南开大学

**RE Challenge3**

**（汇编语言与逆向技术实验 9）**

****

**姓名：申宗尚**

**学号：2213924**

**专业：信息安全**

1. **实验目的**
   1. 进一步熟悉静态反汇编工具IDA Freeware；
   2. 熟悉将反汇编代码进行反编译的过程；
   3. 掌握对于反编译伪代码的逆向分析；
   4. 运用熟悉的编程语言，实现简单的脚本编写

**二．实验环境**

1. ida
2. task3.exe
3. task4.exe

**三．实验原理**

1. **task3**
2. 通过IDA Freeware得到task3.exe的反汇编代码，如图1和图2所示。

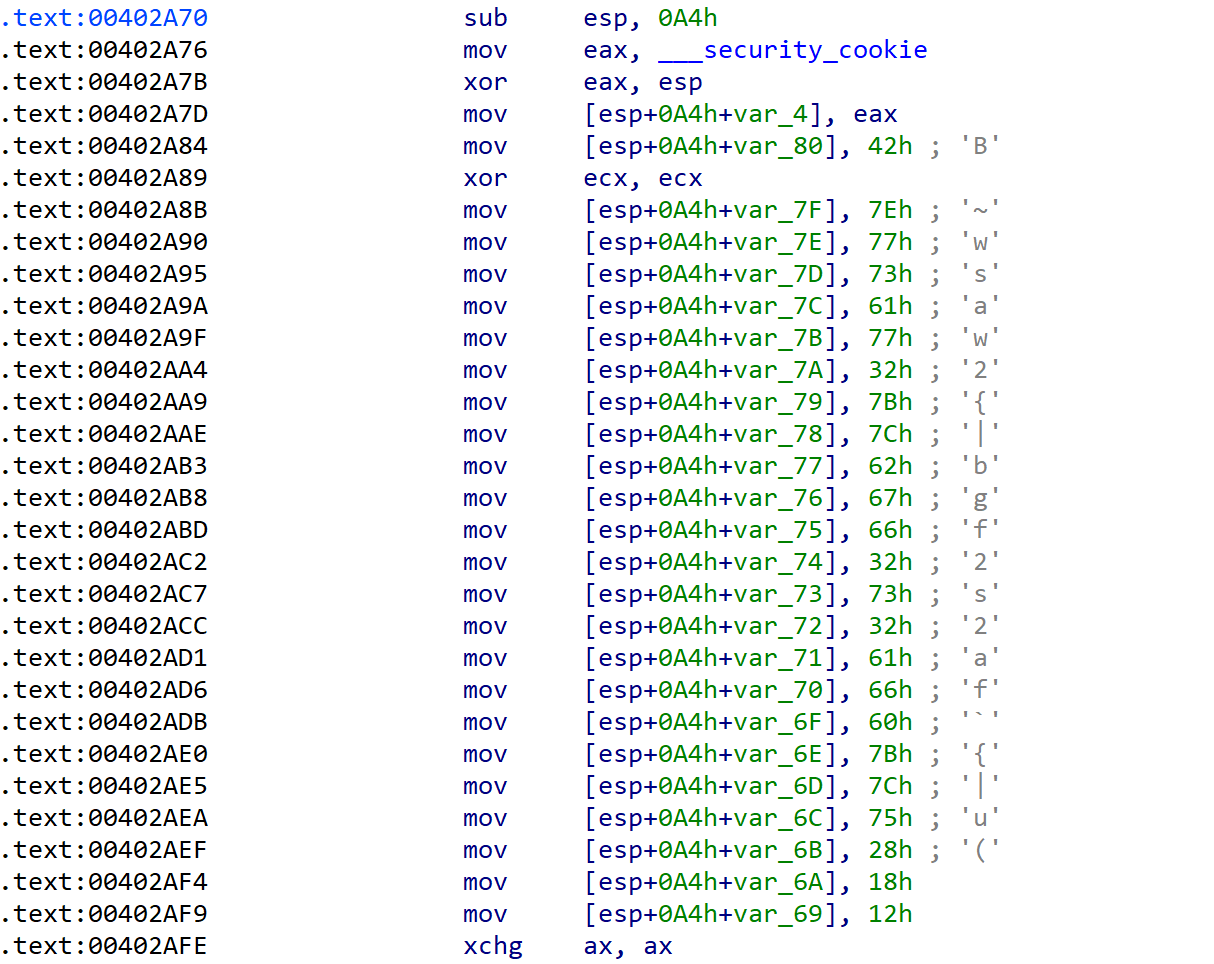


图1 task3.exe的反汇编代码



图 2 task3.exe反汇编代码的图形化显示

1. 使用IDA的反编译功能（F5快捷键）得到伪代码，如图3所示。

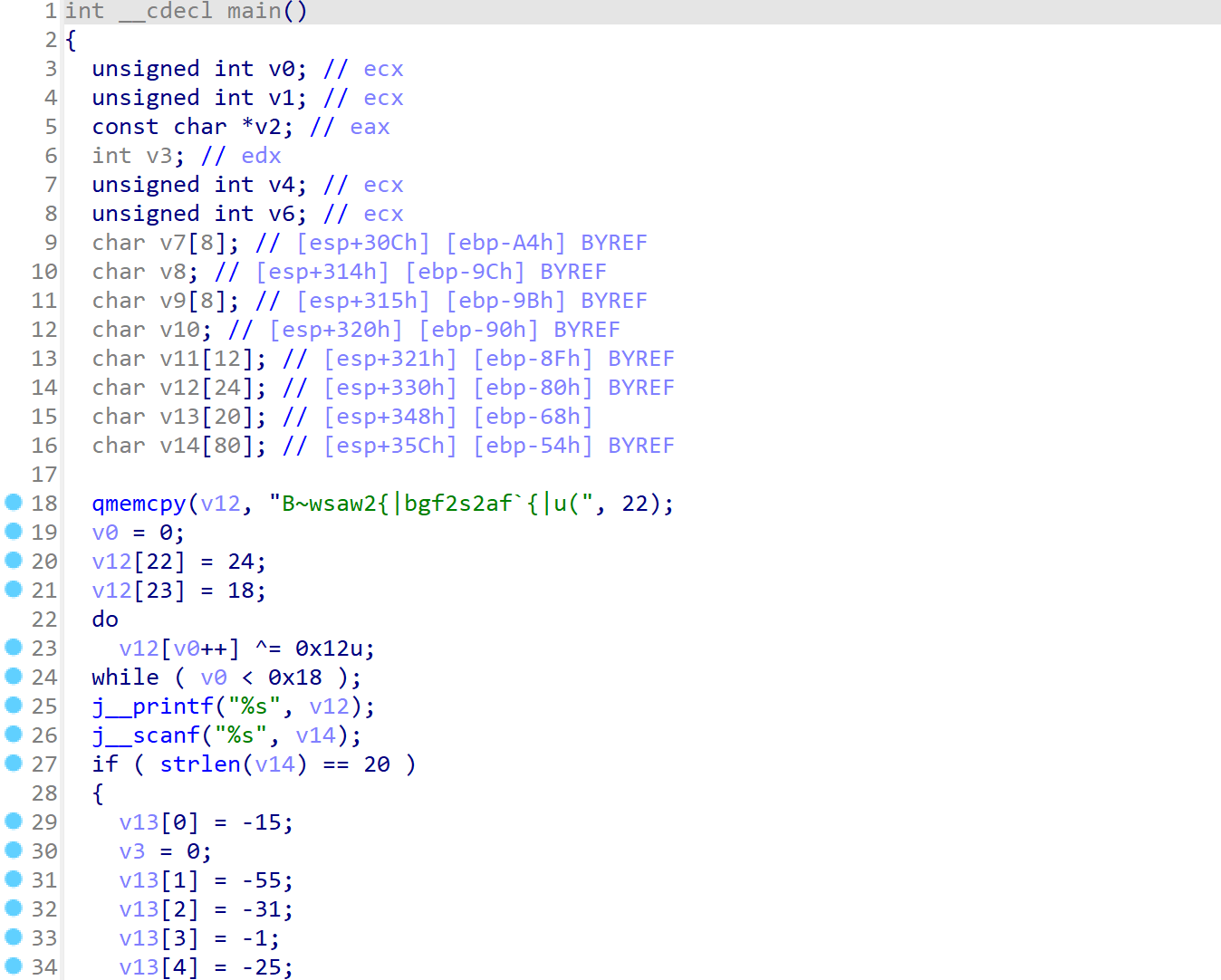


图 3 task3.exe的反编译伪代码

1. 通过对反汇编命令及反编译伪代码的分析，逆向推理出待输入字符串的计算公式
2. 使用熟悉的编程语言（C++、Java、Python等）对待输入字符串进行计算，完成逆向分析挑战。

