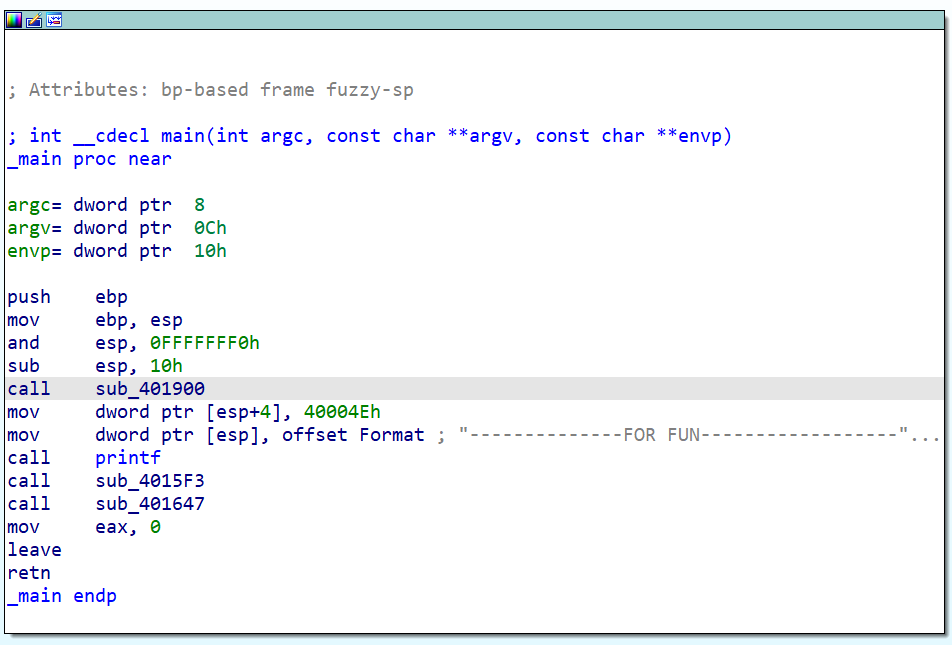
Homework-2-1 Report

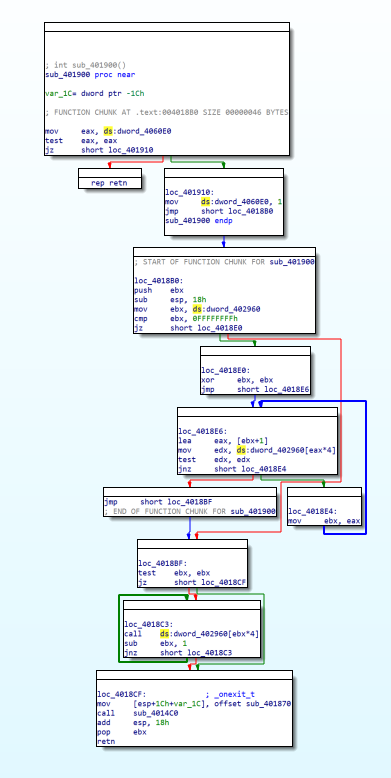
姓名：项 枫 学号：2022211570

**一、程序的伪代码流程图**

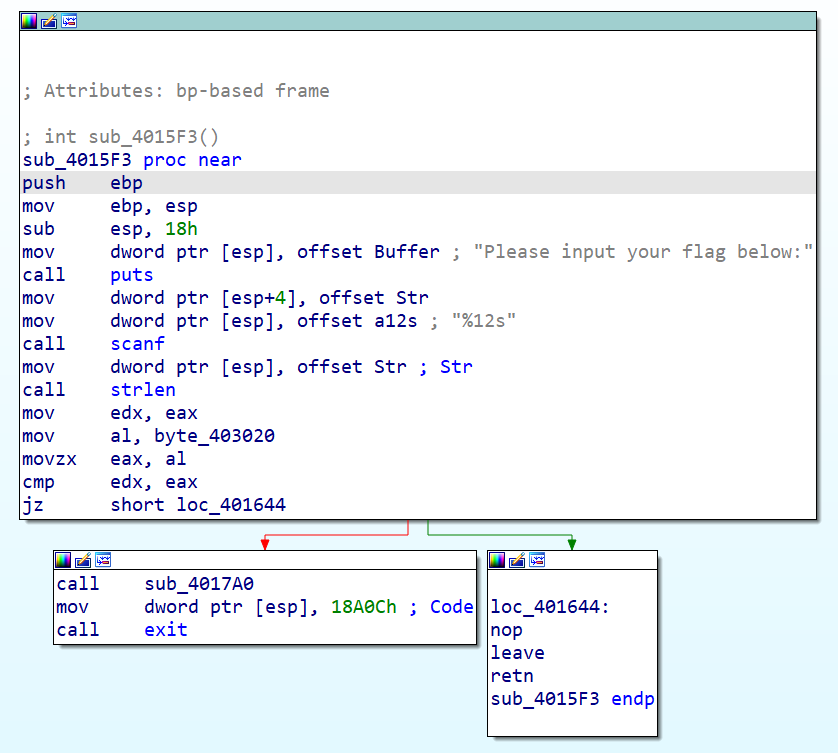
