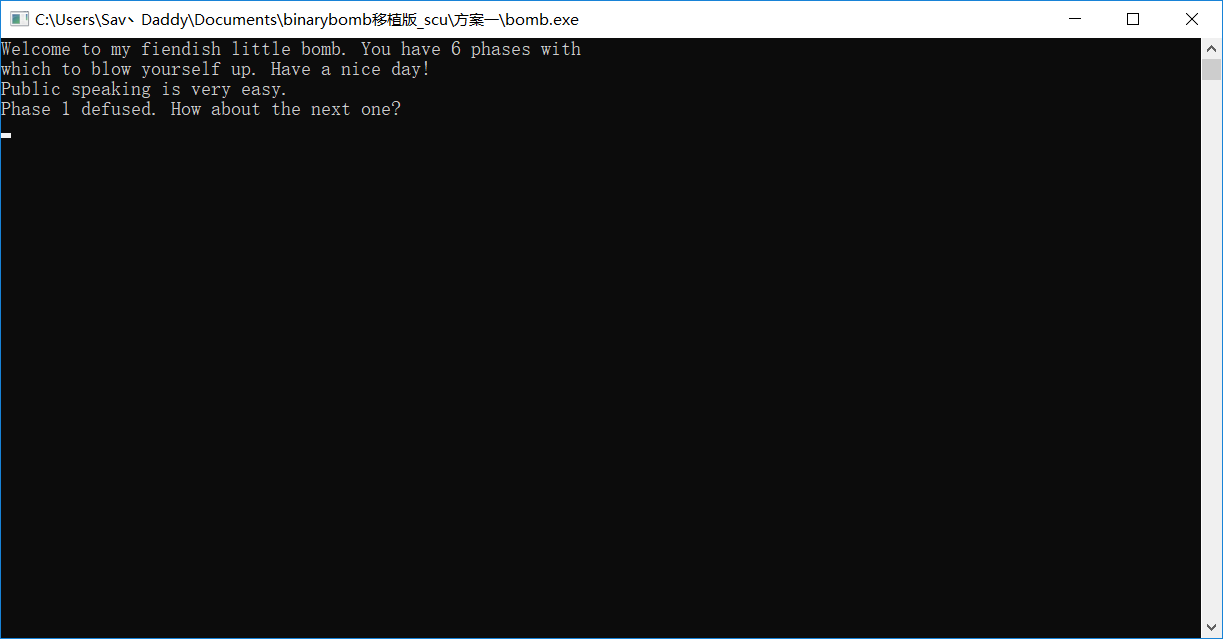
Binary bomb解题过程与结果：由于自己电脑运行windows速度感人，于是借同学的电脑完成的本次实验。

第一个phase，显然是要输入



之后会将输入字符与图中的字符串进行比较然后如果相等则跳转到函数结尾

所以结果为：



第二个phase，首先可以看到他输入了六个数，并且把numbers的第一个元素放进了eax

之后比较eax中的值与1的值

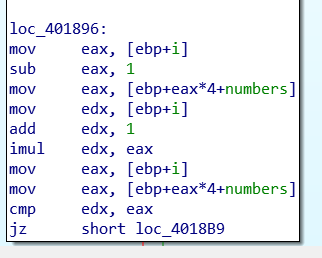


如果不等于则爆炸，于是第一个数是1

接着程序把i赋值为了1



并且进入一个循环



分析这个循环的汇编代码我们可以大致的把他所对应的高级语言代码写出来，首先我们看到程序把i放进了eax寄存器然后令eax减一，然后将numbers数组的指针向后偏移eax倍四个字节，指向numbers的第eax个元素，并将其放入eax，然后将i再放入edx，并且令edx加上1，之后令eax与edx的值相乘，结果放入edx，eax的值我们知道是numbers[I - 1]所以这几句汇编所对应的代码是number[i - 1] \*(i + 1),接着再将i的值放入eax，并且访问numbers的第i个元素，最后判断edx与eax中的值，从以上的汇编我们可以得到的高级语言语句是：