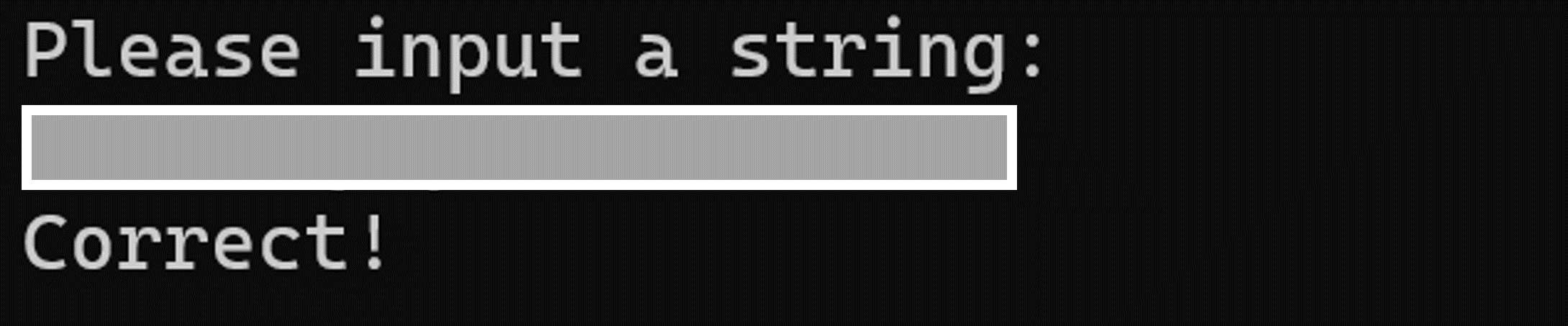


图 4 逆向分析，完成task3练习

1. **task4**
2. 通过IDA Freeware得到task4.exe的反汇编代码，如图5和图6所示。

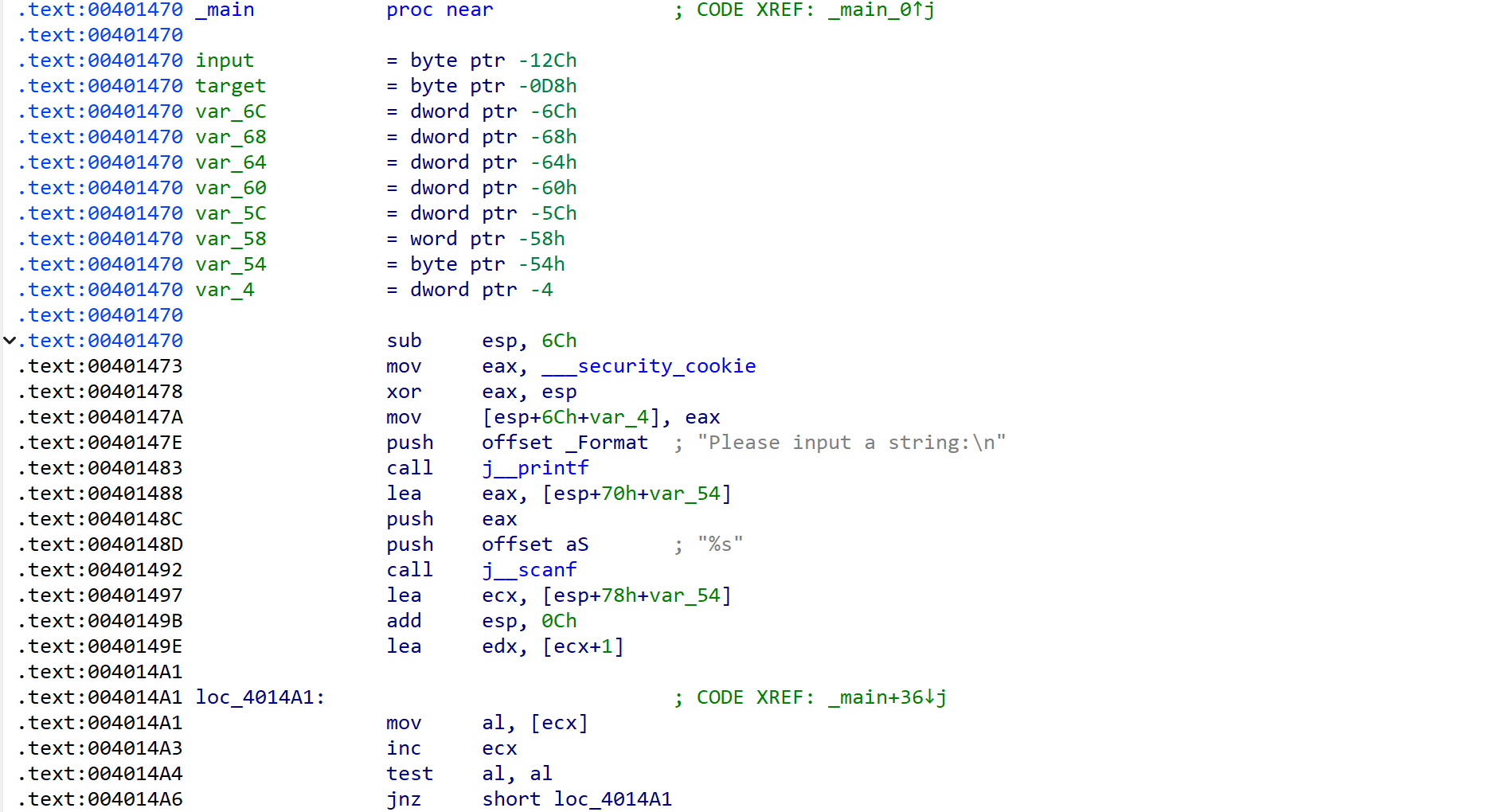


图 5 task4.exe的反汇编代码

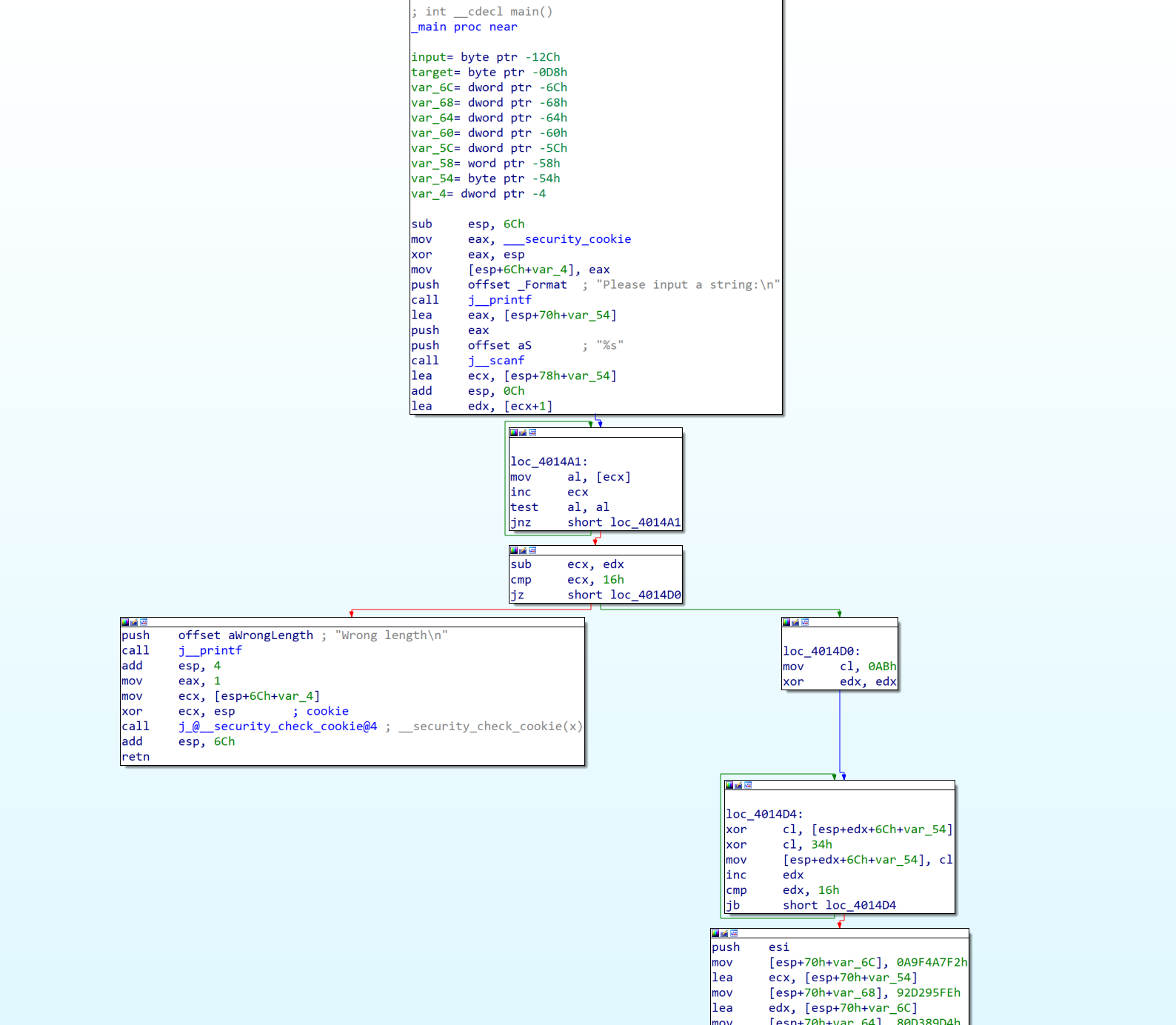