1、main( ):



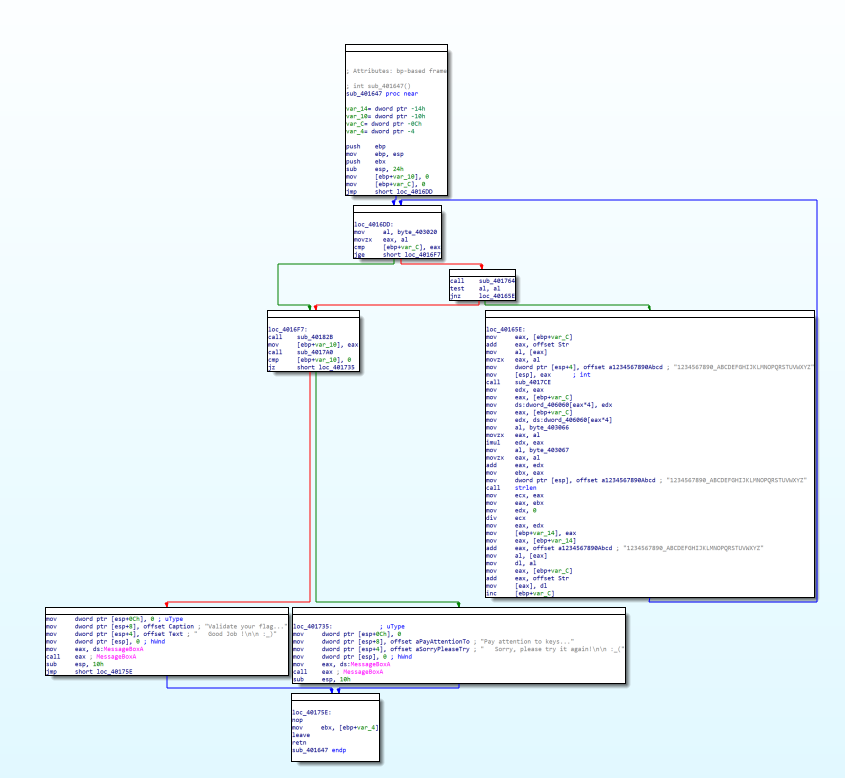
2、sub\_401900( ):



3、sub\_4015F3( ):



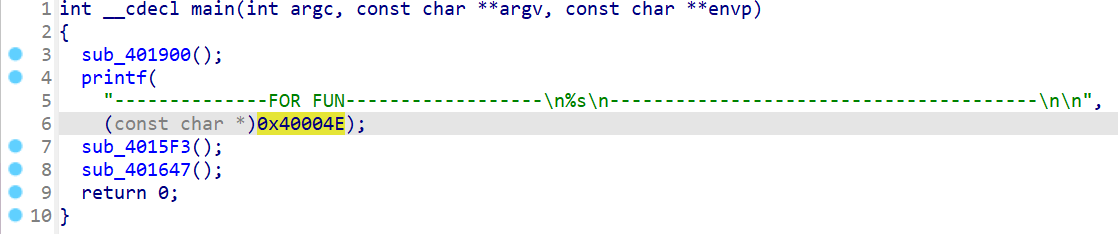
4、sub\_401647( ):



