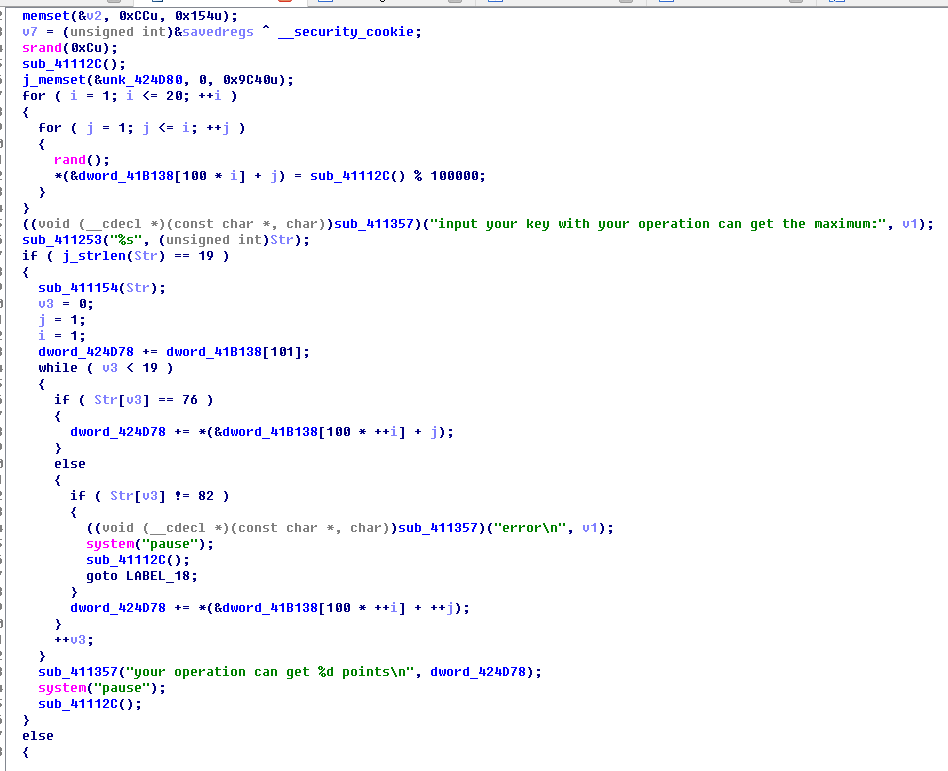
Mountain climbing：

这题加了UPX壳，我还把UPX的UPX!标识符给删了，用网上的一般脱壳机是不能识别的，但可以修复标识符后再用工具脱，或者OD手脱。



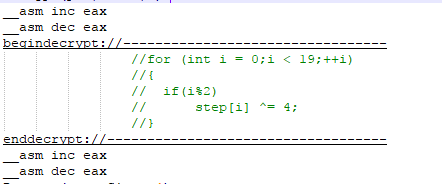
脱掉之后直接IDA。



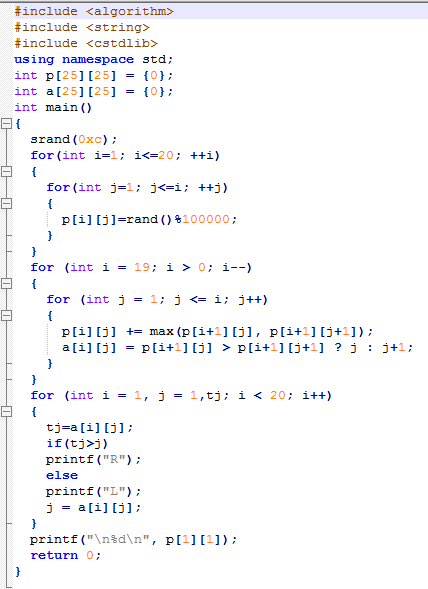
这题需要一定经验，因为分析出来的函数很乱，甚至能看到rand()函数单独在一行，但实际上是在sub\_41112C()这个位置的，还有一些printf和scanf都乱了。但是这个问题在ida7.0里得到了改善

程序的流程是先生成一个地图，是一个数塔，要求用户输入指令，在数塔上模拟移动，每一个目标点都有这对应的积分，要求移动经过的路径的积分最大，需要用DP解决，网上就能找到代码。

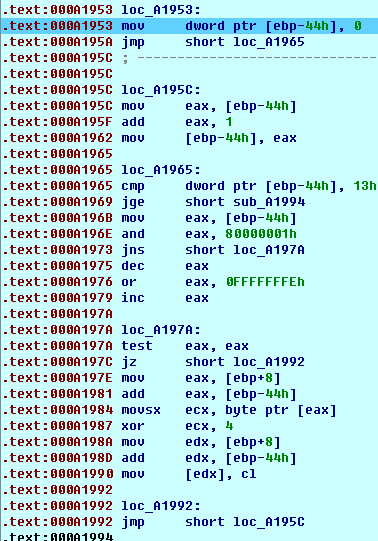
除了杂乱的伪代码外，另一个可能会卡的地方是sub\_411154((int)Str)函数，这个函数的主要功能是动态解码程序部分代码，然后再执行，函数功能如下



其中step是用户输入字串，进行处理之后再传进下面的判断程序，主判断程序的解决脚本如下：

(程序出来的结果需要删除一个字符，因为并没有优化)

