

**汇编语言与逆向技术**

**课程实验报告**

**实验八：Reverse Engineering Exercises – Simple**

****

学 院 网络空间安全

专 业 信息安全

学 号 2212998

姓 名 胡博浩

班 级­­­­ 2022级信息安全

1. **实验目的**
   1. 熟悉静态反汇编工具IDA Freeware；
   2. 熟悉反汇编代码的逆向分析过程；
   3. 掌握反汇编语言中的数学计算、数据结构、条件判断、分支结构的识别和逆向分析
2. **实验原理**
3. **task1**
4. 通过IDA Freeware得到task1.exe的反汇编代码，如图1和图2所示。

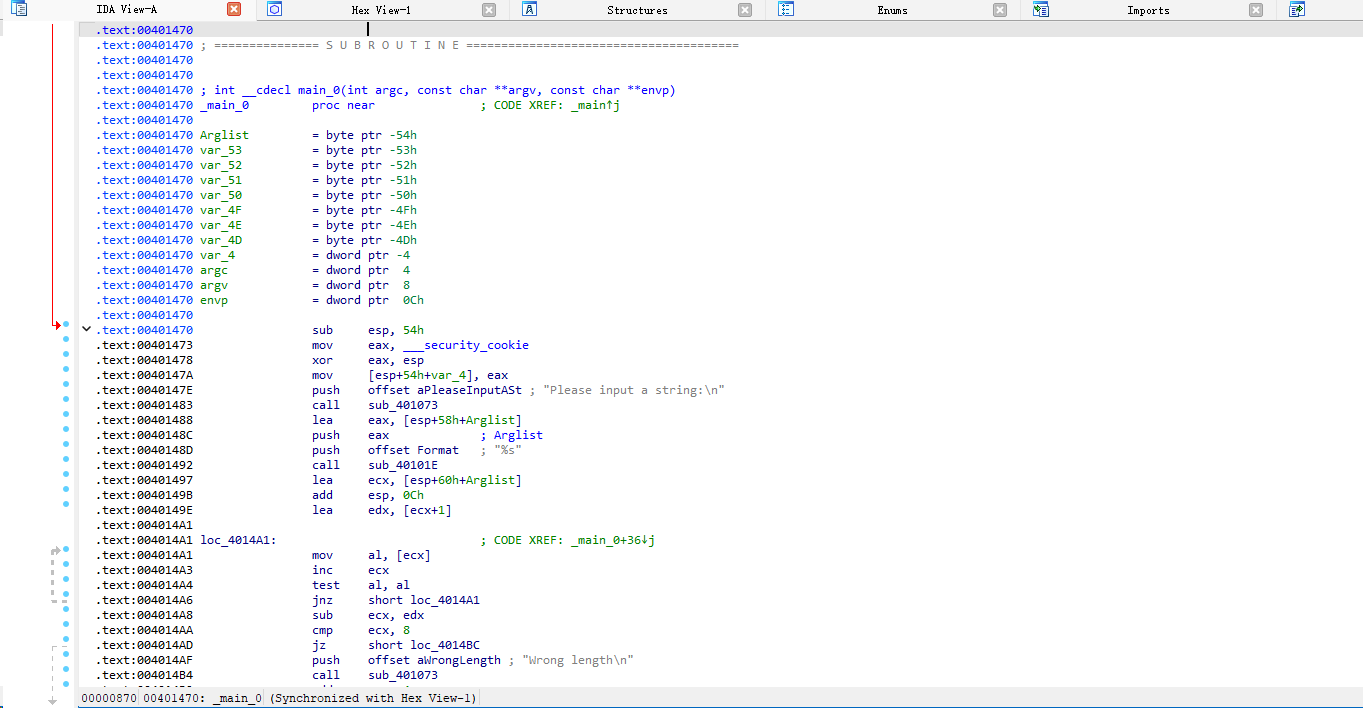


图 1 task1.exe的反汇编代码

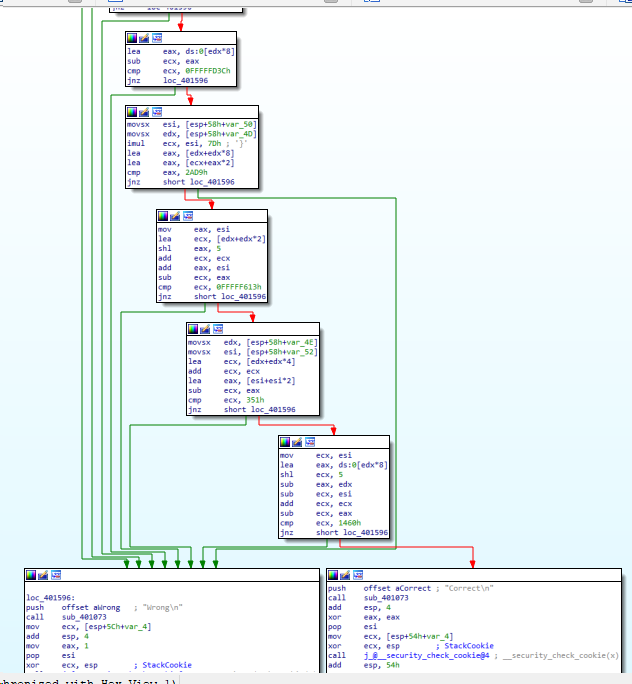


图 2 task1.exe反汇编代码的图形化显示