图6 task4.exe反汇编代码的图形化显示

1. 使用IDA的反编译功能（F5快捷键）得到伪代码，如图7所示。右键点击数字对象可实现数制转换。

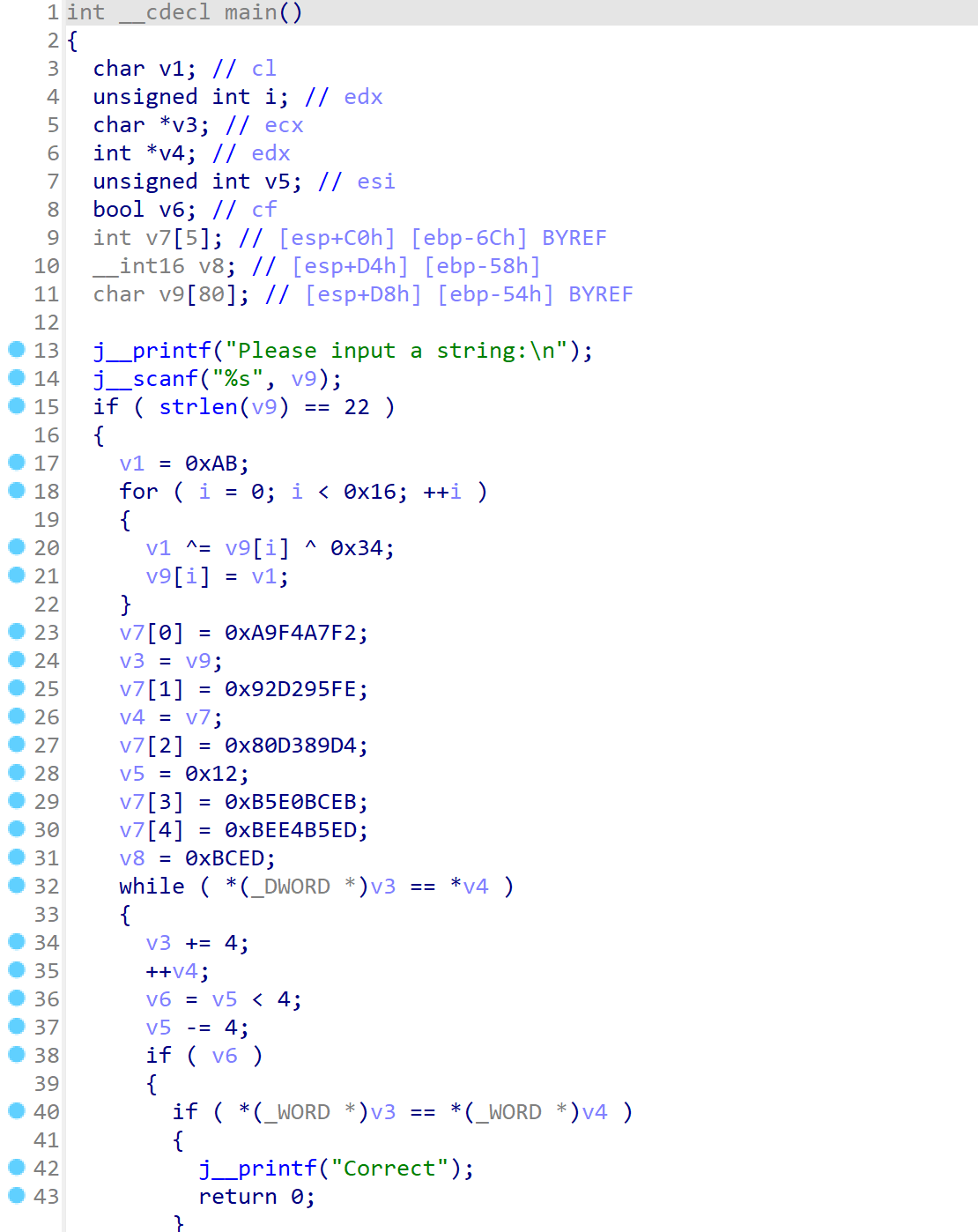


图7 task4.exe的反编译伪代码

1. 通过对反汇编命令及反编译伪代码的分析，逆向推理出待输入字符串的计算公式
2. 使用熟悉的编程语言（C++、Java、Python等）对待输入字符串进行计算，完成逆向分析挑战。

