ICS Lab2说明文档

16302010026 陈涛

**Phase\_1:**

根据函数名知道需要做字符串的比较，所以需要输入一个字符串，而且在调用函数对比字符串前，把一个地址存到了寄存器esi，通过打印地址里面字符串得到字符串：

****

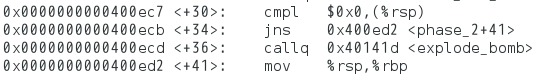
**所以第一题答案为**

**Houses will begat jobs, jobs will begat houses.**

**Phase\_2:**

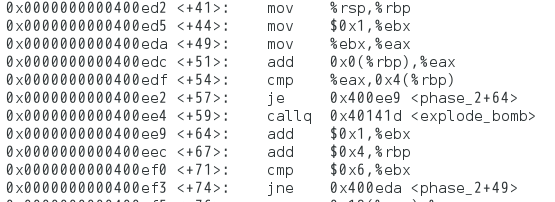
根据函数名<read\_six\_numbers>知，需要输入6个整数。

根据汇编：



知道第一个数为非负数

再根据汇编中的循环：



明白 e1+1=e2；e2+2=e3；e3+3=e4；以此类推，由于第一个数为非负数，所以我取第一个为1，则获得6个数为：1,2,4,7,11,16

**所以第二题答案为：**

**1 2 4 7 11 16**

**Phase\_3:**

发现在scanf函数前使用了地址0x4025af，输出地址：



可知第三题需要输入两个整数。

仔细看汇编，明白switch语句，且发现：

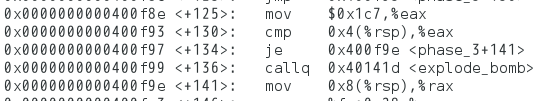


以第一个参数的值乘以8，加上地址0x402420存储的值为跳转地址。

输出地址0x402420的值，为4198286，即0x400f8e



根据汇编：



发现0x400f8e处直接将常数0x1c7与第二个参数作比较，相等即可。所以为了方便，直接将第一个参数赋值为0，那么直接跳到0x400f8e处，第二个参数赋值为0x1c7，即455即可。

**所以第三题一种答案为：**

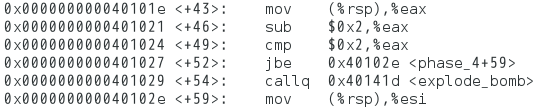
**0 455**

**Phase\_4:**

表示6题中被这一题坑的最惨（被参数位置的变化坑到）。

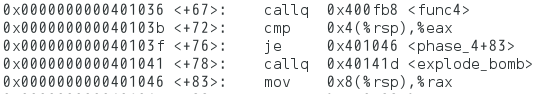
用第三题的方法知道这一题需要输入两个整数。

根据汇编：



发现其中一个参数是在2~4之间，由于前面的原因，一直以为这是第一个参数，所以一直炸。后面调整位置，第二个参数改为2~4之间，终于不再炸。

可是进入func4查看代码，发觉太烦，看不懂。



于是就在地址0x40103b那加了断点，获取func4的返回值。得到：第二个参数是2的时候，返回值66，输入3返回99，输入4，返回132.选了3作为输入。

**所以第四题的一种答案是：**

**99 3**

**另：在做完之后听说有secret phase，由于找不到入口，便向其他人询问了secret phase的密码，知道是在第四题的答案后面加字符串，**

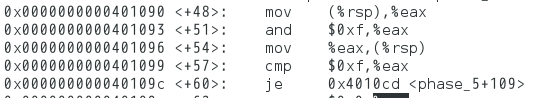
**所以第四题我的答案是：**

**99 3 DrEvil**

**Phase\_5:**

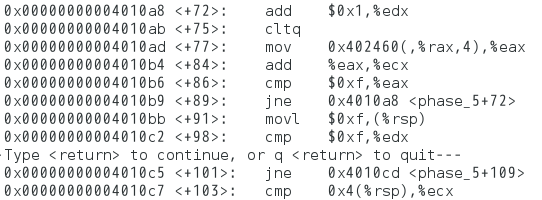
第五题查看地址0x4025af，得知本次需要输入两个整数。

根据汇编：



了解到第一个参数二进制表示的低四位不能为1111，而且第一个参数在与0xf进行按位与运算后将结果保存为参数，所以第一个参数只要其低四位的二进制不为1111即可，大小无影响，姑且设为0~14.

根据汇编：



了解到这里有一个循环，而且参数%edx从1加到了15，也就是说循环的次数必须等于15次，且在最后一次循环结束时，%eax的值必须为15。循环中，还将%eax中的值加入%ecx中，在循环结束之后与第二个参数做对比。

在循环中不难发现%eax的值在变化，每次都从不同地址中取值，通过打印这些地址中的值，发现是一个数组（起始位置为0x402460）中的值，且该数组为

{10,2,14,7,8,12,15,11,0,4,1,13,3,9,6,5}

于是采取逆向分析，将%edx=15，%eax=15代入循环逆向寻找答案，最终得到%eax的初始值为5，所以第一个参数取5即可，再将循环中的值相加，获得%ecx是115，即第二个参数取115.