1. 对反汇编代码和计算过程、条件判断、分支结构等信息进行分析，逆向推出程序的正确输入数据，完成逆向分析挑战。

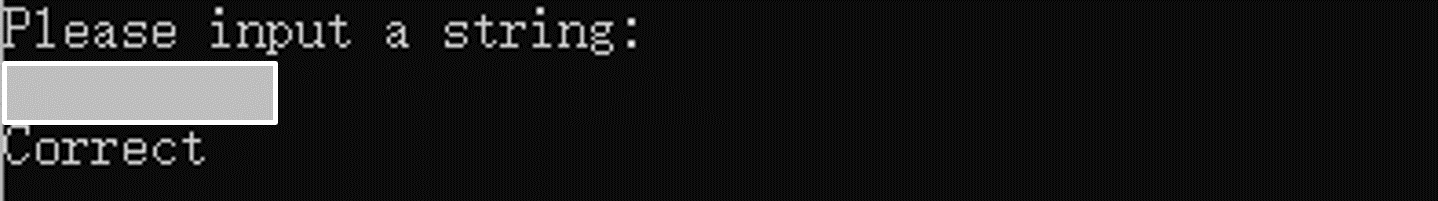


图 3 逆向分析，完成task1练习

1. **task2**
2. 通过IDA Freeware得到task2.exe的反汇编代码，如图4和图5所示。

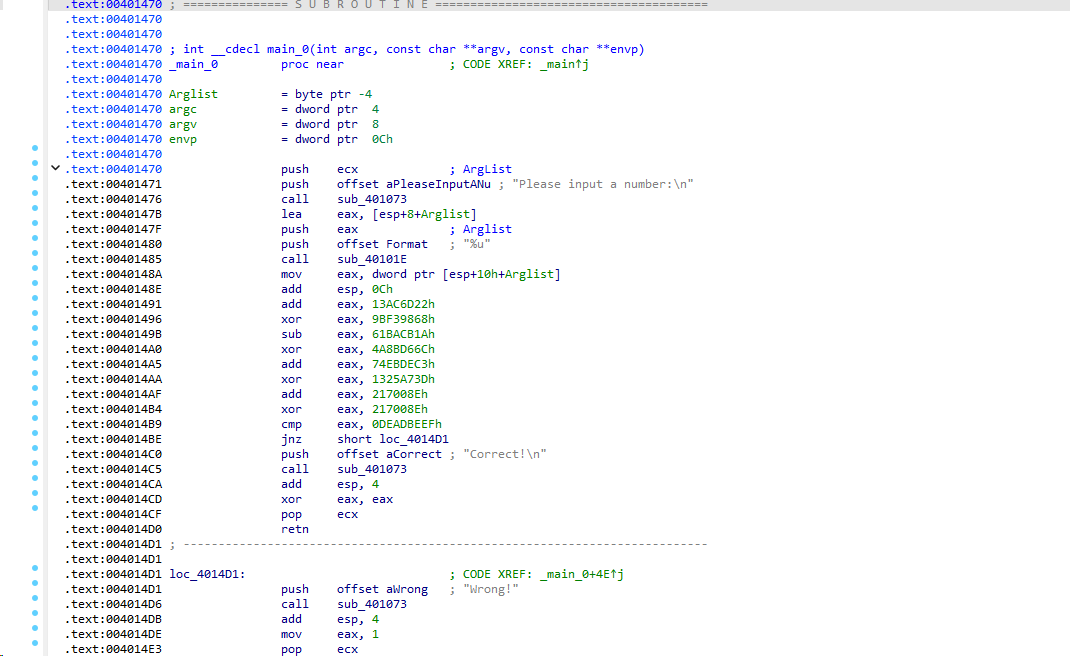


图4 task2.exe的反汇编代码

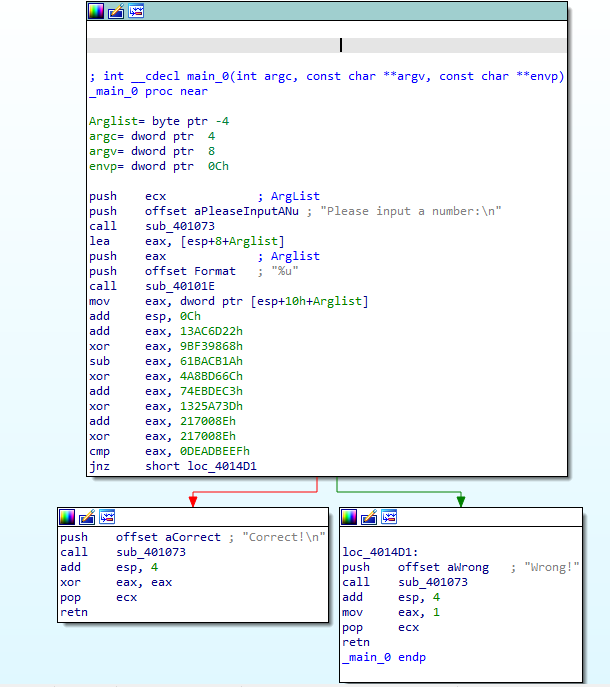


图 5 task2.exe反汇编代码的图形化显示

1. 对反汇编代码的计算过程、条件判断、分支结构等信息进行分析，逆向推出程序的正确输入数据，完成逆向分析挑战。

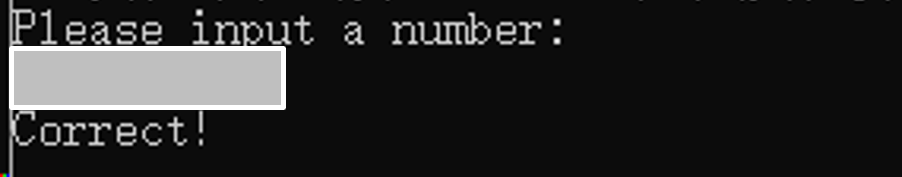
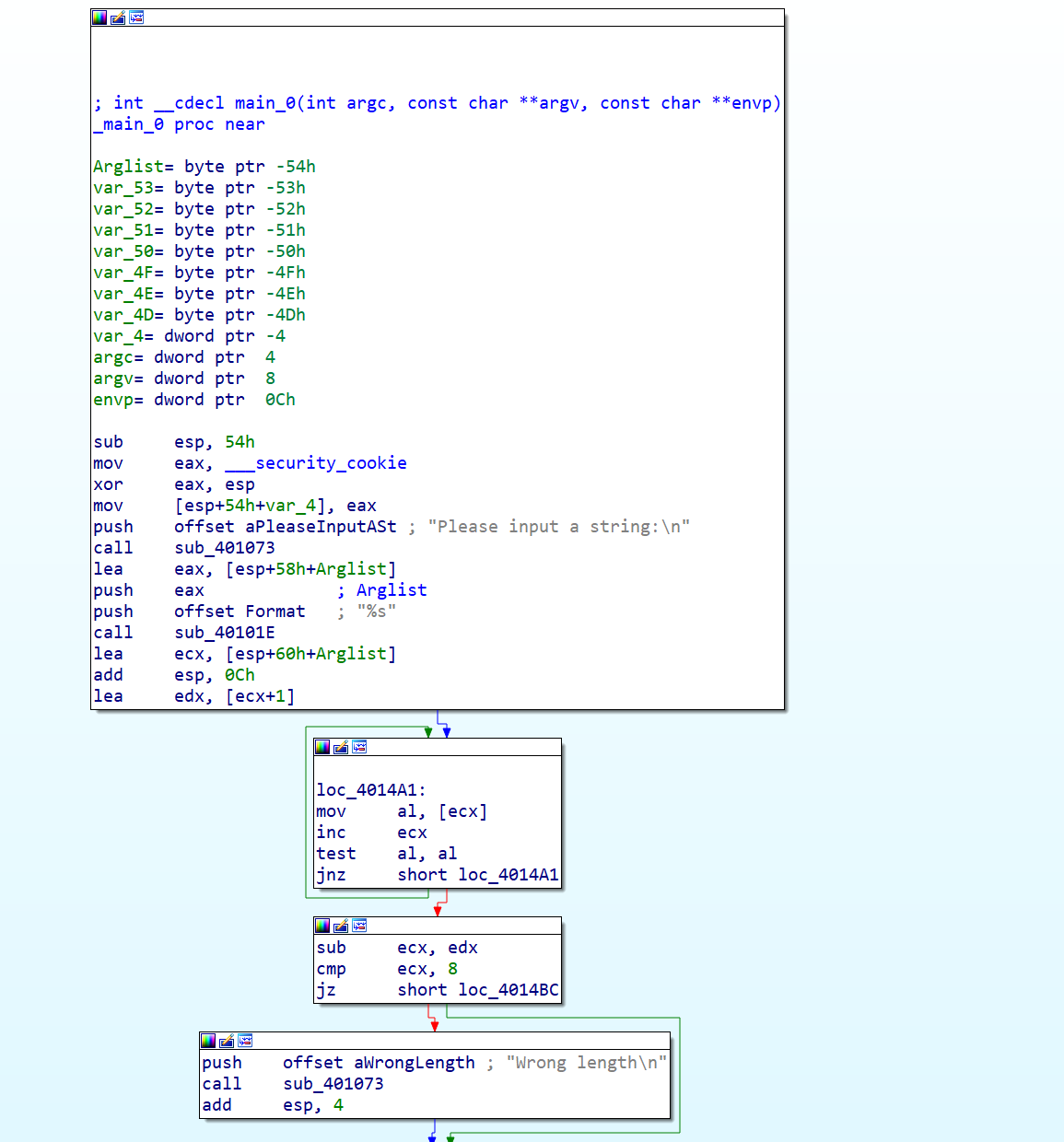