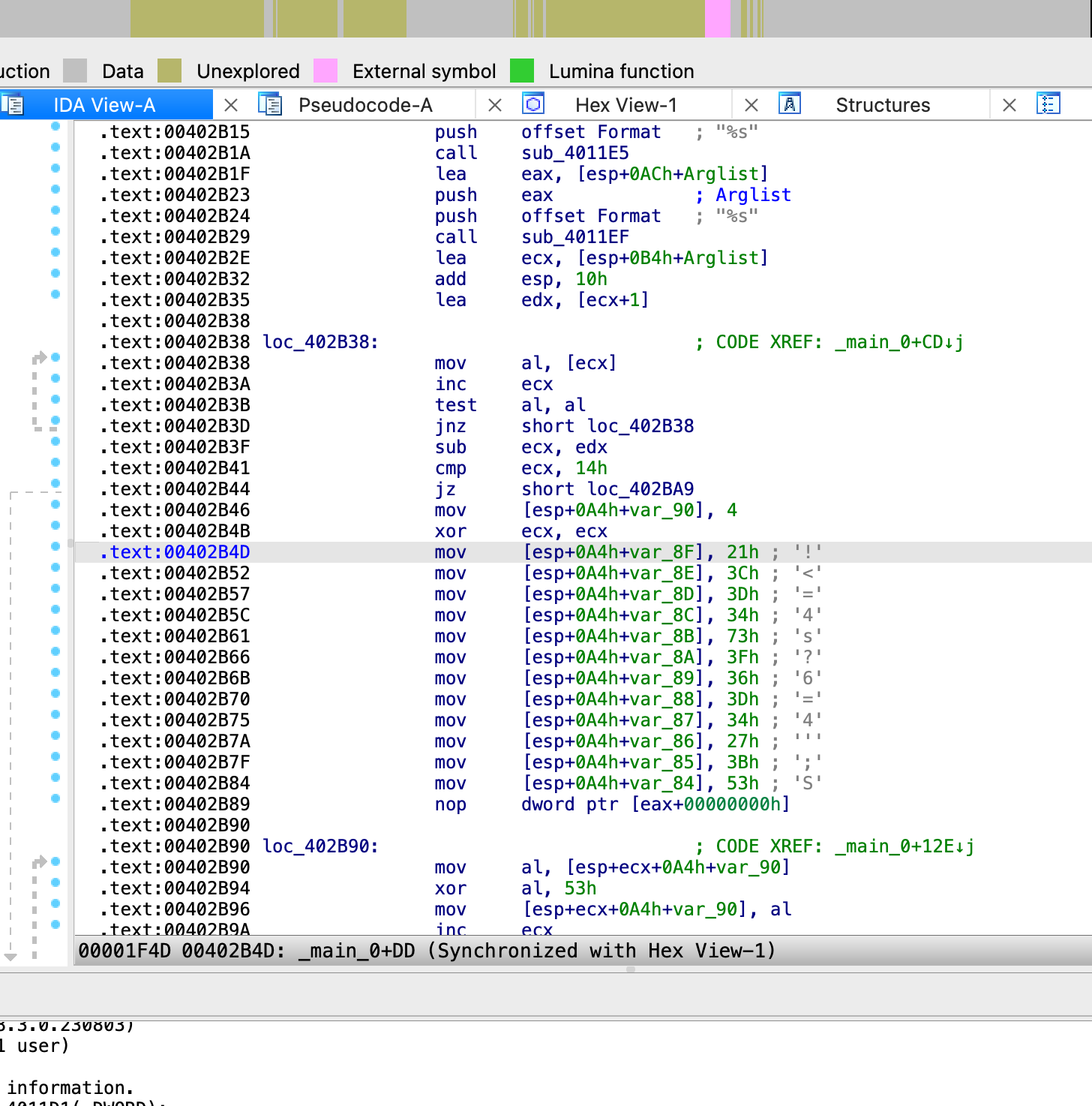


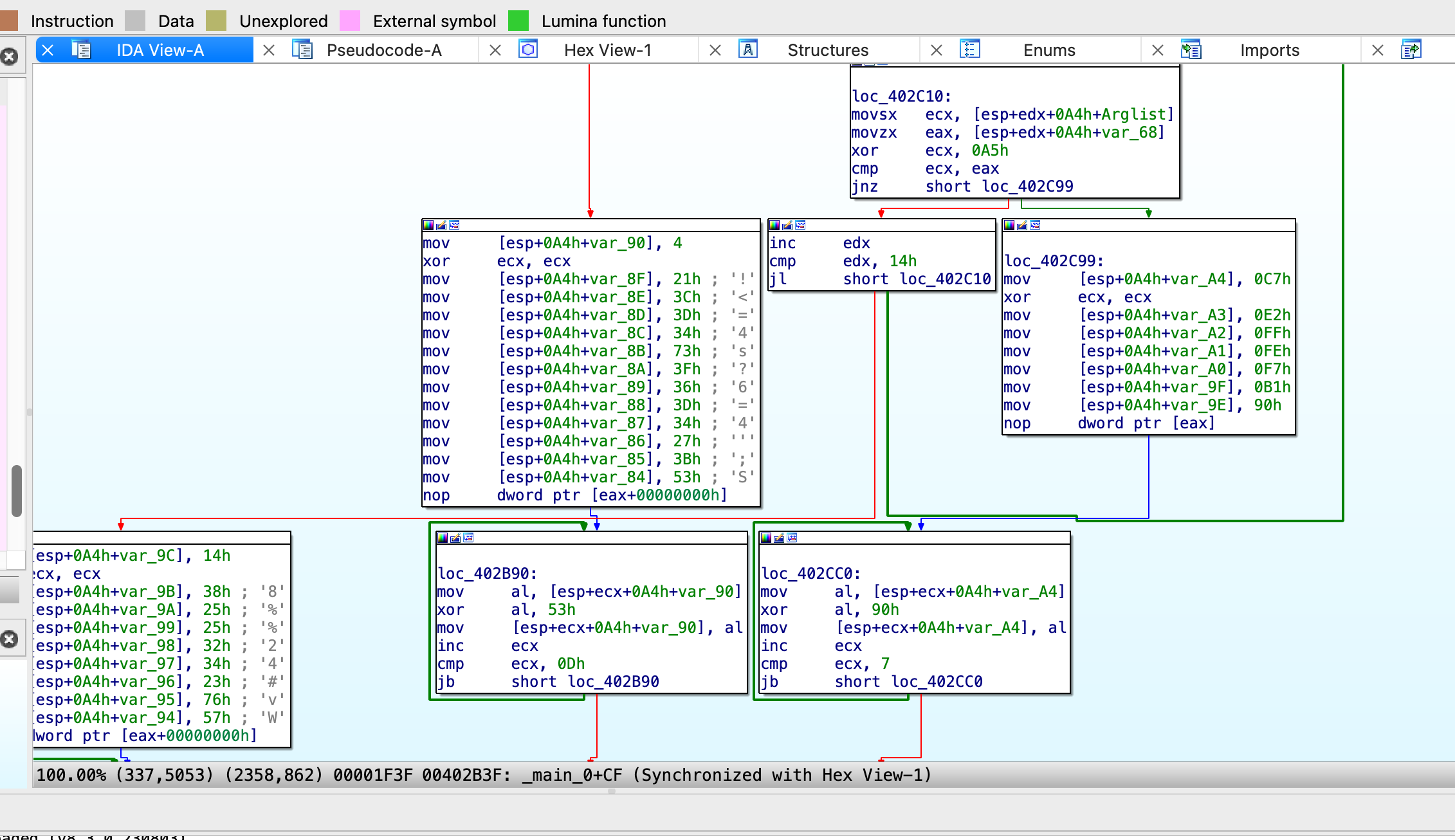
图8 逆向分析，完成task4练习

**四.实验内容**

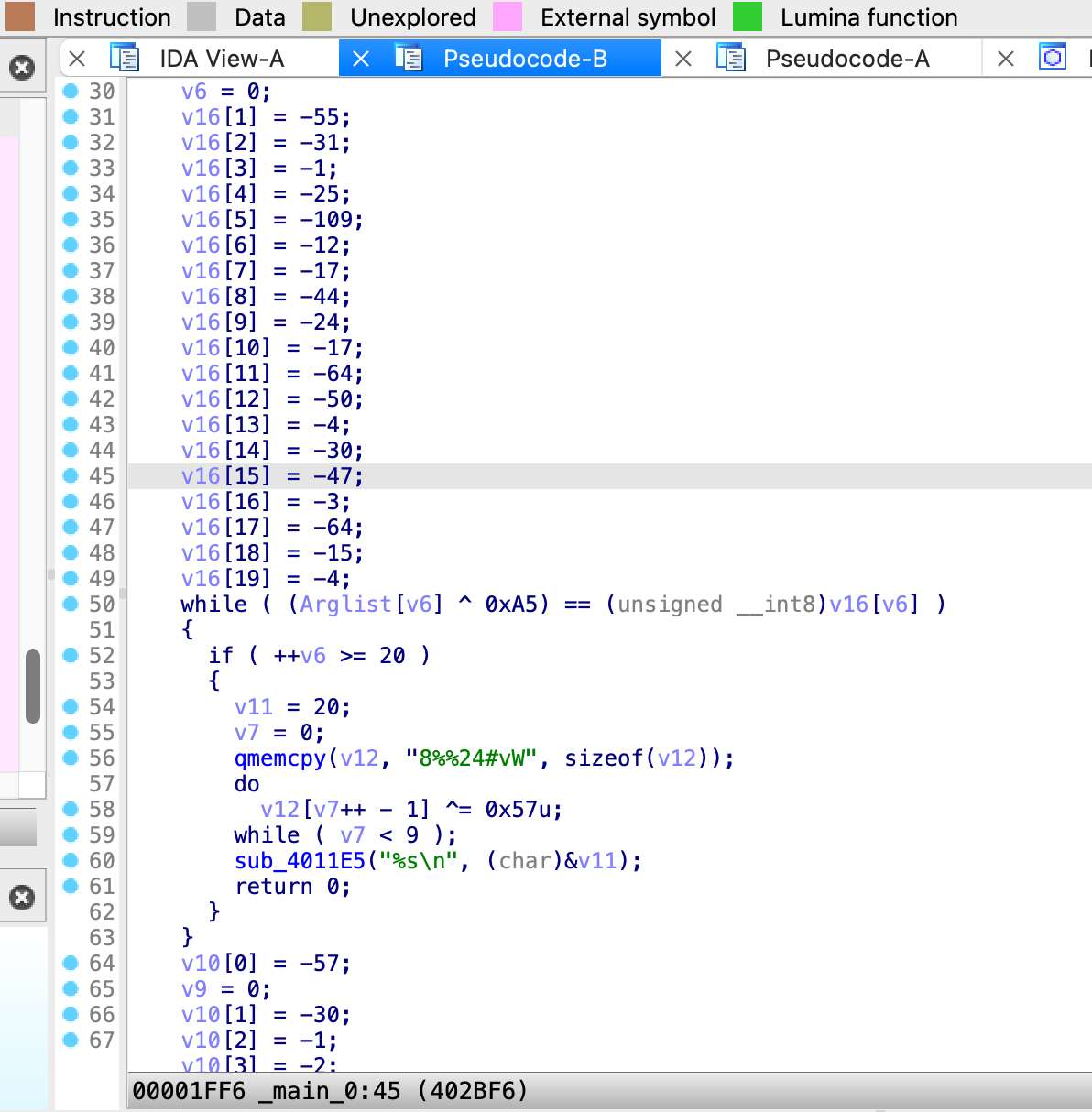
**（1）Task3**

如图，将task3.exe导入ida64 Freeware中，可以得到二进制代码的反汇编代码