**二、分析字符串“This program cannot be ...”**

1、内存地址

由下图可知，该字符串内存地址为0x40004E。

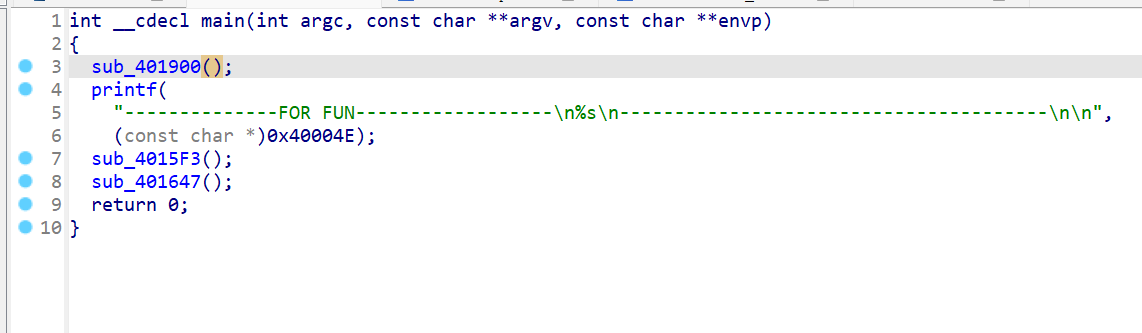


2、字符串长度为39。

**三、逆向分析与思考的逻辑过程**

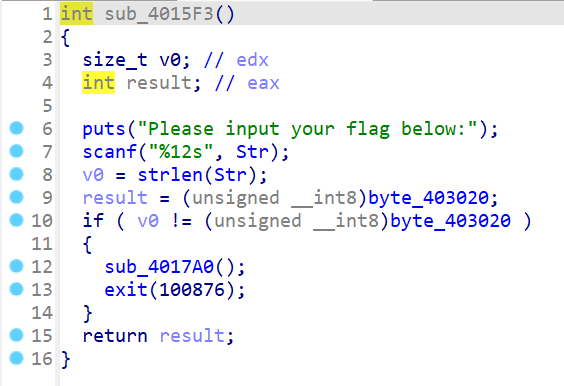
1、程序函数分析

（1）main函数



调用了sub\_4015F3( )，sub\_401647( )函数，将分别在（2）（3）进行分析。

（2）sub\_4015F3( )函数

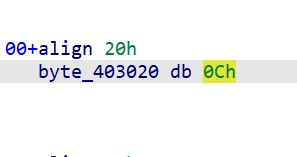


第6行：输出字符串“Please input your flag below:”