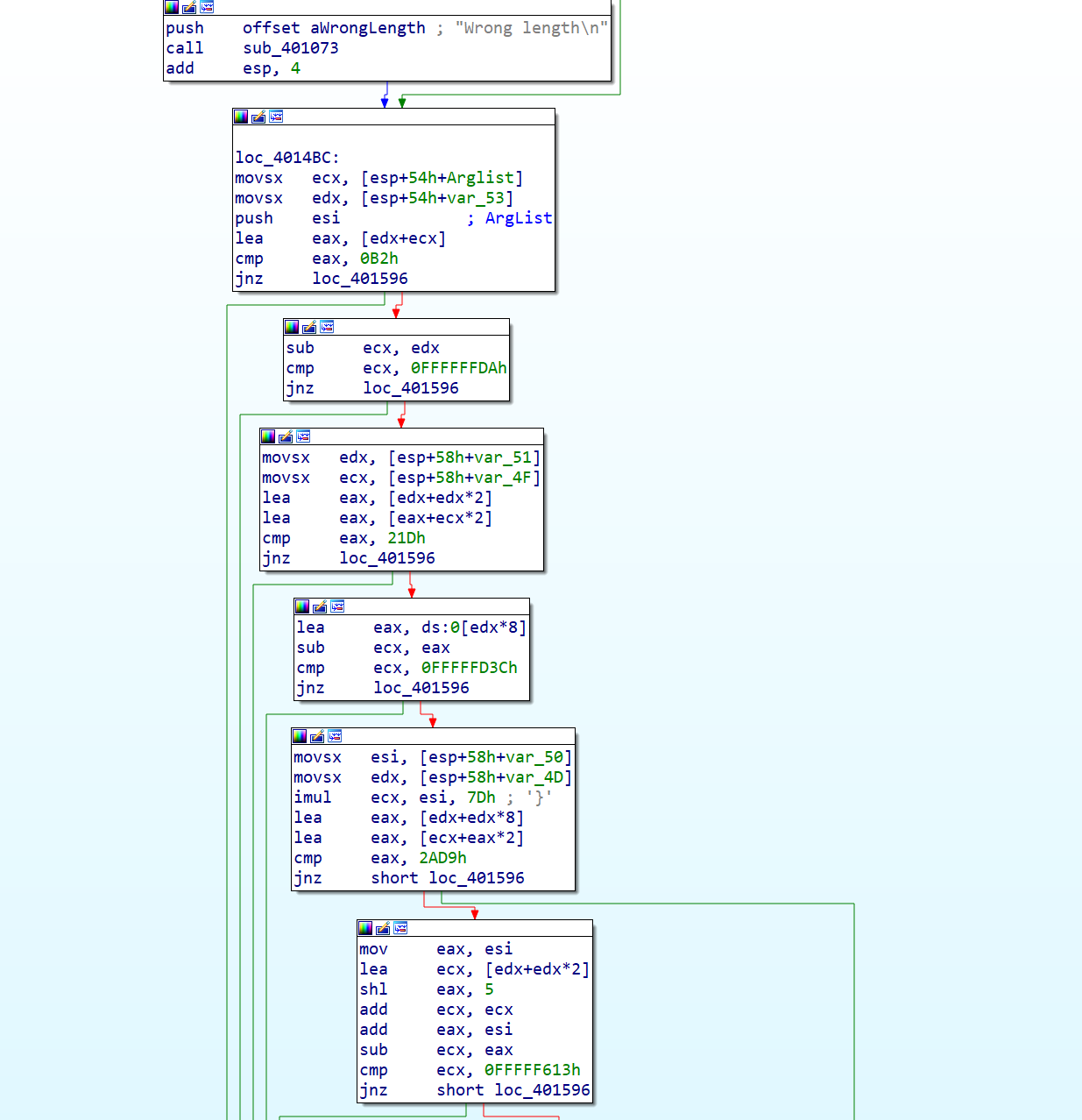
图 6 逆向分析，完成task2练习

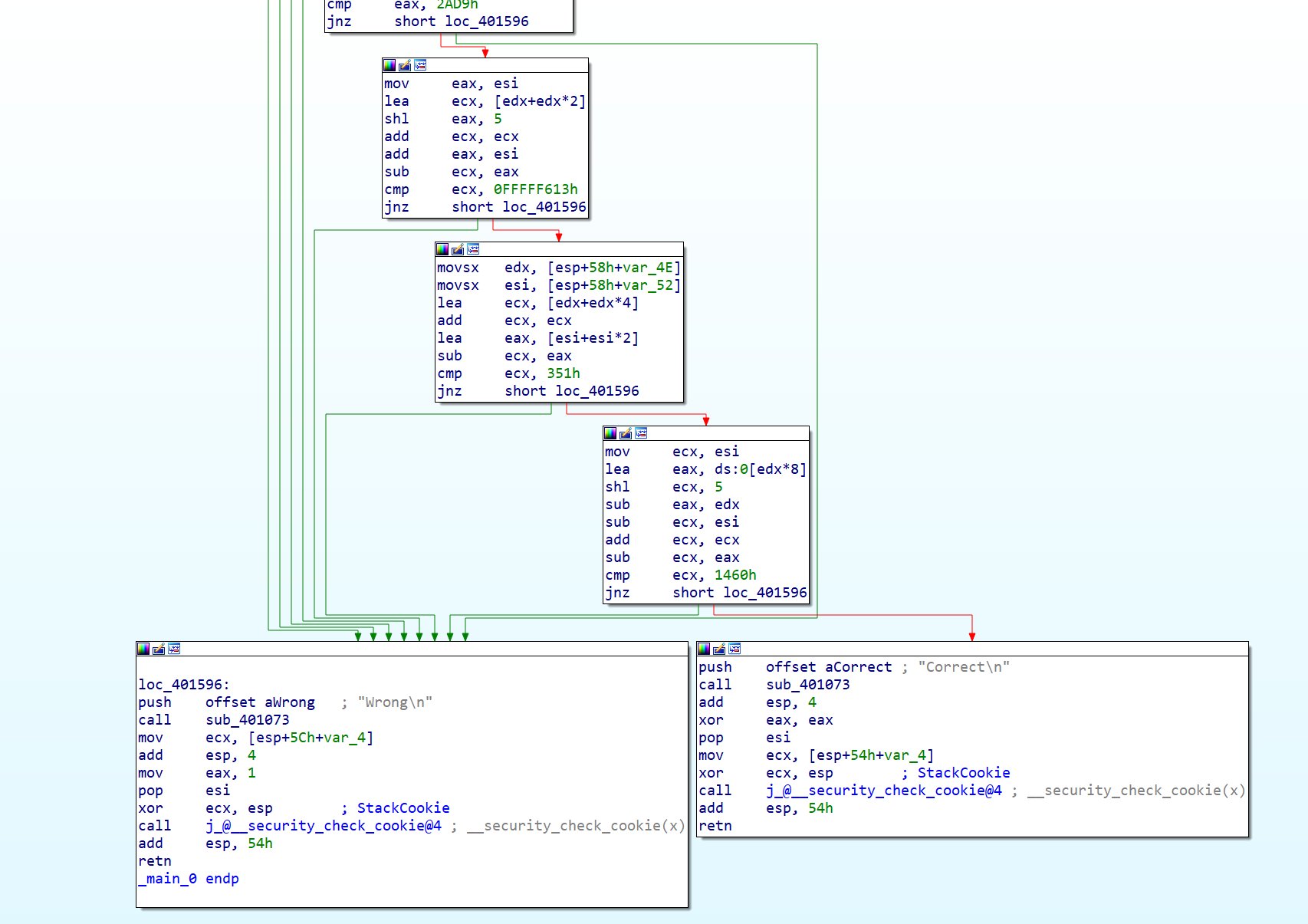
1. **实验步骤**

**（一）task1**

1. 使用IDA Freeware，获得二进制代码的反汇编代码，提供截图。