出来的字串再按照动态解码的函数内容



对字串进行异或就是答案了

Flag:zsctf{RVRVRHLVRVLVLVRVLVL}

Take the maze:

一开始不打算出压轴。。。但是看到题库的大佬扫题的战绩，想了想还是出道压轴的比较好。

这题的核心是VM+自定义检验，检验是以一个迷宫的形式检验字串能否按对应的操作到达终点。

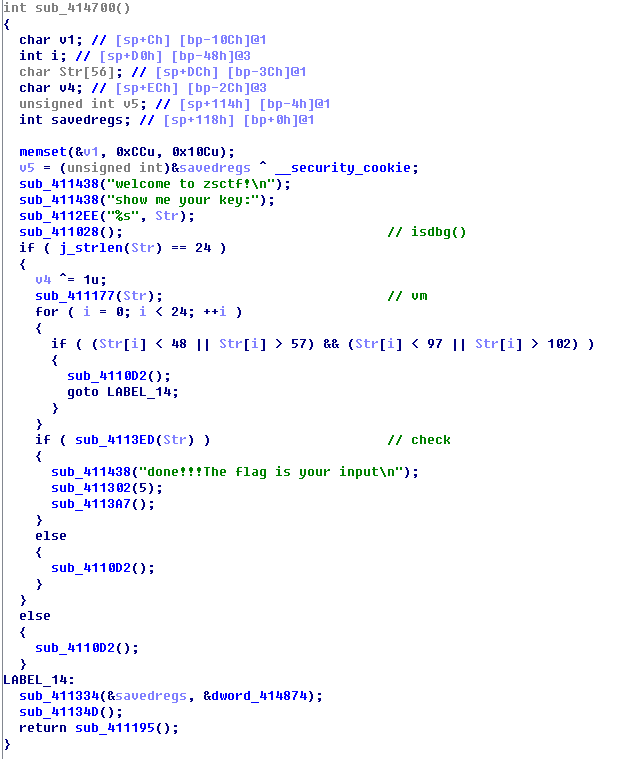
首先在程序某些地方加了”慵懒”的debug检测，防一手猛男们调试，也存在栈不平衡来了一手防F5，需要patch C:\Users\Administrator\AppData\Roaming\Tencent\TIM\Temp\6PZT)JB{JDQ[FE~$M)}6GW0.gif



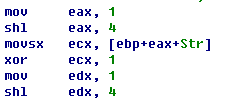
首先要将程序patch后才能F5，搞定之后进入主题。

一开始是一个入门级vm，可以尝试翻译字节码。实在翻译不出来也可以用黑盒的方式测试，也为了方便调试，写了32位的。

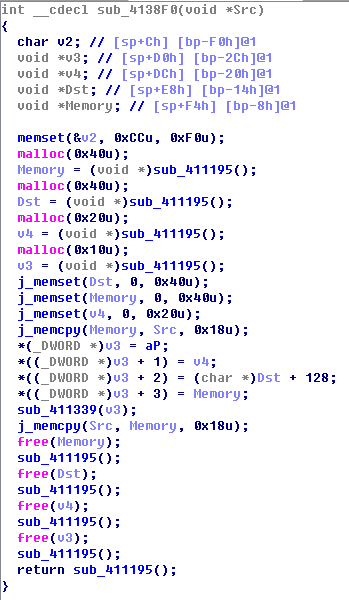
IDA打开，用字串定位主函数。



这里有个问题就是出现了一个v4^1u，在伪代码中并说明v4得指向，查看汇编

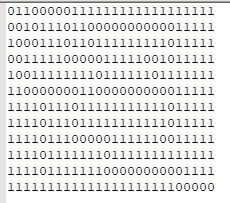
可得是指向字串的第17位(str[16])。

而从程序可以确定输入字串的长度为24，输入之后就在vm中进行处理。

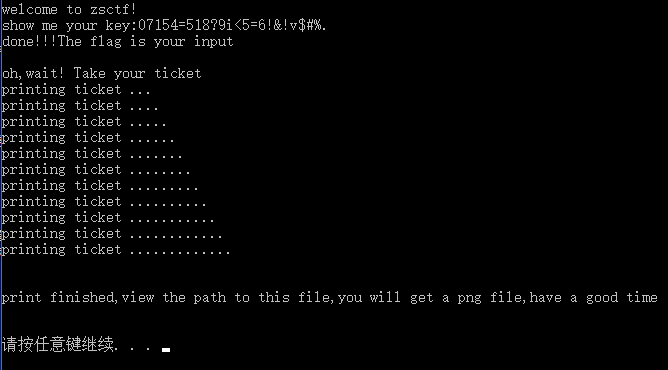


其中aP就是要翻译的字节码，通过翻译或者黑盒就可以得出VM的行为是将字串的每一位异或i++，i从0开始，即可推测出处理前的字串，然后返回主函数，进入判定函数

判定函数有点长，就不截图了，核心是一个迷宫，通过输入字串作为索引，将预设好的被索引字串作为移动标识，分析代码可得到地图为



而且程序中的移动存在循环，而且看到对输入字串的处理后可以知道实际上移动方向后面是跟着次数，即可得出最后正确的字串，然后再逆推过vm部分，就可以得到正确的输入字串，输入后得到



可以看到文件目录下出现了一个png文件，是一个二维码，



扫描可得

Flag:zsctf{07154=518?9i<5=6!&!v$#%.Docupa}