第7行：读入字符串Str，此处“%12”可知字符串长度为12。

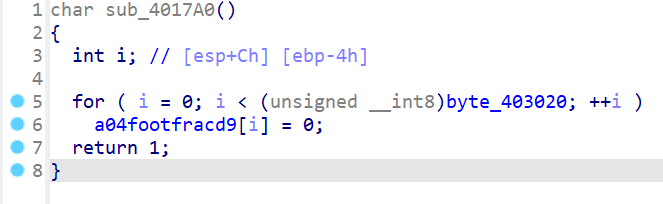
第8行：计算Str长度v0。

第9行：result值为12。如下图，byte\_403020的值为0Ch，十进制值即为12。

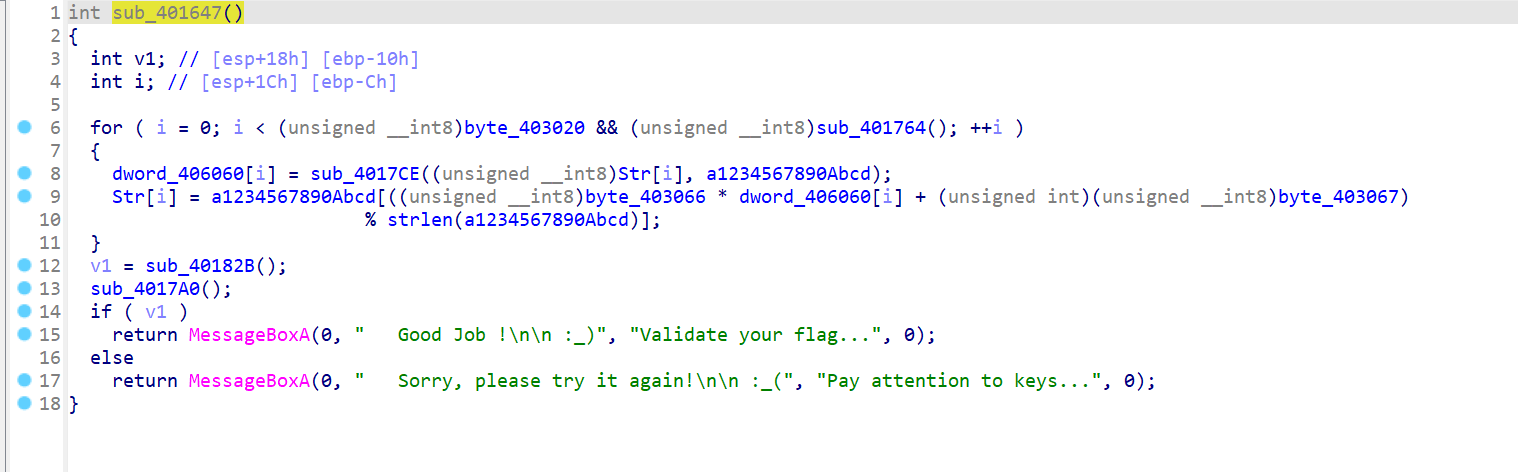


第10行：判断字符串Str长度是否为12。

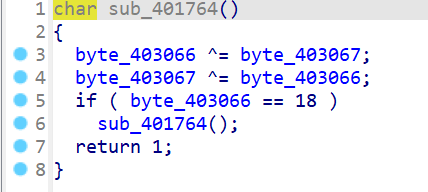
第12、13行：字符串Str长度不为12，跳转sub\_4017A0( )函数。如下图，该函数功能为将a04footfracd9数组初始化为全0，并且exit。

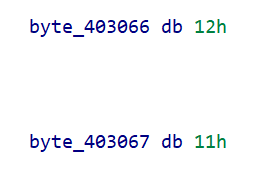


（3）sub\_401647( )函数

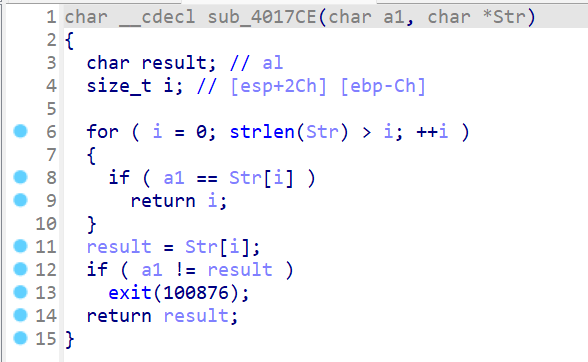


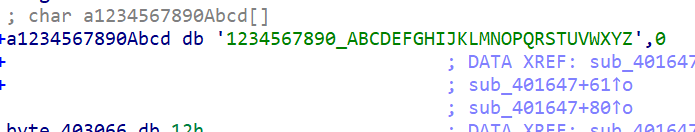
第6行：for循环，循环12次，将输入字符串进行加密（第8-10行）；此处需要注意sub\_401764( )函数，如下图，每次循环都更新byte\_403066 和byte\_403067的值，初值为12h（对应十进制18）和11h（对应十进制17），执行该函数可知，i为偶数时，byte\_403066 =3，byte\_403067=18；i为偶数时，byte\_403066 =17，byte\_403067=3。