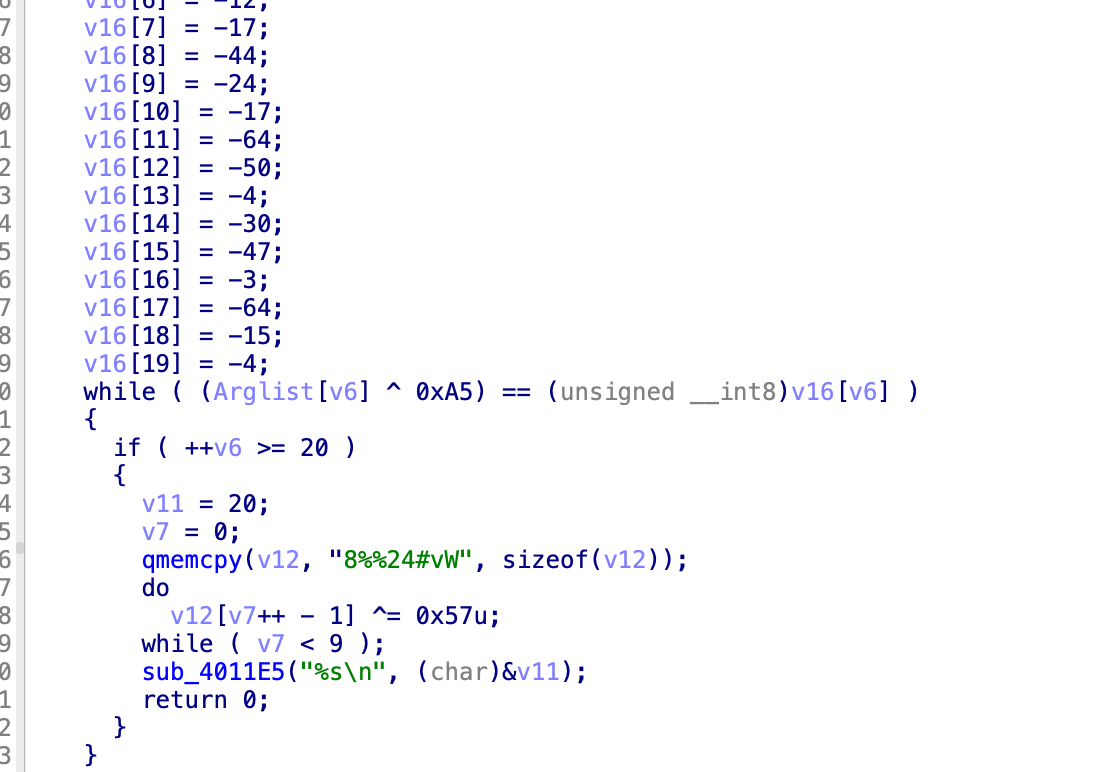




通过ida的反编译功能，得到伪代码：



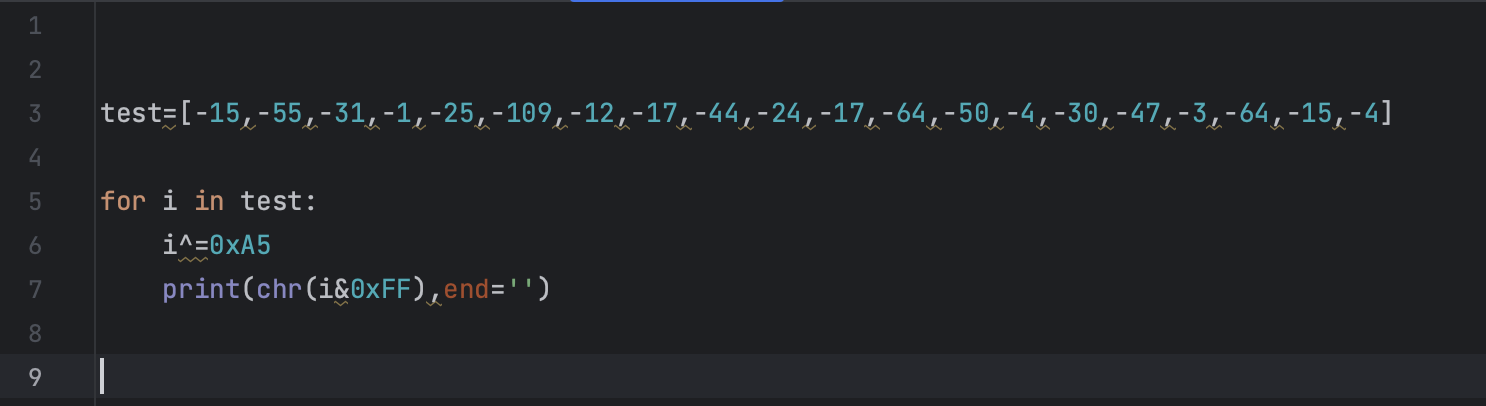
从而，我们可以通过对该反汇编代码的计算过程、数据结构、条件判断、分支结构等信息进行分析，逆向推理出程序的正确输入数据，完成逆向分析挑战。

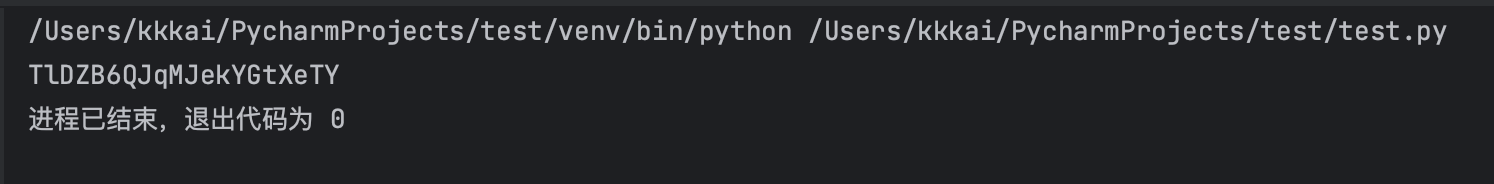


由这段代码可以推出，该程序将输入的字符串逐位与0xA5进行异或运算后，与定义的v16数组逐位进行数据的对比，如果全部正确，则输出“Correct”，否则输出“Wrong”（还有长度错误判断，如果输入的字符串长度不对，也提示错误）

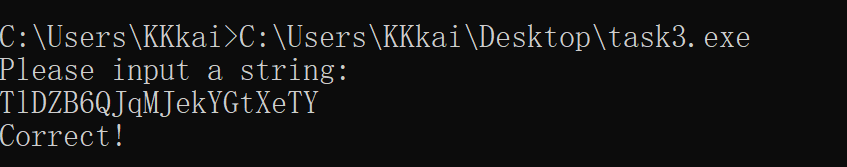
则应该有，输入字符串str[i]^0xA5=test[i]