For(int i = 1;i <=5;i ++){

If(numbers[i - 1] \* (i + 1) != numbers[i]) explode();

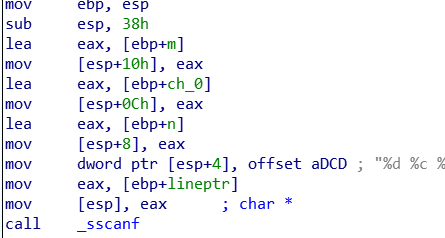
}

所以根据上述代码可以得到剩下的5个数是

1 2 6 24 120 720

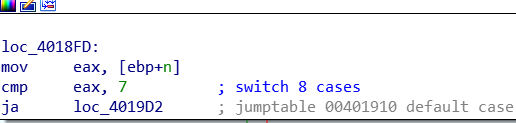
第三阶段

首先观察第一个块的汇编



很显然这是给sscanf传入三个变量的地址然后输入三个参数数n ch\_0 以及m，然后进行下一步的操作 其中m和n是整数，ch\_0是一个字符

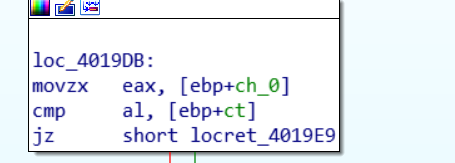
然后是一个switchcase 其中显然是switch(n)