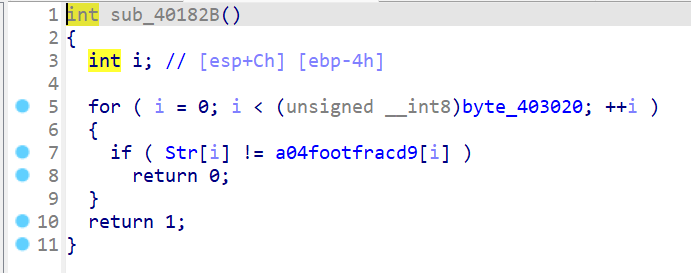
第8行：执行sub\_4017CE( )函数，将函数值赋给dword\_406060[i]，如下图，该函数功能为：查找字符Str[i]在字符集a1234567890Abcd中的位置，由下图可知该字符集为1234567890\_ABCDEFGHIJKLMNOPQRSTUVWXYZ，并返回该位置，为第9行进行加密过程做准备。

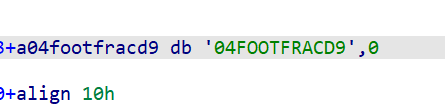




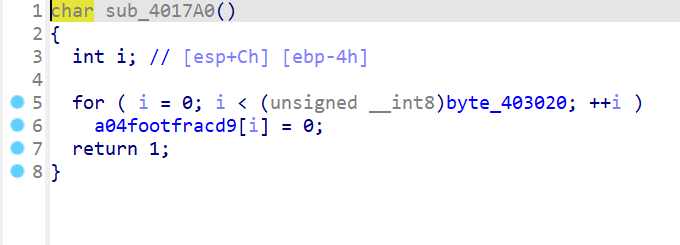
第9、10行：仿射加密过程。Str[i] =byte\_403066 \* dword\_406060[i] +byte\_403067)% strlen(a1234567890Abcd)]。

第11行：执行sub\_40182B( )函数，将函数值赋给v1，如下图，该函数功能为：判断加密后的字符串Str与a04footfracd9的字符串（由下图可知该字符串为04FOOTFRACD9）是否相等，相等返回1，否则返回0。