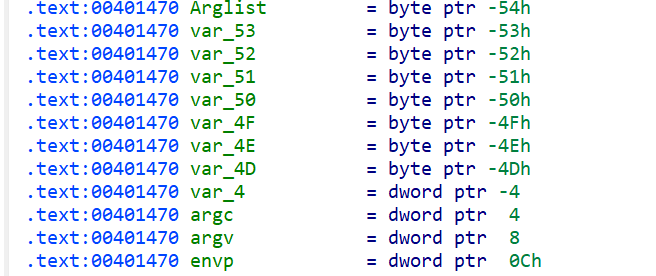






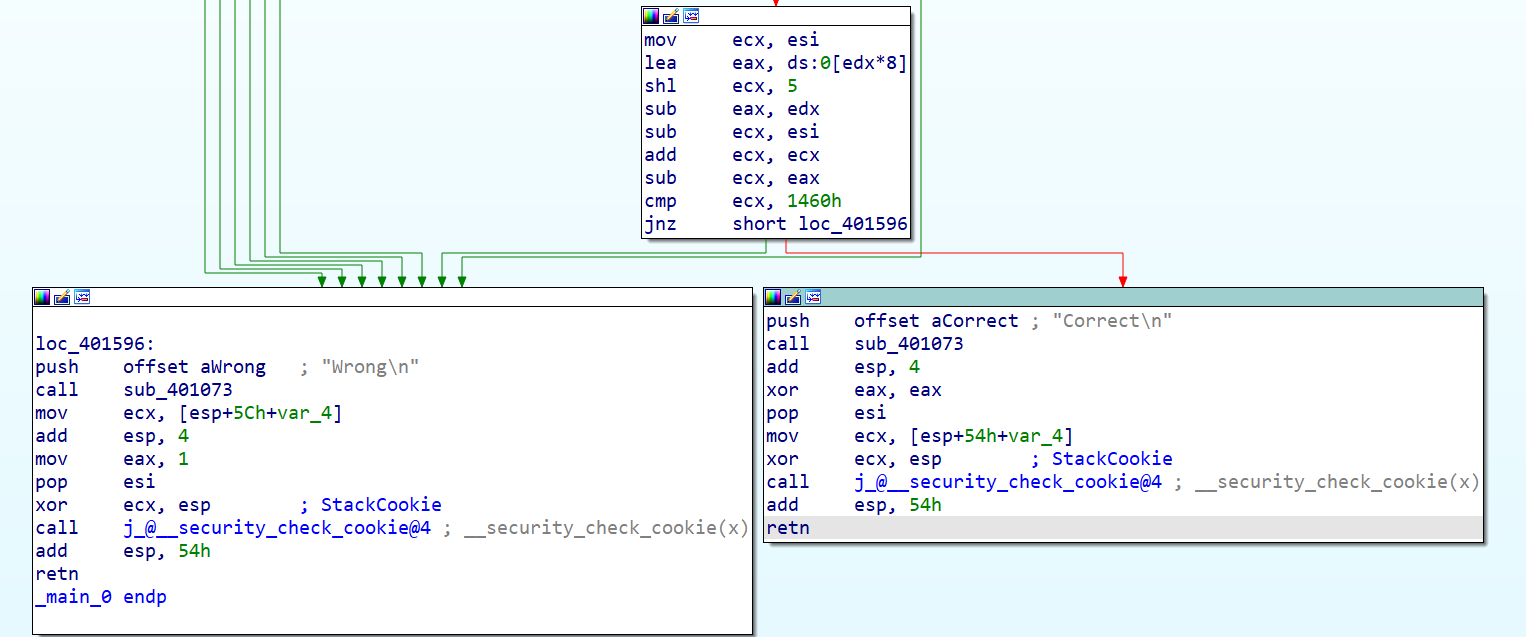
1. 逆向分析二进制代码的计算过程、数据结构、条件判断、分支结构等信息，在实验报告中记录逆向分析的详细过程。

①查看数据段，发现有8个字符型数据，4个双字节型数据。



②从后往前，查看怎样才能输出“成功信息”