即str[i]=test[i]^0xA5

从而，编写python代码如图，即可得出正确的输入字符串：

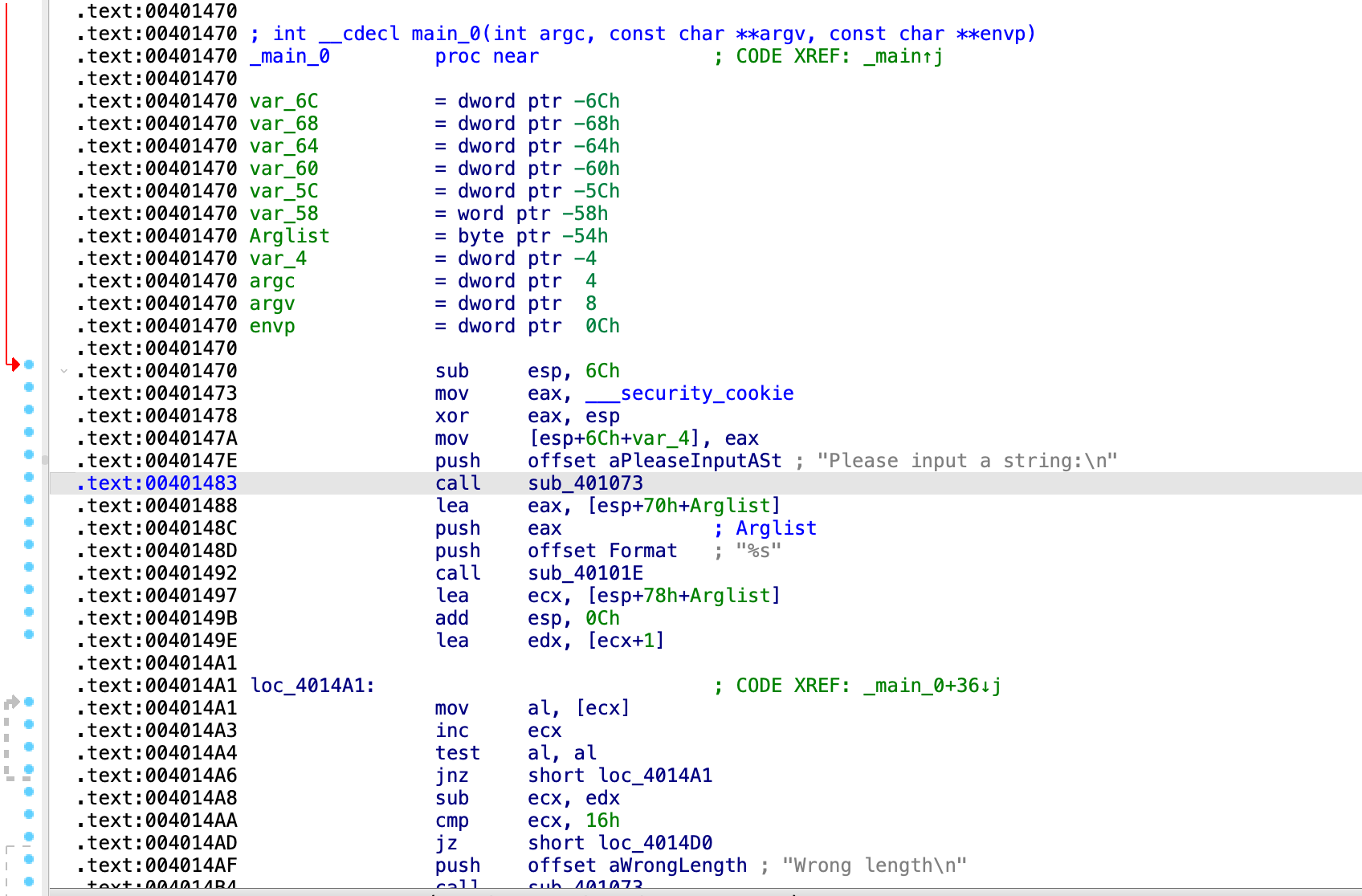


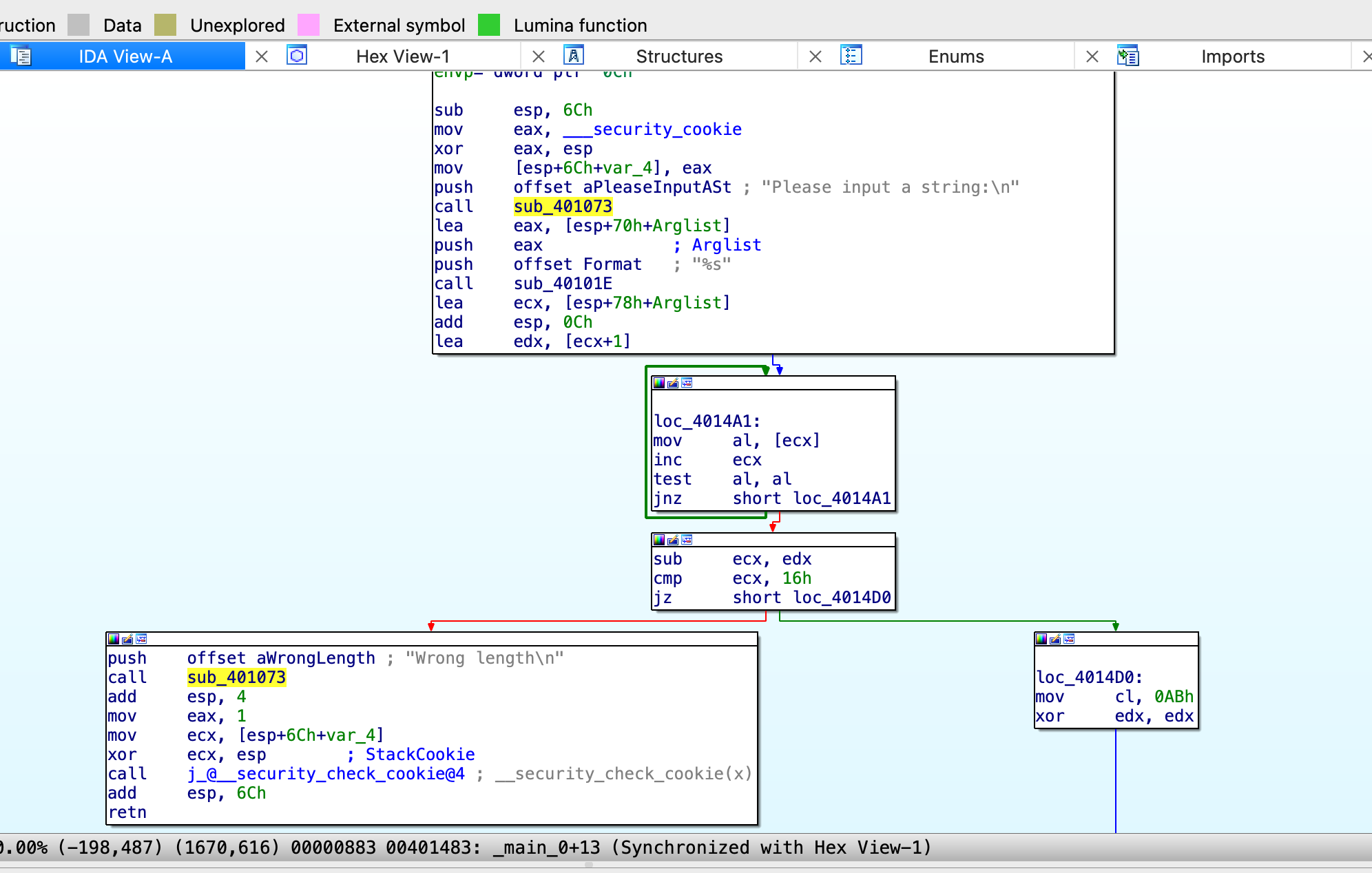
上图为程序运行结果，将结果字符串TlDZB6QJqMJekYGtXeTY输入程序中，得到正确答案：