当然这个时候我们应该选择其中一条路径分析，因为最终都可以走到ret

以case0为例子，显示给char型变量ct（声明时为byteptr）赋值了一个71h，然后将m与309h做比较若不相等则爆炸

然后看若相等跳转到的汇编块

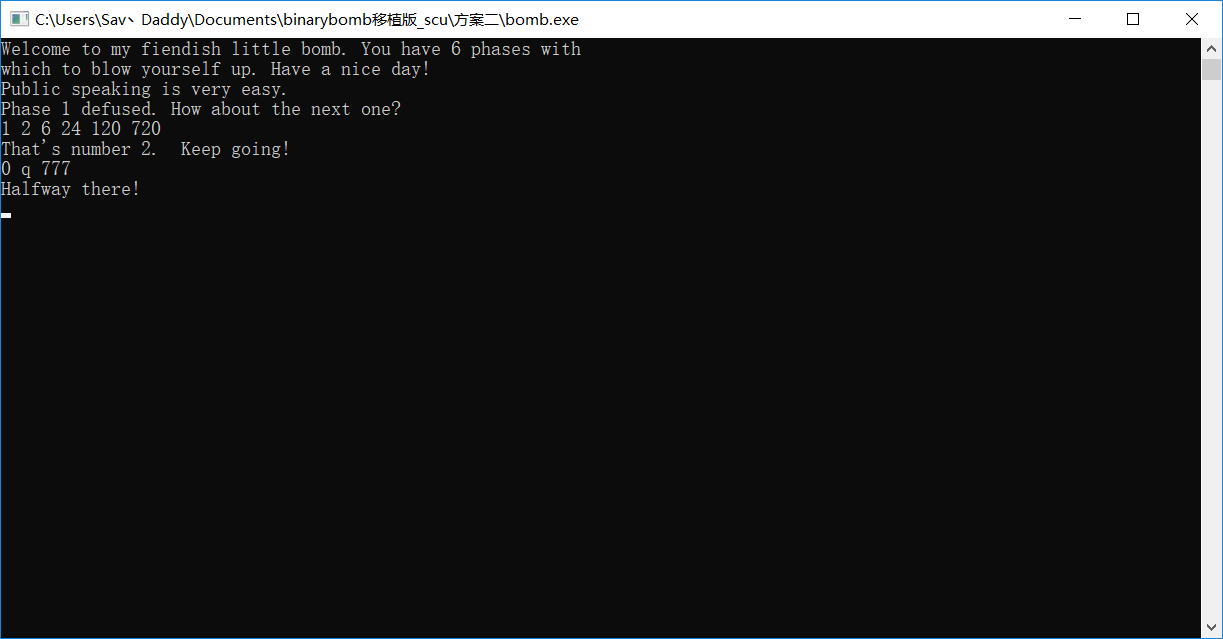


显然这是在对比ct的值和ch\_0的值，所以最后ch\_0的值是71h

71h对应的ascii字符是q

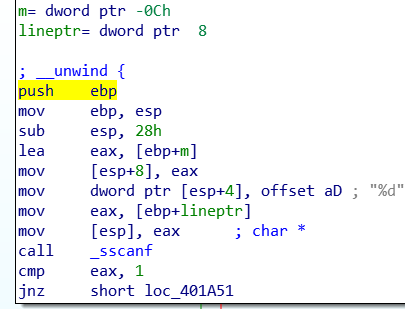
所以最后m = 309h ch\_0 = q n = 0

换算成10进制就是 0 q 777

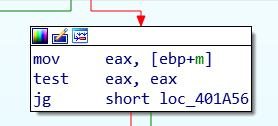


Phase 4

首先看到汇编，用sscanf输入了一个整数，存在变量m当中

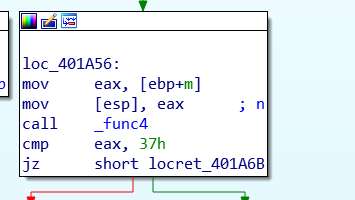


然后将m的值放入eax并且执行test操作，我们知道test实际上是进行按位与，所以

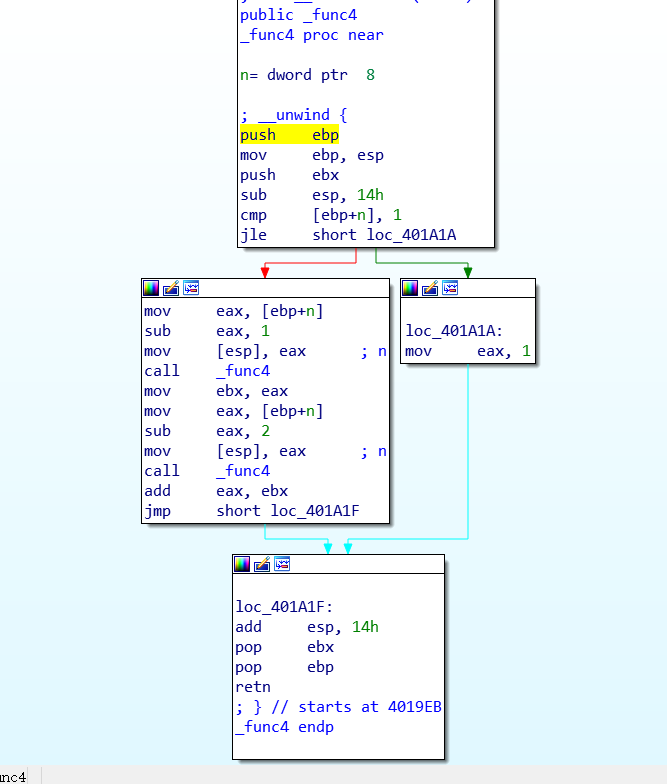


所以m不能为0

之后汇编代码调用了func4，并且比较func4的返回值与37h的关系



所以接着我们来看func4



从这里可以看到程序大概是在执行一个递归的调用过程，并且如果n小于等于1了之后返回一，再来看若n不为1，我们可以看到，程序先是将n - 1放入eax，然后传递给递归的func4，然后再同理n - 2 传递给finc4.，最后将其结果edx eax加起来，所以我们可以写出高级语言代码为：return func(n - 1) +func(n - 2)

然后写出对应的高级语言程序之后

int Fibo(int n) {

if (n <= 1) { return 1; }

else {

return Fibo(n - 1) + Fibo(n - 2);

}

}

int main()