1）从下图可知，条件为ecx == 1460h



2）以此类推，还需要满足7个条件，分别为：

ecx==351h

ecx==0FFFFF613h

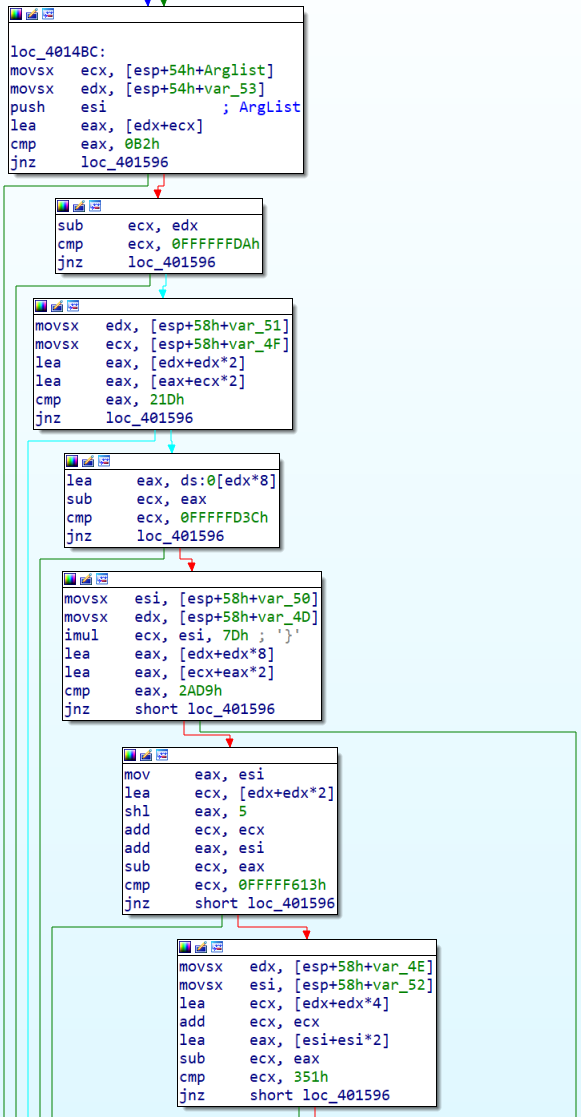
eax==2AD9h

ecx==0FFFFFD3Ch

eax==21Dh

ecx==0FFFFFFDAh

eax==0B2h

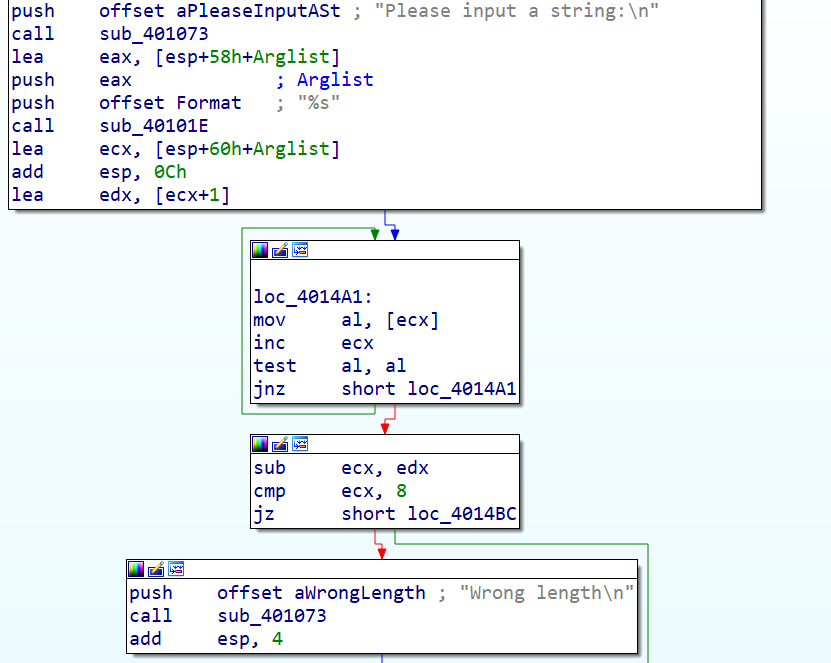


3）到这里发现还有一次错误判断，成功条件为ecx==8，应该是判断输入字符串的字母的个数是否为8。具体如下：

a.让eax=esp+4，并压入栈中；让ecx=esp+6, edx=esp+7

b. test指令检测al是否为空，循环后ecx记录最后一个字符后两位的地址

c.检查ecx-edx是否为8，若成功，ecx应该为eax+1+8+2（eax压入了栈中，应该加上）即esp+15



4）到了最前面，发现这个程序先是定义了几个数据，然后提示我们输入一个字符串，并保存在栈中，之后根据偏移量来获取每一个字符。



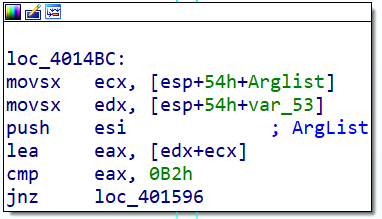
③到此程序的大概脉络已经清晰

接下来具体查看8个条件的判断，根据前面获得的成功条件进行梳理

1）成功条件：eax = 0B2h

即edx+ecx = 0B2h

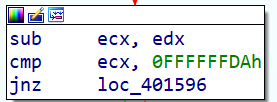
即s1+s2= 0B2h=178



2）成功条件：ecx = 0FFFFFFDAh

即ecx-edx = 0FFFFFFDAh