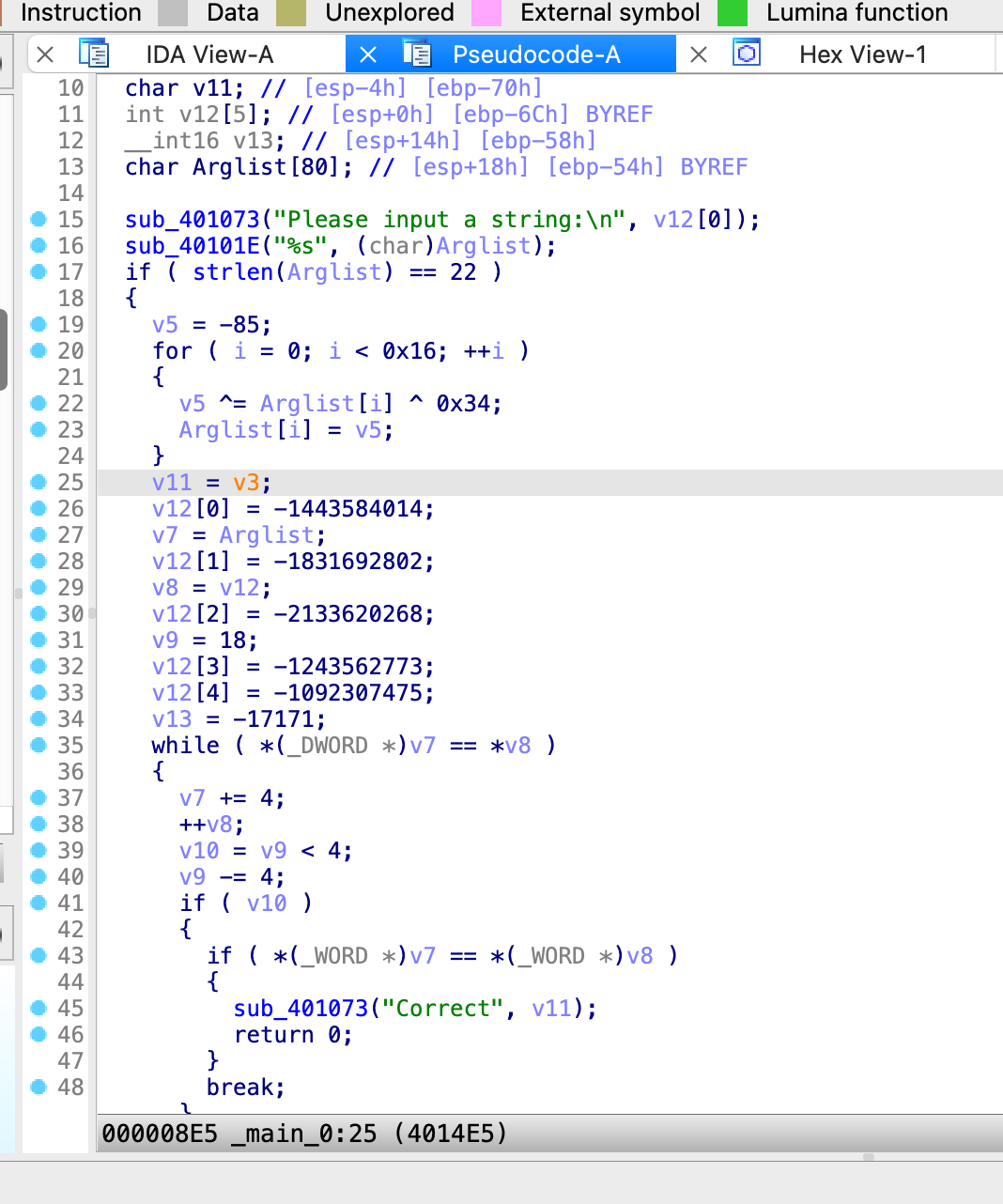
**（1）Task4**

如图，将task4.exe导入ida64 Freeware中，可以得到二进制代码的反汇编代码





使用反汇编功能，将其转为C++代码



分析代码可知，该程序提示用户输入一个字符串，然后对字符串每一位进行异或加密（先初始化了一个v5=-85，随后在每次循环中对每一位Arr[i]=Arr[i]^v5^0x34对异或处理，再更新v5的值为Arr[i]，然后i+1进行下一位的加密）。

随后对加密后的字符串与v12进行比对，如果全部匹配，则输出Correct，否则输出Wrong。

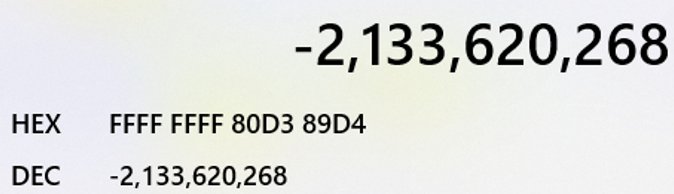
则由上述分析，先将v12中储存的数字分析得：



V12[0]存储为 F2 A7 F4 A9 （小端字节序），且v12定义为DWORD（4字节），则可舍去前4字节的FFFF FFFF。同理，可得



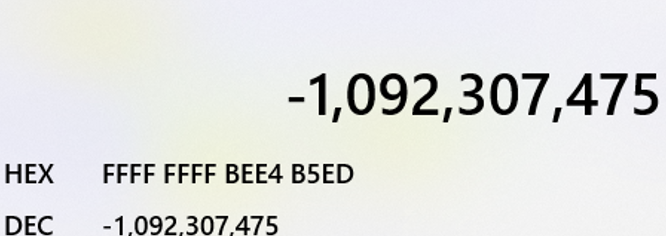
V12[1]存储为 FE 95 D2 92



V12[2]存储为 D4 89 D3 80



V12[3]存储为 EB BC E0 B5



V12[4]存储为 ED B5 E4 BE

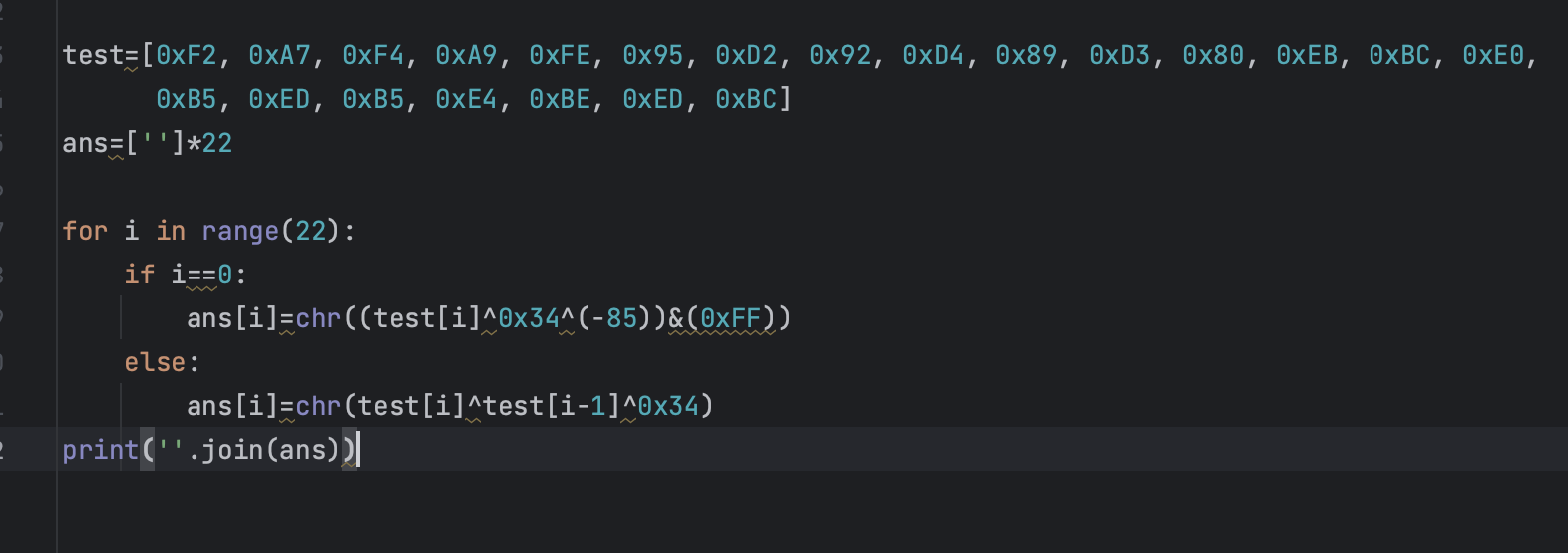


对应的最后一个WORD类型存储为 ED BC

则可知，作为DWORD存储的最后数据应该为：

F2 A7 F4 A9 FE 95 D2 92 D4 89 D3 80 EB BC E0 B5 ED B5 E4 BE ED BC

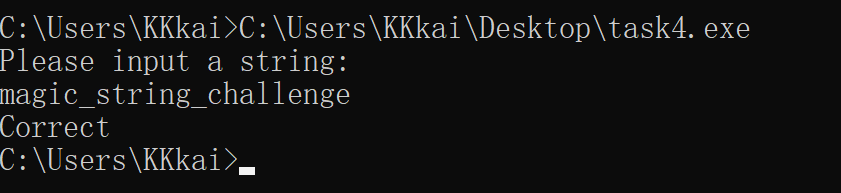
以F2为例，由Arr[0]^v5^0x34=F2得，Arr[0]=F2^0x34^v5,再更新v5=Arr[0]，通过相同的式子计算后面的Arr部分，python代码实现如下：



运行程序，得出答案



运行task4.exe程序，输入得到最终结果



**五.附件（python代码）**

**Task1:**

test=[-15,-55,-31,-1,-25,-109,-12,-17,-44,-24,-17,-64,-50,-4,-30,-47,-3,-64,-15,-4]  
  
for i in test:  
 i^=0xA5  
 print(chr(i&0xFF),end='')

**Task4:**

test=[0xF2, 0xA7, 0xF4, 0xA9, 0xFE, 0x95, 0xD2, 0x92, 0xD4, 0x89, 0xD3, 0x80, 0xEB, 0xBC, 0xE0,  
 0xB5, 0xED, 0xB5, 0xE4, 0xBE, 0xED, 0xBC]  
ans=['']\*22  
  
for i in range(22):  
 if i==0:  
 ans[i]=chr((test[i]^0x34^(-85))&(0xFF))  
 else:  
 ans[i]=chr(test[i]^test[i-1]^0x34)  
print(''.join(ans))