{

int m = 3;

int result = 0;

while (true) {

result = Fibo(m);

if (result == 55)

break;

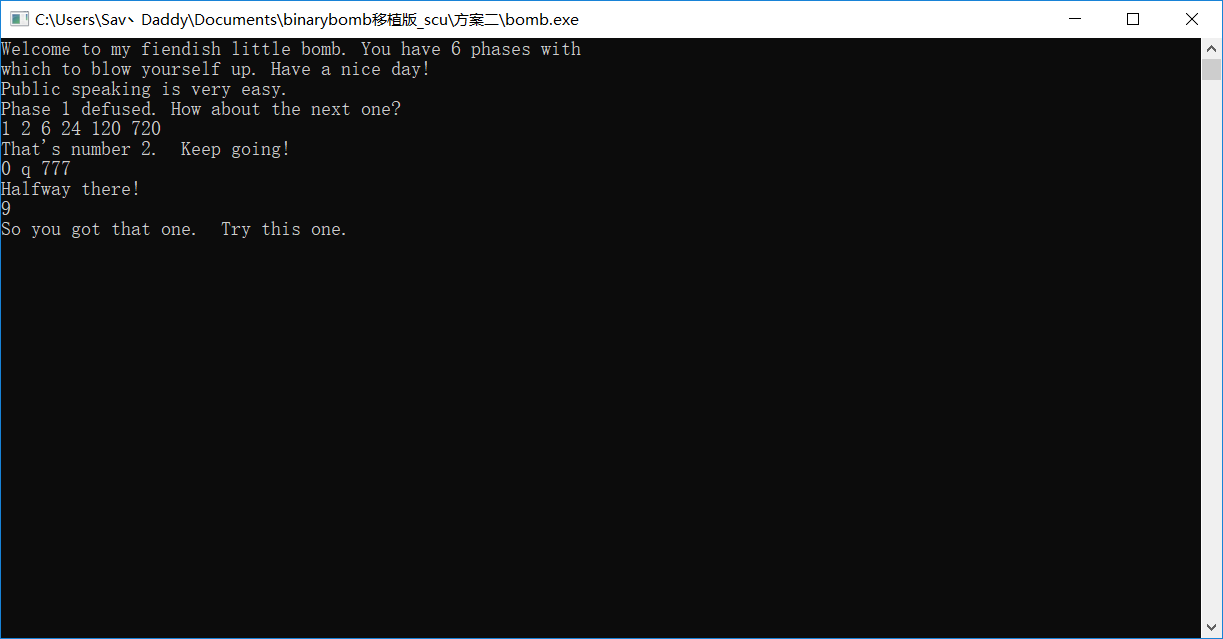
m++;

}

printf("%d", m);

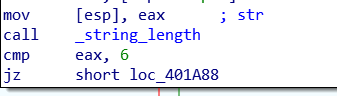
return 0;