第13行：执行sub\_4017A0( )函数，如下图，该函数功能为：将字符串a04footfracd9初始为000000000000。



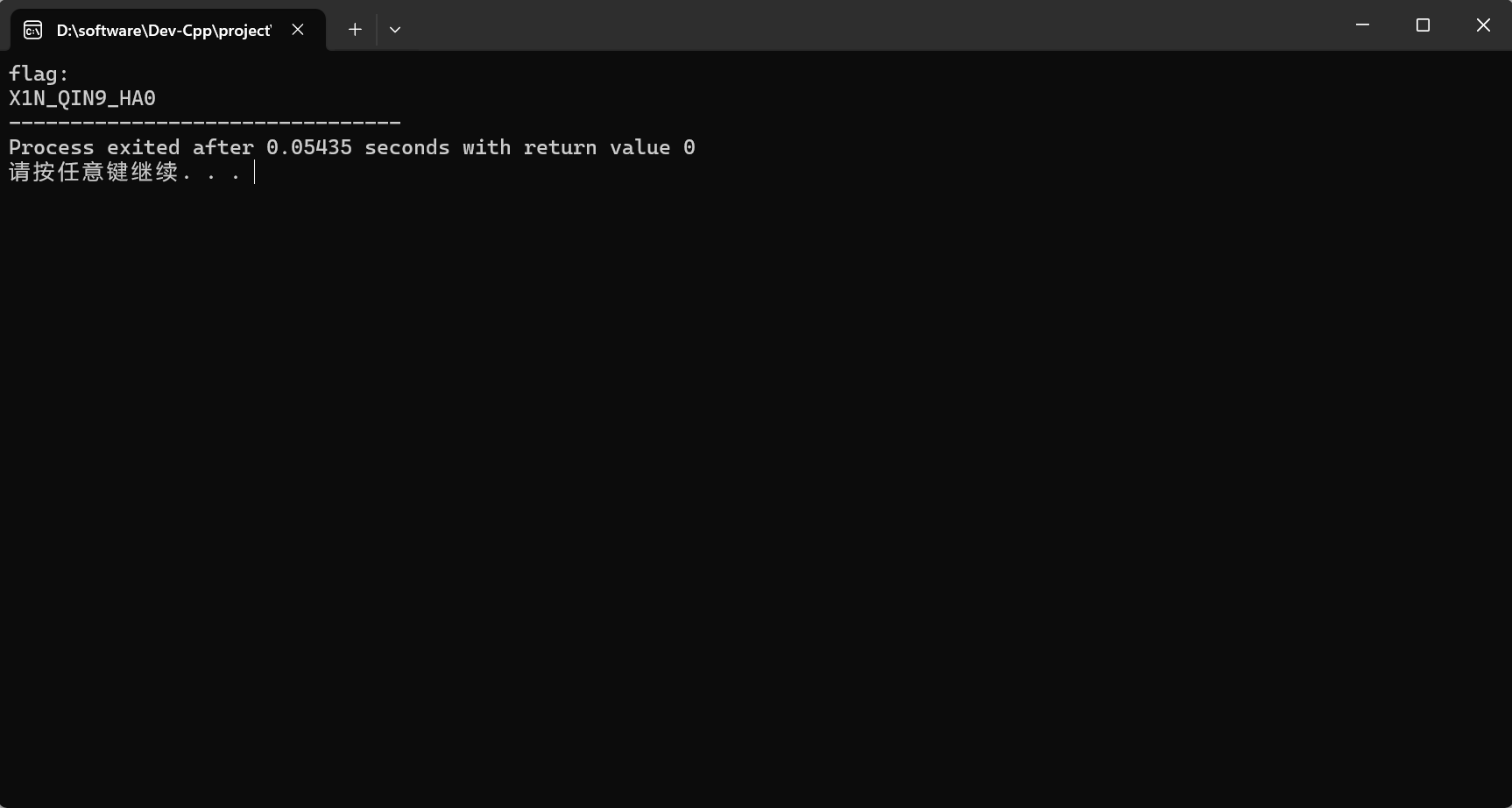
第14-17行：如果v1为1，则输出“ Good Job !\n\n :\_)”，证明flag通过；如果v1为0，则输出“ Sorry, please try it again!\n\n :\_(”，证明flag未通过。由此便可知，第11行a04footfracd9处的字符串（04FOOTFRACD9）即为密文。

