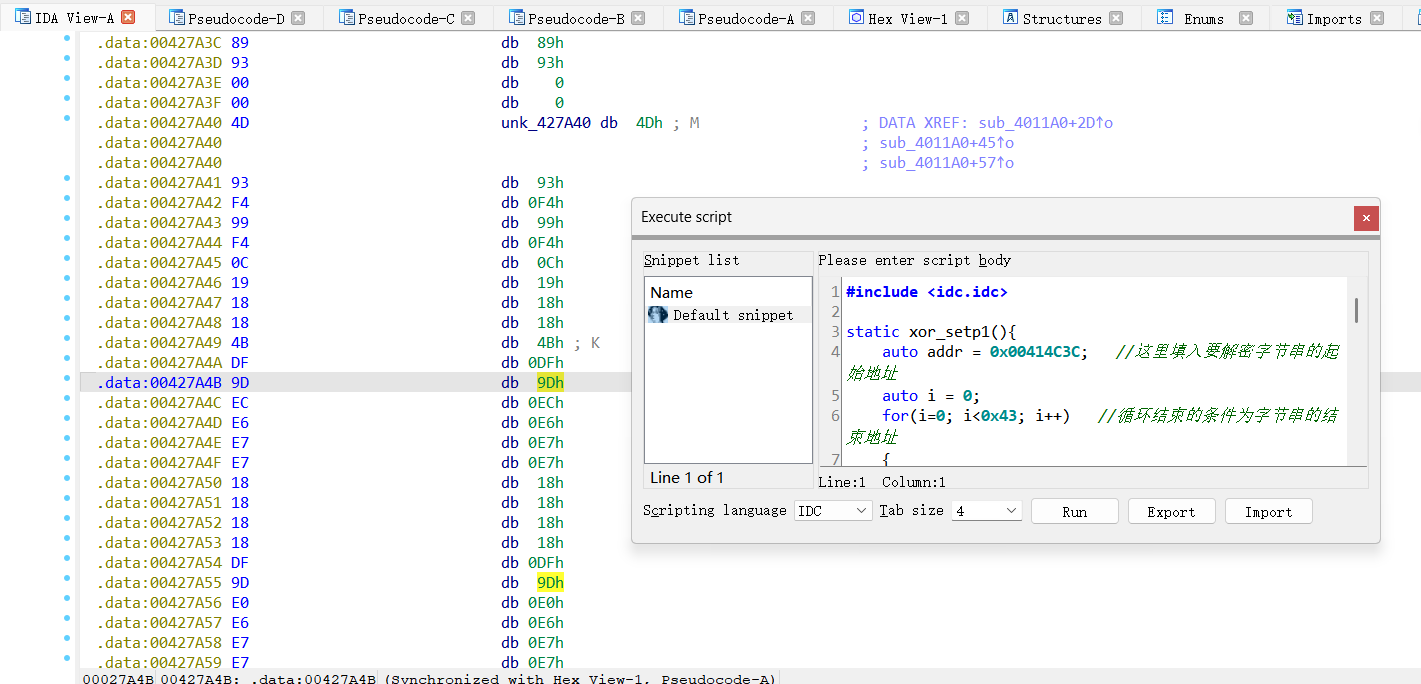
}

static main()

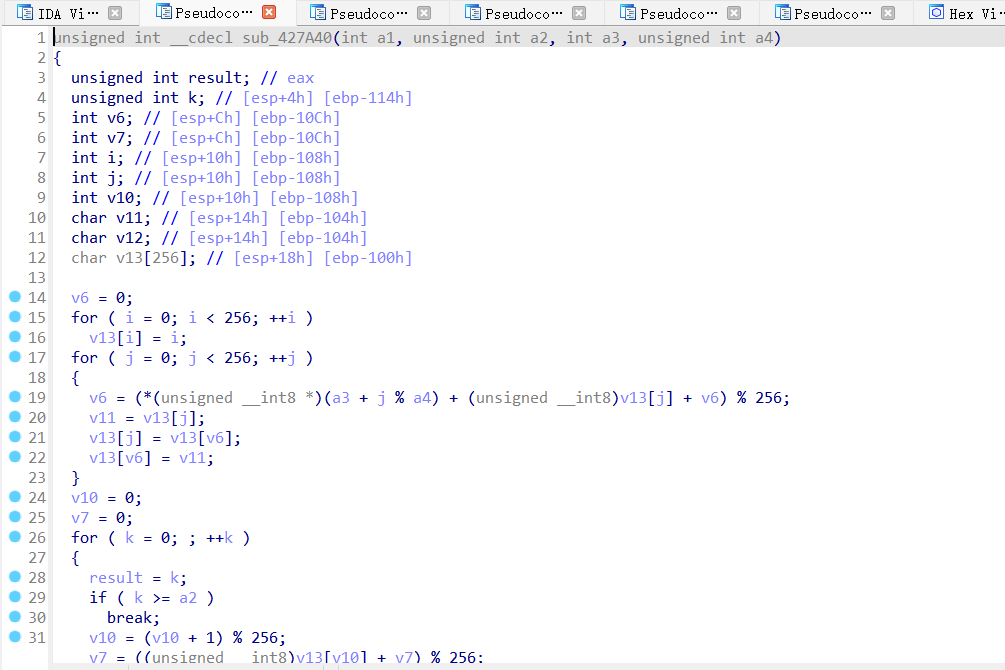
{

xor\_setp1();