}得到m的值为9



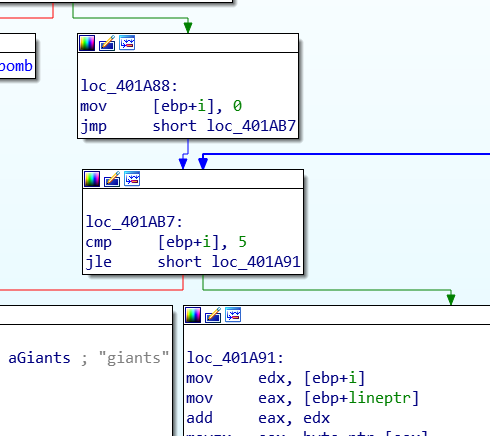
Phase 5

首先可以看到程序调用了strlen来获取输入的字符串大小



如果长度不等于6则引爆

所以需要输入一个长度为6的字符串

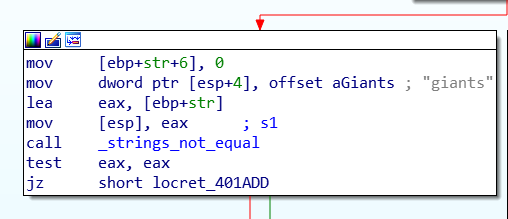


然后是一个赋值，现将i的值赋值为0然后进入循环，每次循环都要比较i与5的大小，如果大于5则退出

首先看循环体内部的操作



显然首先访问传入数组lineptr的第i个元素的地址然后把它的第一个字节无符号扩展放进eax，之后再对其前8个bit进行有符号扩展，然后将得到的结果与0fh做按位取与操作，这些汇编合在一起我们可以知道程序在做的工作是取得传入输入lineptr的第i个元素，将其转换为二进制之后取其前四位，然后将这个数作为下标去访问那个is开头的常量字符串，最后将结果放在str的第i位，之后i++

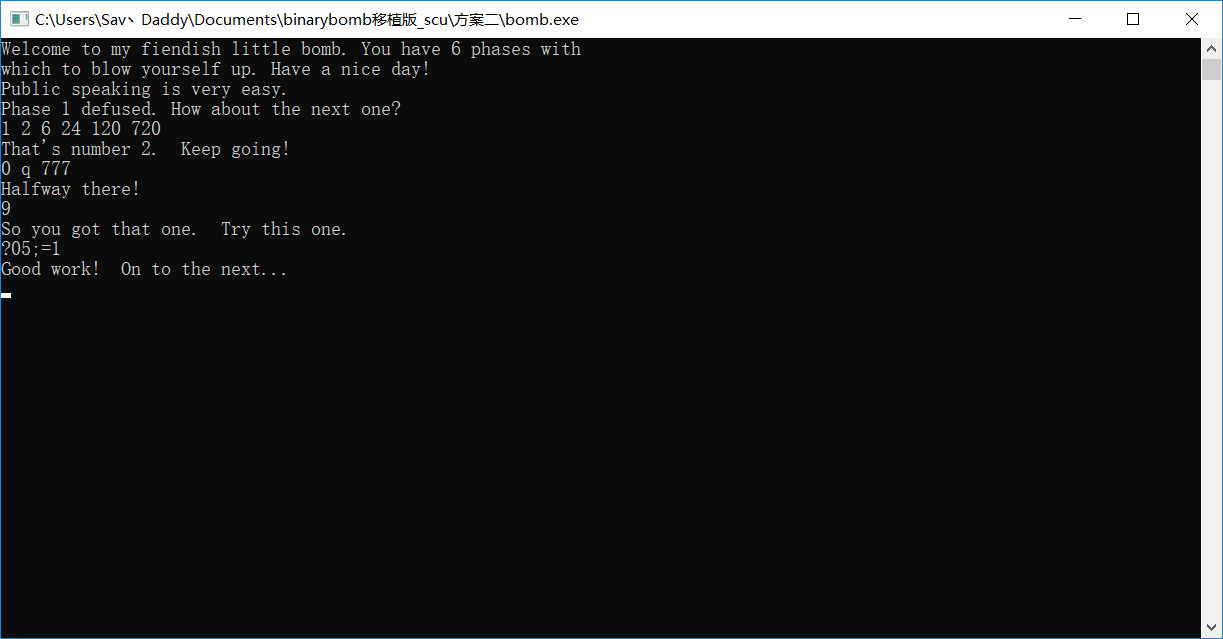


然后可以看到程序在循环过后再str数组后面加了一个空0，然后调用字符串比较函数判断str是否与giants这个字符串相等，如果相等则完成phase5.