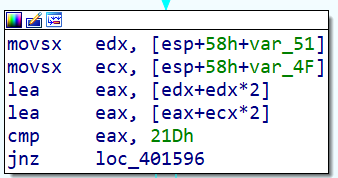
即s1-s2 = 0FFFFFFDAh=-38



3）成功条件：eax = 21Dh

即3\*edx+2\*ecx = 21Dh

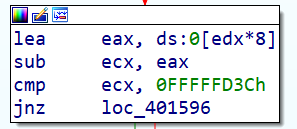
即3\*s4+2\*s6 = 21Dh=541



4）成功条件：ecx = 0FFFFFD3Ch

即ecx-8\*edx = 0FFFFFD3Ch

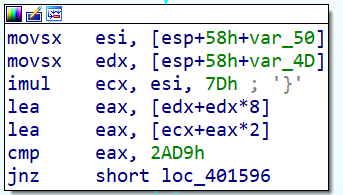
即s6-8\*s4=0FFFFFD3Ch=-708



5）成功条件：eax = 2AD9h

即esi\*7Dh+edx\*9\*2 = 2AD9h

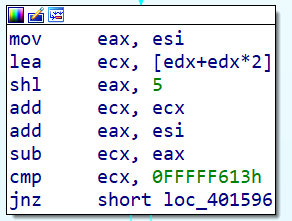
即125\*s5+18\*s8=2AD9h =10969



6）成功条件：ecx = 0FFFFF613h

即edx\*3\*2-32\*esi-esi = 0FFFFF613h

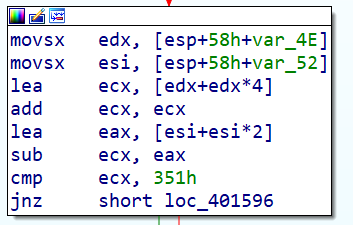
即6\*s8-33\*s5=0FFFFF613h =-2541



7）成功条件：ecx = 351h

即edx\*5\*2-esi\*3 = 351h

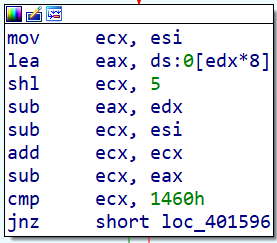
即10\*s7-3\*s3=351h=849



8）成功条件：ecx = 1460h

即(esi\*32-esi)\*2-(edx\*8-edx) = 1460h