2、解flag程序

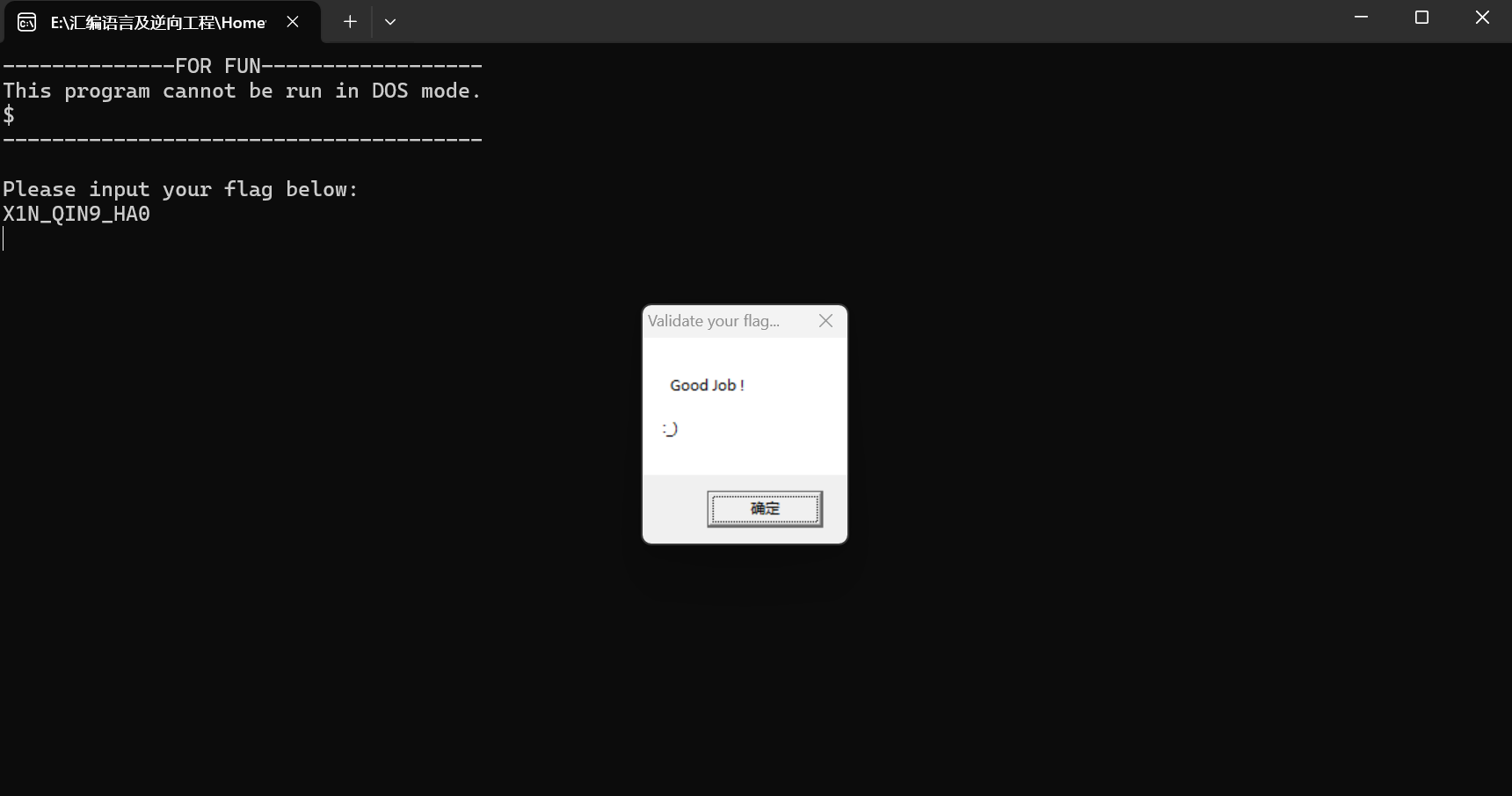
（1）程序代码（C++）如下：

1. #include<bits/stdc++.h>
2. **using** **namespace** std;
3. **int** main(){
4. **int** key\_1\_1=3;//i为偶数时的key\_1
5. **int** key\_2\_1=18;//i为偶数时的key\_2
6. **int** re\_key\_1\_1=25;//key\_1\_1的逆
7. **int** key\_1\_2=17;//i为奇数时的key\_1
8. **int** key\_2\_2=3;//i为奇数时的key\_2
9. **int** re\_key\_1\_2=24;//key\_1\_2的逆
10. **char** c[]="04FOOTFRACD9";//密文
11. **char** dic[]="1234567890\_ABCDEFGHIJKLMNOPQRSTUVWXYZ";//字符集
12. **char** result[20]={0};//结果flag
13. **int** i,j,c\_len,dic\_len,temp;
15. c\_len=strlen(c);//密文长度
16. dic\_len=strlen(dic);//字符集长度
17. **for**(i=0;i<c\_len;i++){
18. **for**(j=0;j<dic\_len;j++){
19. **if**(dic[j]==c[i]) temp=j;
20. }
21. **if**(i%2==0) temp=((temp-key\_2\_1+dic\_len)\*re\_key\_1\_1)%dic\_len;//i为偶数时解密
22. **else** temp=((temp-key\_2\_2+dic\_len)\*re\_key\_1\_2)%dic\_len;//i为奇数时解密
23. result[i]=dic[temp];
24. }
26. cout<<"flag:"<<endl;
27. cout<<result;
28. **return** 0;
29. }

（2）程序运行结果



3、flag成功通过



**四、收获和感受**

最开始忽略了sub\_401764( )函数，误认为密钥对为（18，17），导致flag一直不通过，最后在冷静的观察与分析下，理解了该函数的功能，密钥对是（3，18）和（17，3）的反复循环，最终flag成功通过！

通过本次作业，对IDApro软件的使用有了更深的理解，也初步学会了逆向分析。

最后，感谢潘老师课上辛勤的教学。