所以由上面可以知道，我们需要第一个字符的ascii值前四位为字符g在字符串isrveawhobpnutfg中的下标位置，同理可以得到其他的字符，同时可以注意到，15对应的ascii字符是一个控制符，所以不能采用，11111（2）代表的也是一个控制字符，所以只能从111111（2）开始选择字符

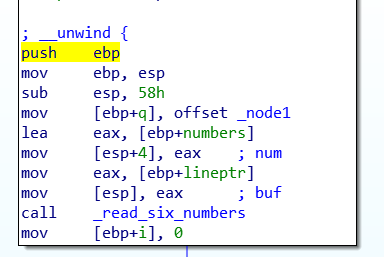
所以答案为：

?05;=1



Phase 6

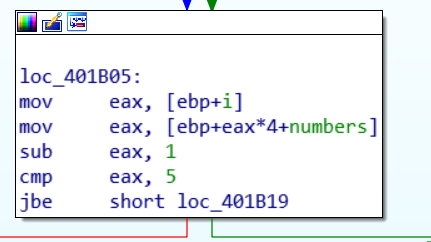
首先看到程序用与phase2相似的方式读入了六个数



然后将i赋值为0

接着开始第一个循环

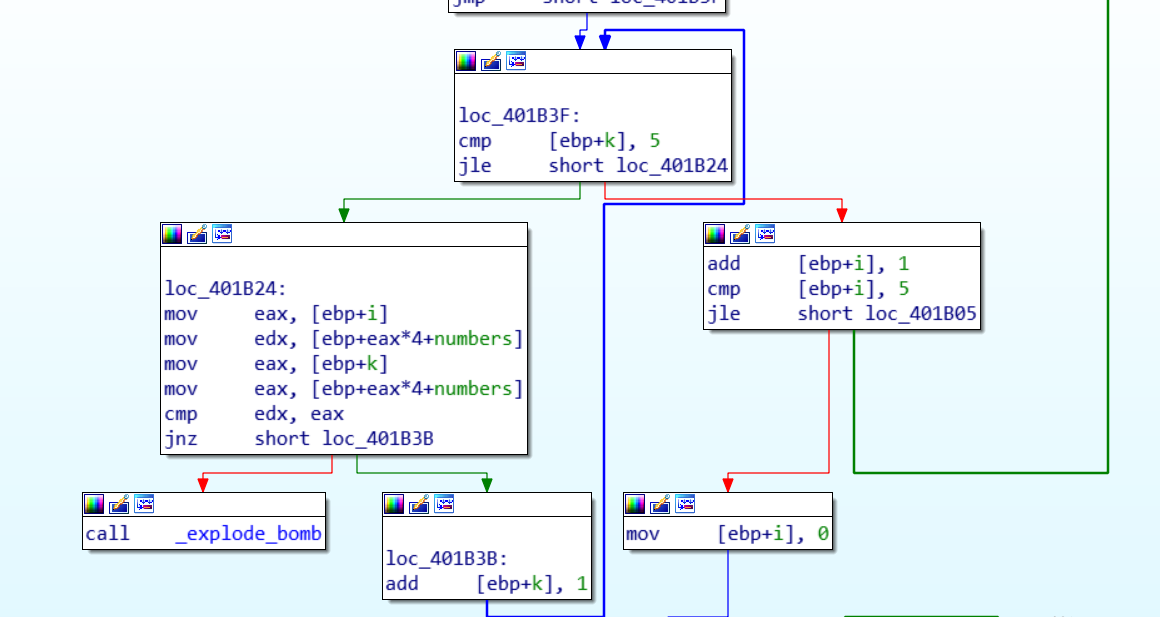
