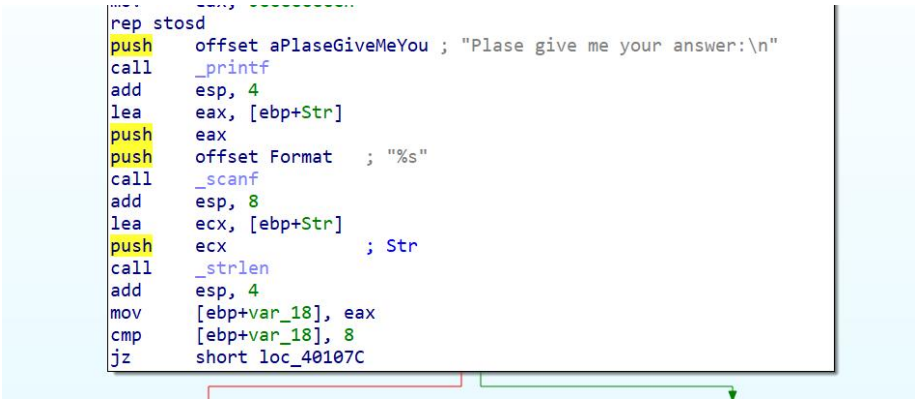
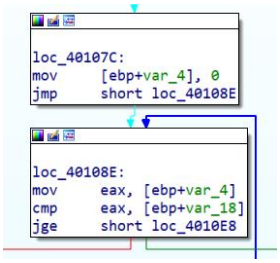
汇编语言与逆向工程第一次作业：对 Cpp1.exe 逆向分析

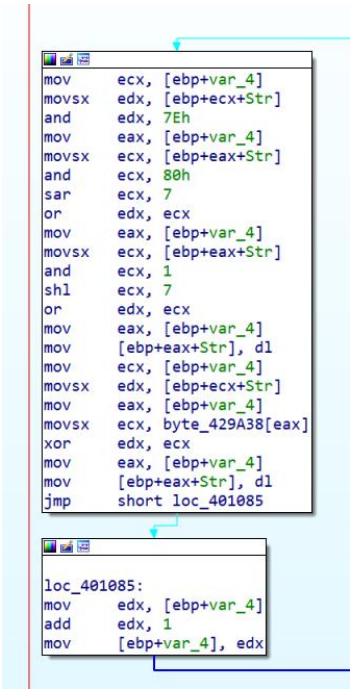


在找到“Plase give me your answer:”后定位到函数，
函数_scanf 有两个参数，将输入的字符串存入 Str；
然后取 Str 的地址到 ecx
函数_strlen 得到 Str 的长度在 eax
比较，若长度是 8 位，跳绿色的线