即62\*s3-7\*s7=1460h=5216



④总结整个程序框架

1）输入挑战字符串

条件判断： cmp ecx, 8 比较字符串长度和8。

如果字符串长度不等于8，输出错误信息，否则继续执行。

2）字符串的各个部分验证

s1+s2==178

s1-s2==-38

3\*s4+2\*s6==541

s6-8\*s4==-708

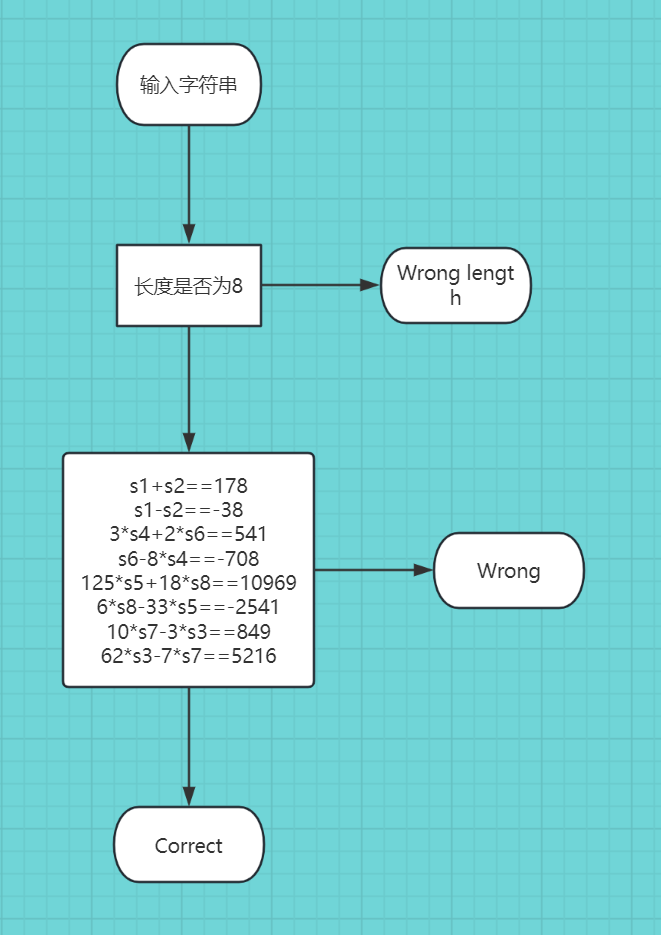
125\*s5+18\*s8==10969

6\*s8-33\*s5==-2541

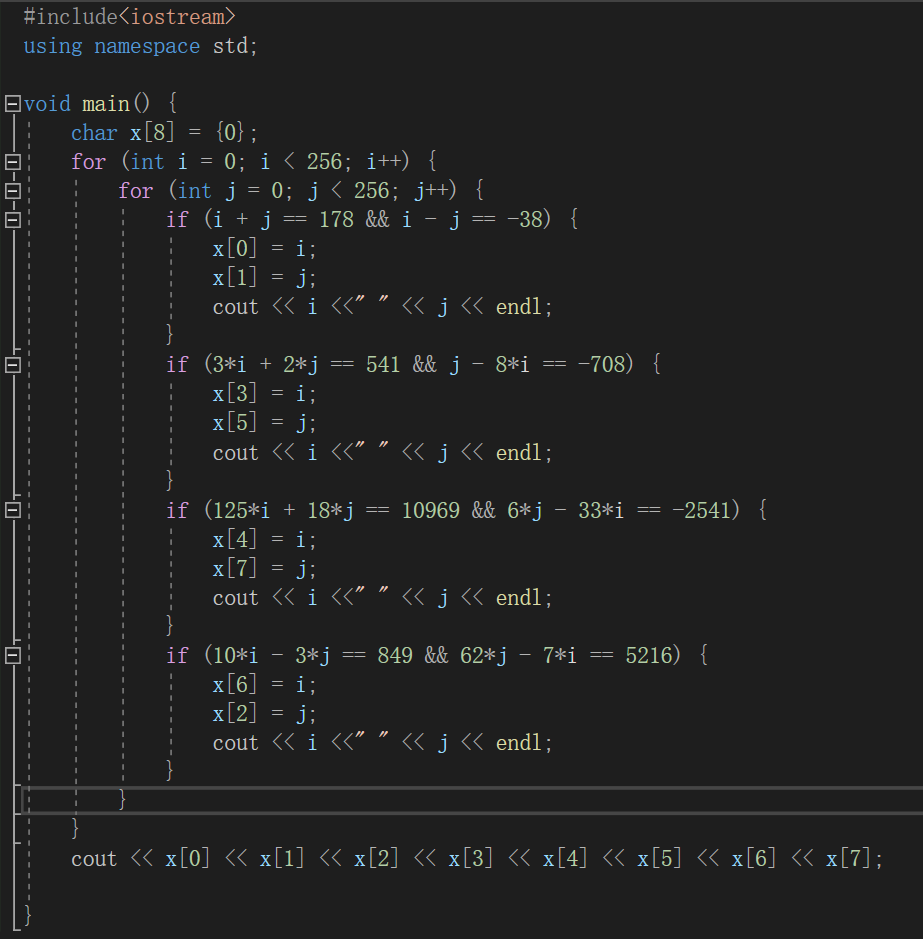
10\*s7-3\*s3==849

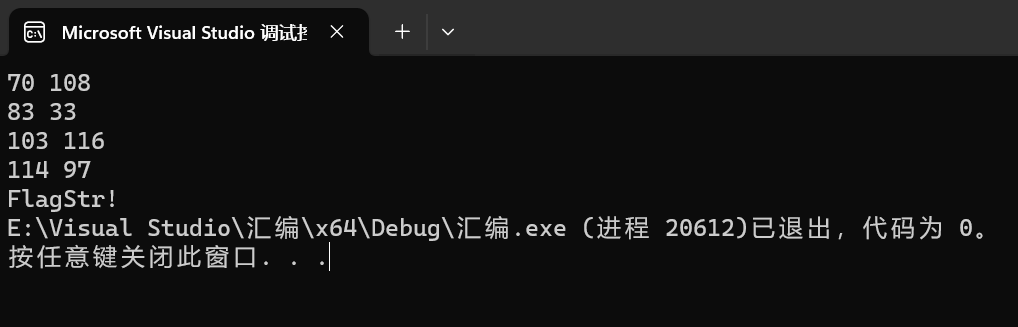
62\*s3-7\*s7==5216

⑤流程图



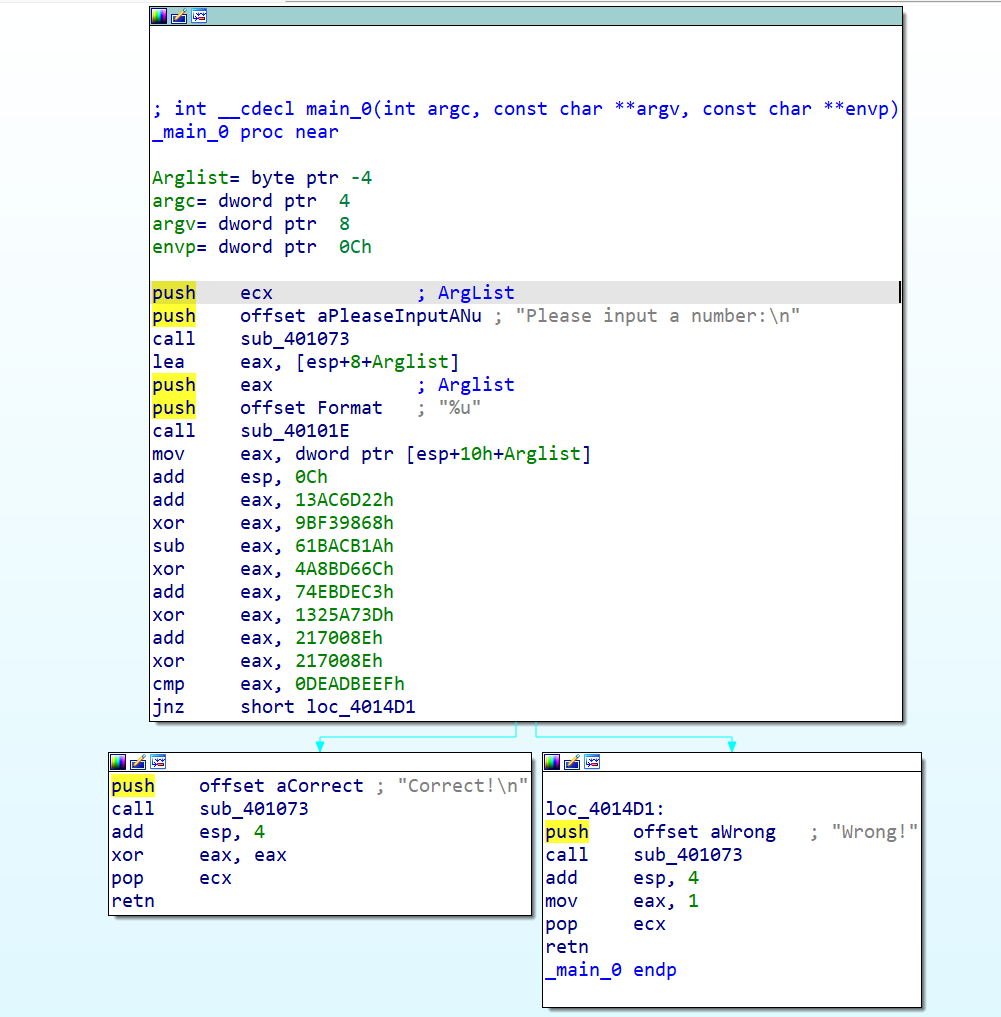
3.解方程组，得到字符串





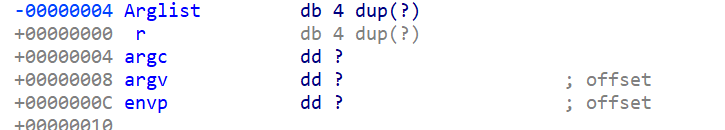
**（二）task2**

1. 使用IDA Freeware，获得二进制代码的反汇编代码，提供截图。



1. 逆向分析二进制代码的计算过程、数据结构、条件判断、分支结构等信息，在实验报告中记录逆向分析的详细过程。

①查看数据段，发现有2个字符型数据，3个双字节型数据。



②分析怎样才能输出成功信息

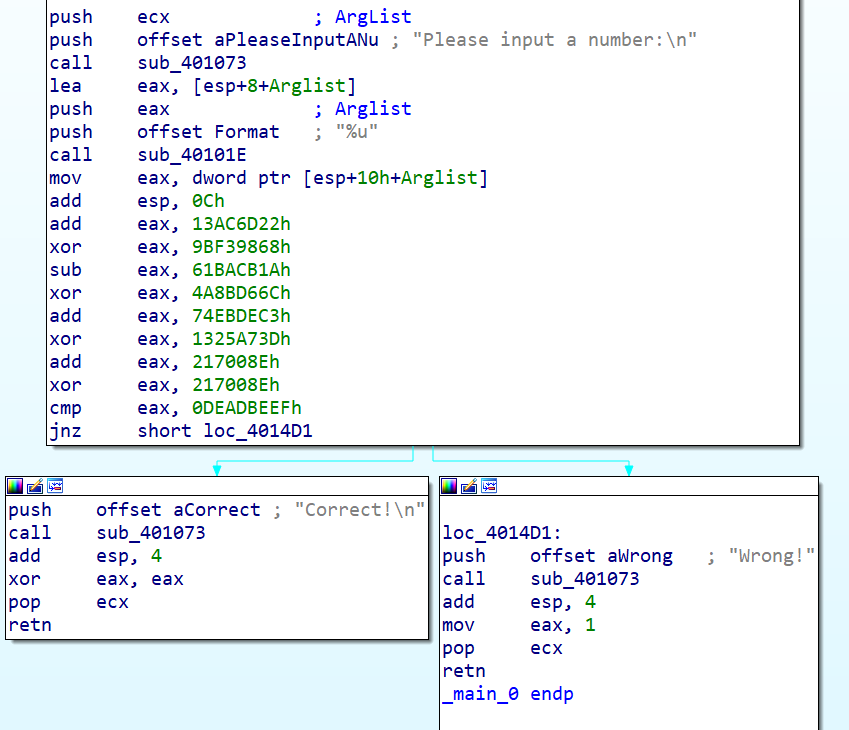
1）程序先是定义了几个数据，然后提示我们输入一个数字，并保存在栈中，之后根据偏移量将其保存在eax中。