首先看循环的第一个部分，



循环首先是访问了numbers

数组的第一i个元素，然后比较这个元素减一的值与5，如果大于五则炸弹爆炸，所以五个数都应该是小于等于6的数，

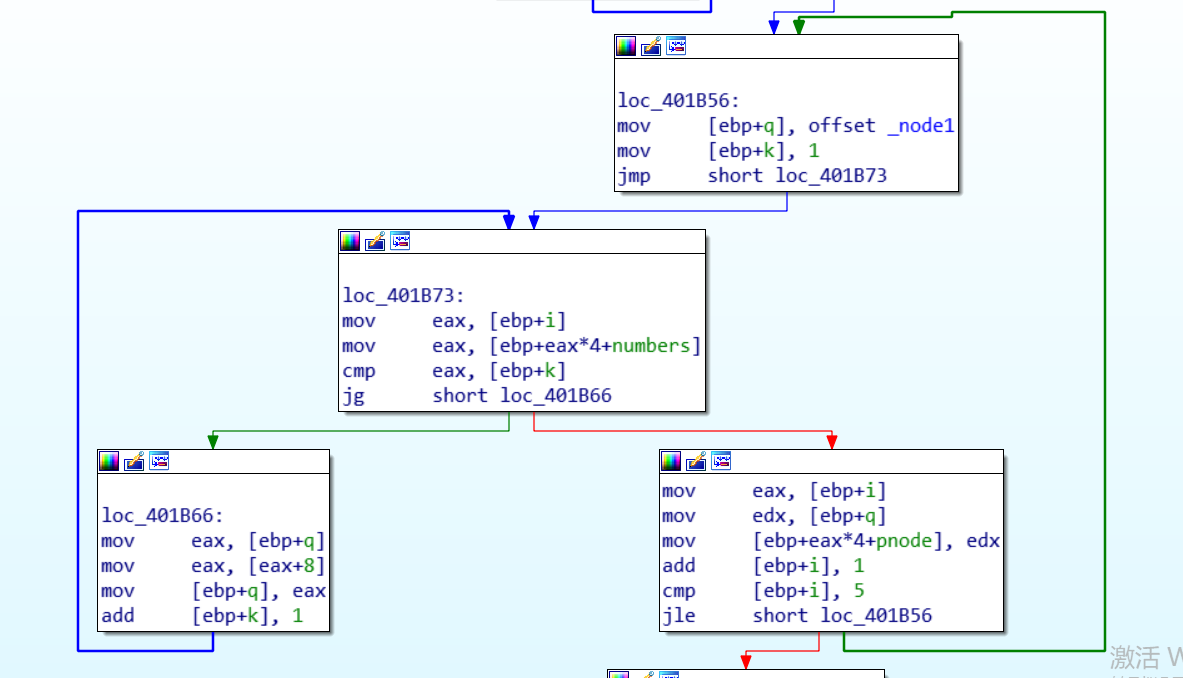
接着来看第二个部分



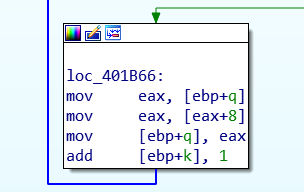
先是将k赋值为i+1，如果k小于等于5，则到左边继续执行

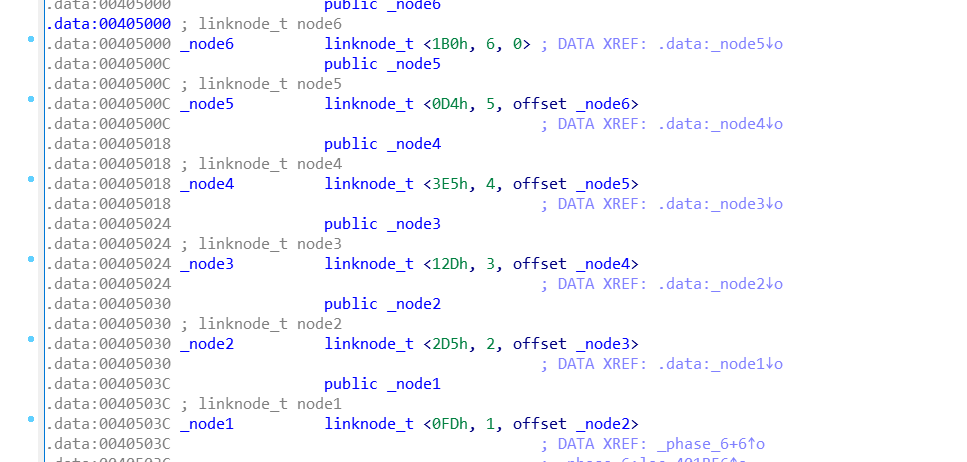
在左边，可以看到循环比较了numbers[i] 和numbers[k]的值，如果相同，则炸弹爆炸，所以numbers里面不能出现重复的元素，