}



接着将00427A40，以指令形式进行解释，并且创建函数，得到如图程序。



发现该函数中

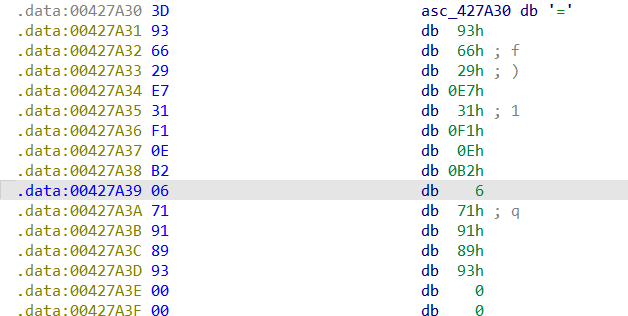
第一个循环是sbox初始化过程，得到数组 v13[] = {0, 1, 2, 3, ……, 255}

第二个循环主要是根据密钥，逐个对数组v13的256个值进行交换操作。

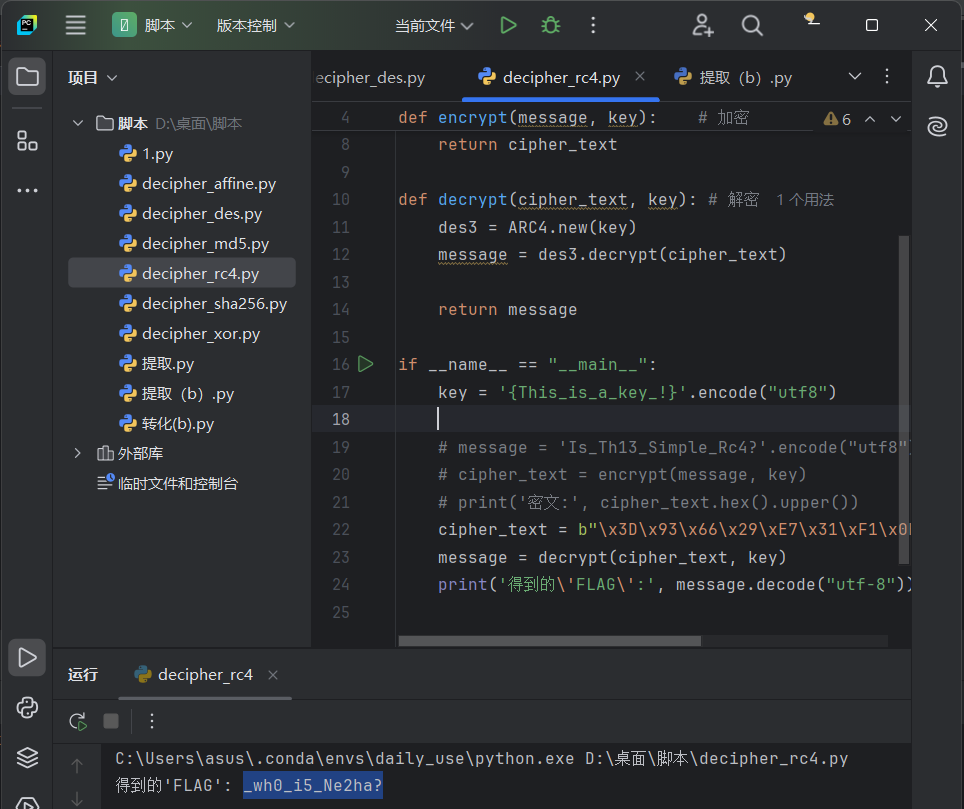
第三个循环其利用两个变量来选择一个S盒中的值作为函数的返回值，并打乱S盒的数据。

根据以上推断出加密算法为RC4，使用RC4解密程序进行解密。

密文为asc\_427A30处存储的数据为b"\x3D\x93\x66\x29\xE7\x31\xF1\x0E\xB2\x06\x71\x91\x89\x93"



密钥为刚才提及的{This\_is\_a\_key\_!}，使用RC4脚本进行解密，解密得到后半部分flag为\_wh0\_i5\_Ne2ha?



综合以上得到最终flag为Te11\_me,\_wh0\_i5\_Ne2ha?

验证flag=Te11\_me,\_wh0\_i5\_Ne2ha?发现程序通过了。