2）接下来将eax中的数据进行一系列的加法，减法和异或运算。

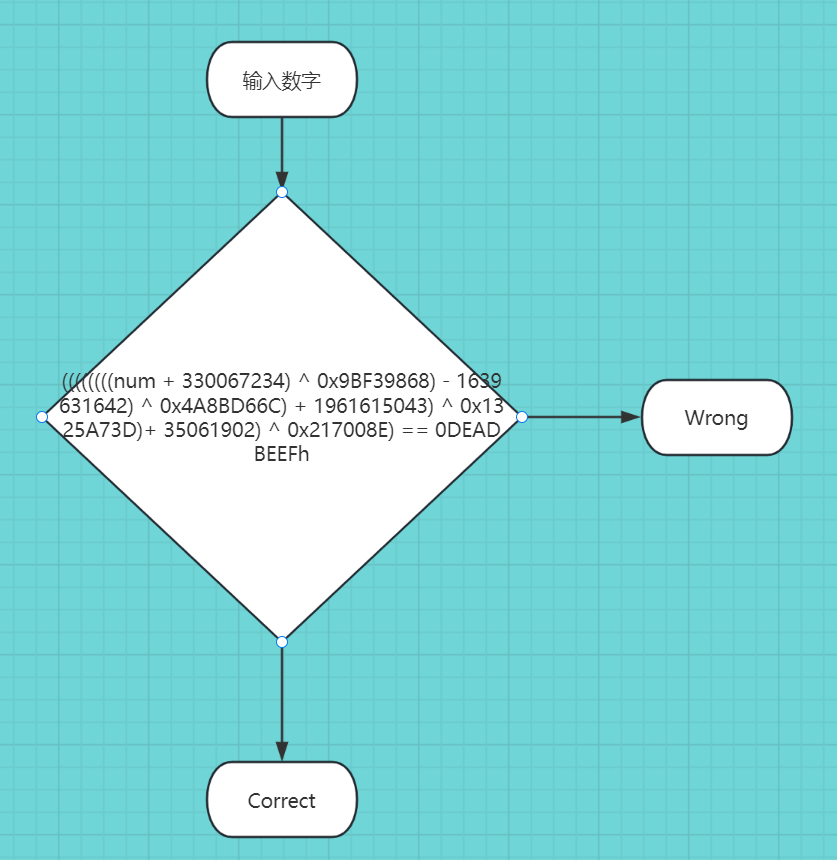
3）判断eax是否等于0DEADBEEFh，等于则成功。

表达式具体如下：

((((((((num + 330067234) ^ 0x9BF39868) - 1639631642) ^ 0x4A8BD66C) + 1961615043) ^ 0x1325A73D)+ 35061902) ^ 0x217008E) == 0DEADBEEFh



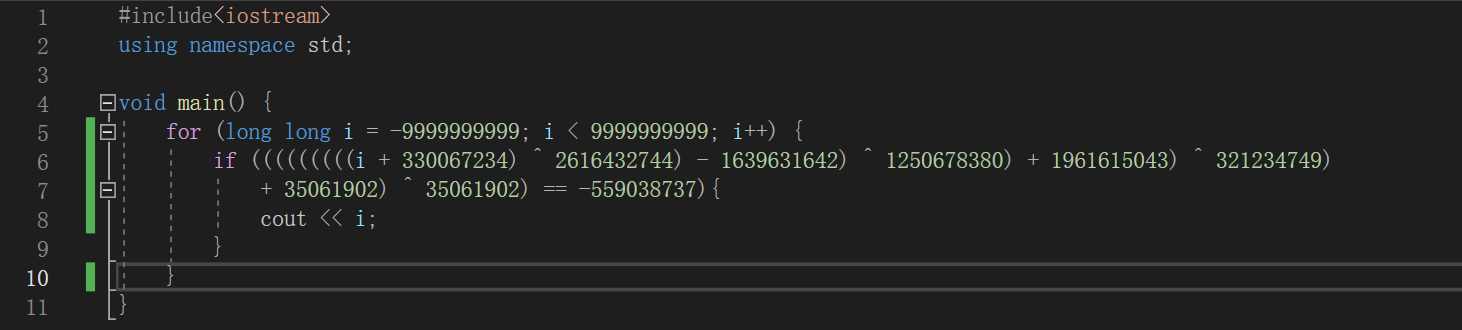
③流程图

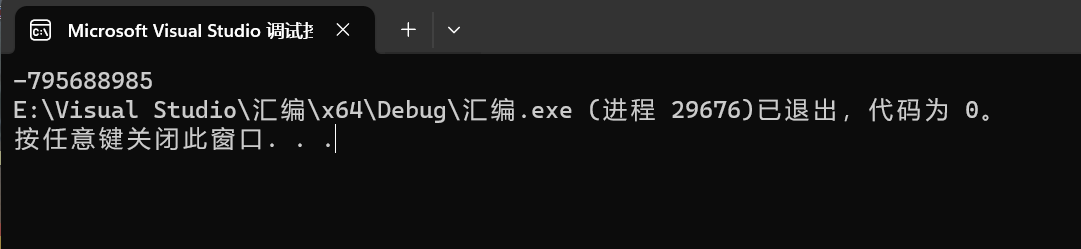


3.解方程，得到数字

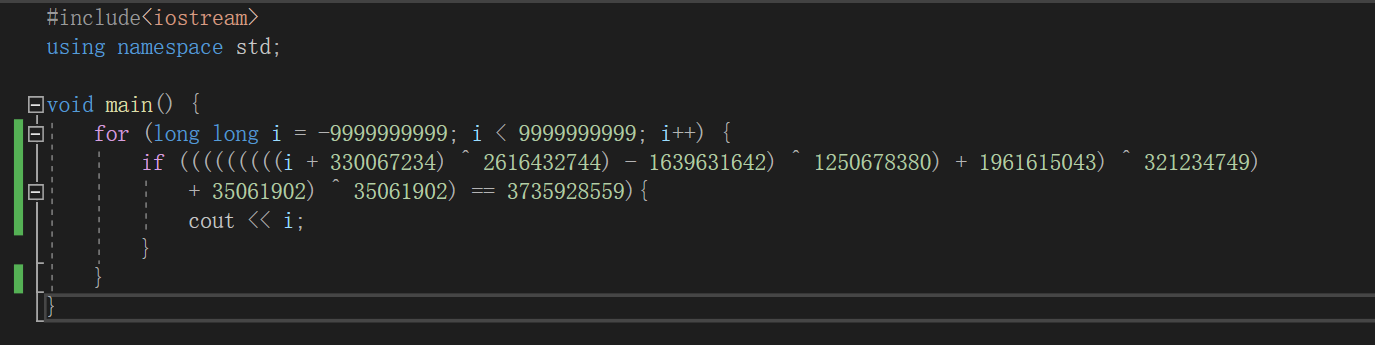
根据0DEADBEEFh编码的正负数分别计算

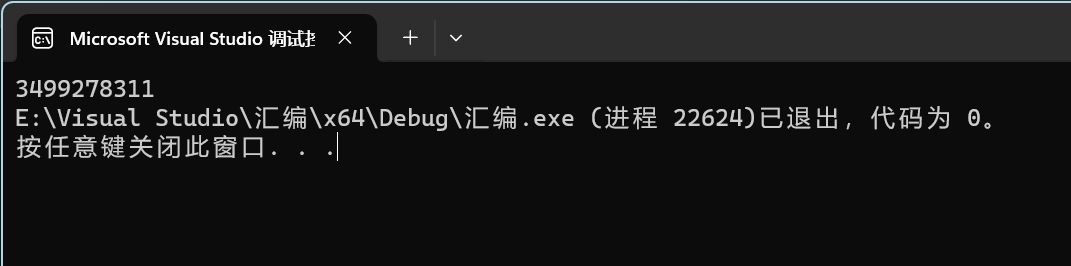
1）0DEADBEEFh为负数





2）0DEADBEEFh为正数





1. **实验结果**

运行程序，根据提示输入字符串和逆向挑战的结果，获得“Correct！”输出，将成功的截图复制到实验报告中。

Task1: FlagStr!

Task2: 3499278311或-795688985

运行程序，输出成功信息，表明实验成功