这个循环过后i被赋值为0

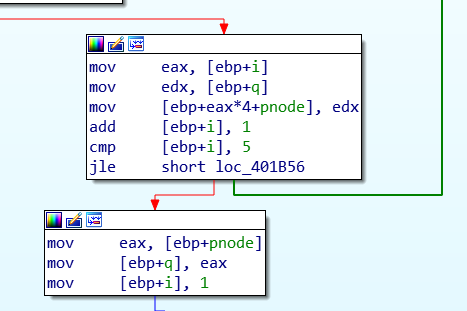


接着显示将p指向对象node1，然后将k值赋值为1，然后进入循环，首先比较numbers[i]与k的值，如果numbers[i] > k 则到左边继续循环

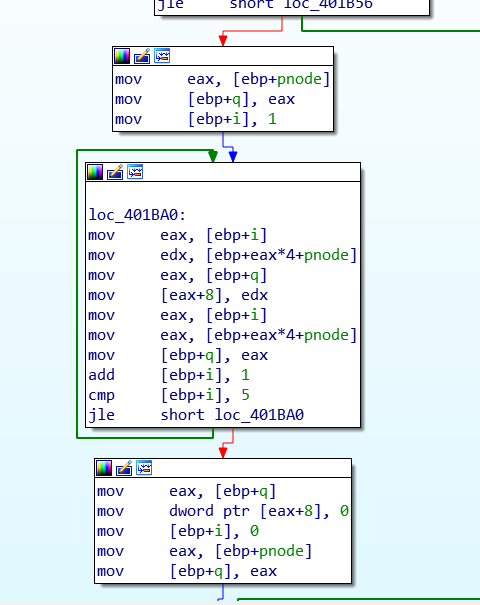




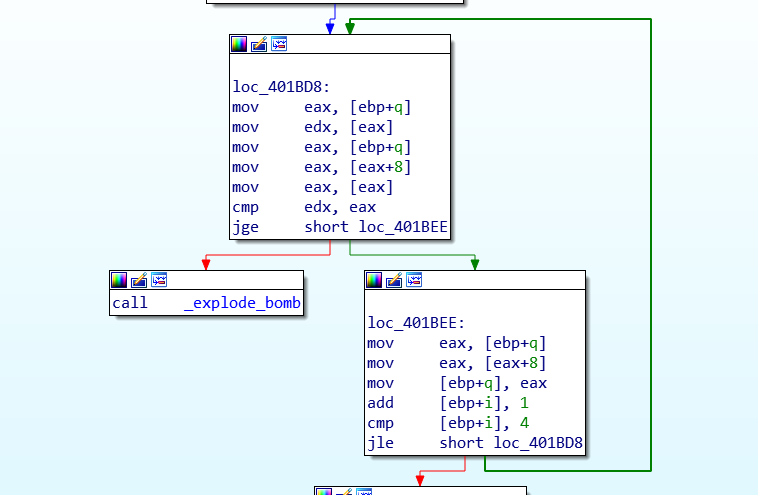
观察发现node是一个存储在数据段的链表，所以左边这些代码的功能是，将q = q->next,然后令k + 1，所以这些代码的目的是将指针p移动到当前numbers[i]里面的数字在的节点的地方



接着讲将这个节点的地址放在数组pnode当中然后继续循环



然后将q指向pnode的第一个元素，之后将i的值赋值为1，在这个循环里面，首先是将pnode数组中的第i个元素这职位q的next节点然后将q节点指向pnode[i]，我们可以看出这是在把链表按照输入数字顺序重新安排了一遍



可以看到这里最后的循环是检查q和q->next有没有保持降序关系，所以我们现在知道，输入的六个数是作为六个节点的索引，同使我们的输入要使得这六个节点的值降序排列

查看6个node的节点值我们可以知道，



六个数分别为：

4,2,6,3,1,5