然后看跳到右面后：

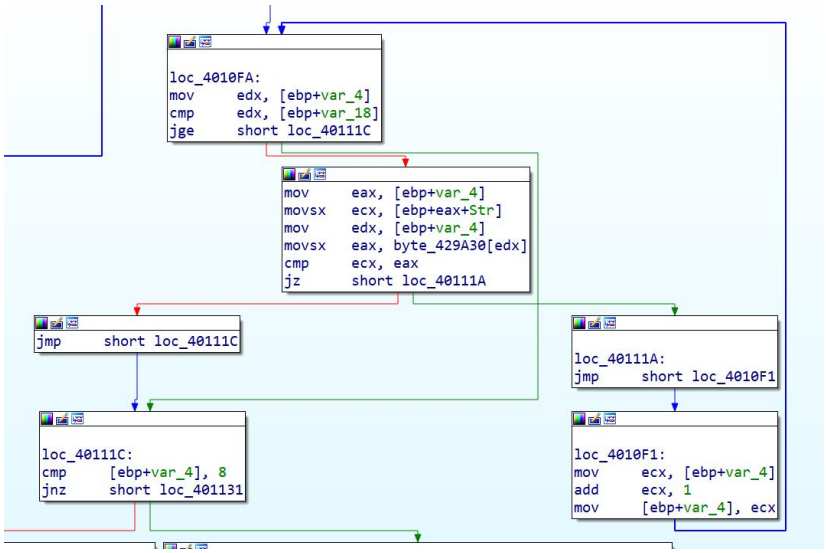


var_4 是初始化的 0；下面用这个 var_4 和 8 比较，大于等于 8 跳到右面，
所以要先向左跳



左边：ecx 存那个 var_4
[ebp+ecx+Str]所以就是 Str[var_4]的值
and，和 01111110 与运算，取八位二进制数的中间 6 位到 edx
and，和 10000000 与运算，取八位二进制数的最高位到 ecx
然后 ecx 右移 7 位
再把 ecx 和 edx 组合到一块到 edx
再把最低位移到最高位
然后取 byte_429A38[eax]到 ecx
用 ecx 对 edx 异或加密
再循环

总的来说，就是每一位密码要把 8 位 2 进制的最高位移到最低位，最低位移到最高位；然后和 byte_429A38[eax]异或加密，byte_429A38[eax]中为{6，7，8，9，10，11，12，13}



这里又是一个 for 循环，循环数是 var_4,从 0 到 8，和 byte_429A30[var_4]里面的数比较
而 byte_429A30[var_4]分别是{52h,C7h,C2h,CDh,EEh,EBh,FEh,F5h}

知道了 52h,C7h,C2h,CDh,EEh,EBh,FEh,F5h 密文再反推明文。
把这 8 个数分别和 6，7，8，9，10，11，12，13 异或，再把最低位和最高位调换就能得到明文

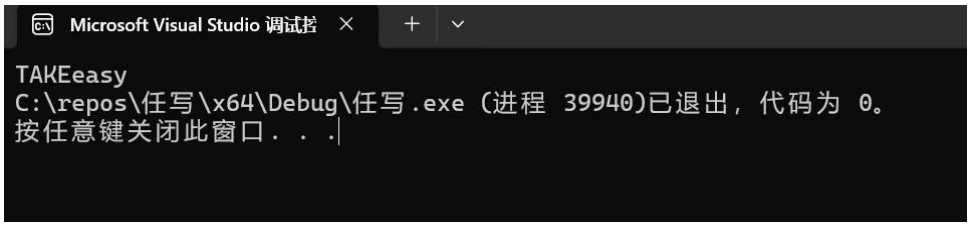
C++代码：

Main.cpp

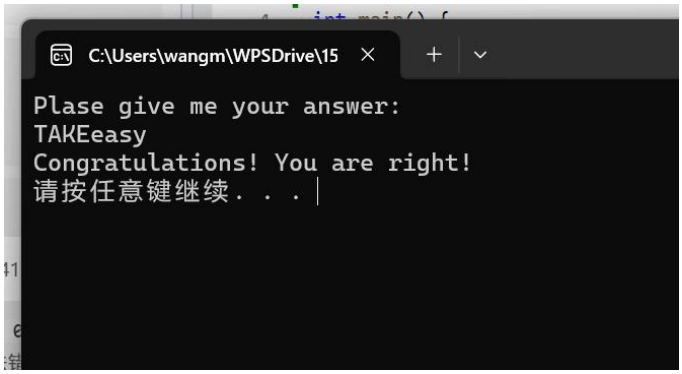
任写 (全局范围)

```cpp
#include<iostream>
using namespace std;

int main() {

    int h = 0;
    int b = 0;
    int l = 0;

    int array[8] = {0X52,0X0C7,0X0C2,0X0CD,0X0EE,0X0EB,0X0FE,0X0F5};
    for (int i = 0; i < 8; i++)
    {
        array[i] = array[i] ^ (i + 6);
        h = array[i] & 0X80;
        b = array[i] & 0X7E;
        l = array[i] & 0X01;
        l=l << 7;
        h = h >> 7;
        array[i] = l | b | h;
    }

    for (int i = 0; i < 8; i++)
    {
        cout << char(array[i]);
    }

    return 0;
}
```

运行结果：



测试：



结果正确！