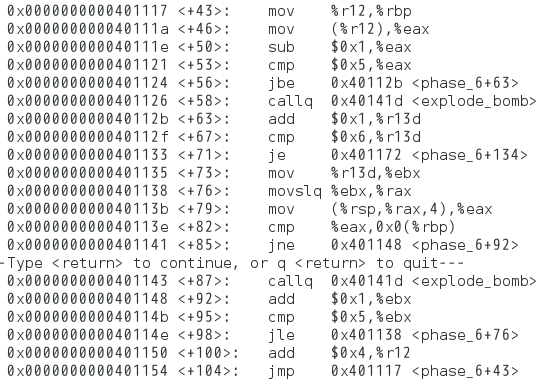
**所以第五题答案是：**

**5 115**

**Phase\_6:**

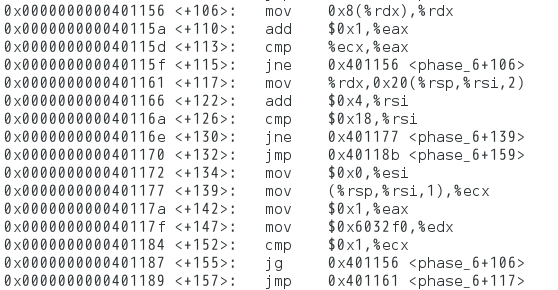
本题由函数可知需要输入六个整数。

根据汇编：



由这里的嵌套循环知：输入的六个参数各不相同，而且必须在1~6之间，所以可以确定六个参数为1,2,3,4,5,6.但排列顺序尚未可知。

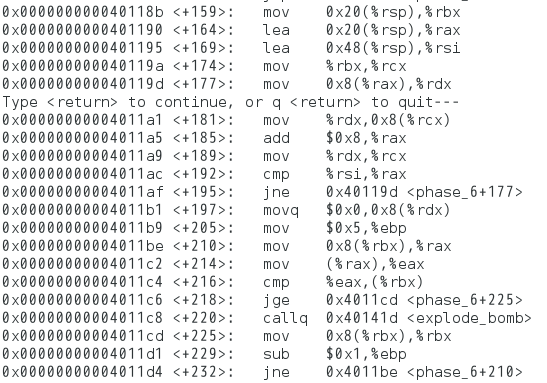
根据汇编：



知道每个参数会对应一个地址，并将地址存入栈中，但是由于参数先后次序不知，所以不知道栈中地址的先后顺序，通过这段汇编可以确定每个参数对应的地址，以及该地址所存储的值：

|  |  |  |
| --- | --- | --- |
| 参数值 | 对应的地址（地址被存入栈） | 地址对应的值 |
| 1 | 0x6032f0 | 817 |
| 2 | 0x603300 | 888 |
| 3 | 0x603310 | 548 |
| 4 | 0x603320 | 347 |
| 5 | 0x603330 | 826 |
| 6 | 0x603340 | 729 |

再根据汇编：



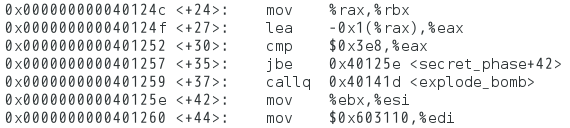
第0x401119d到0x4011af行是将栈值做一些变化，而接下来则是比较一下值的大小。这段汇编合起来的效果就是表明：参数的先后顺序是按照他们对应地址所对应的值的大小排序，顺序是从大到小。根据上面的表格可知：值从大到小为：888>826>817>729>548>347,所以对应参数顺序为：2,5,1,6,3,4。

**所以第六题答案为：**

**2 5 1 6 3 4**

**Secret\_phase:**

其实原先无法进入这个题目，后面向大佬求教了一下，发现了进入这题的密码。



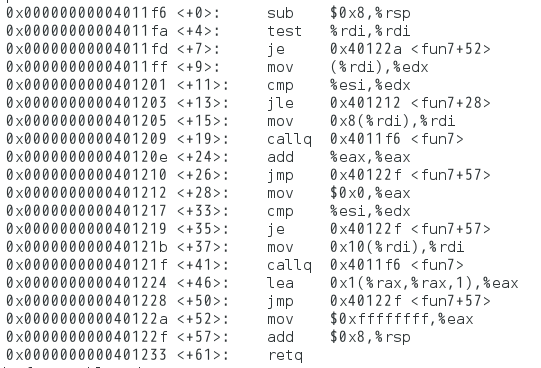
根据这段汇编可以知道，输入的参数在1~1001之间，且打印出地址0x603110的值为36。

根据汇编：



在出fun7之后，返回值与2进行比较，所以知道，参数在进入fun7之后返回值为2。

查看fun7汇编：



发现是一个递归函数，第一次进入fun7，由于返回值需要2，所以第一次进入fun7之后，发现参数必须小于36，且只能通过0x401209来调用fun7，有可能使返回值为2。且调用fun7之后，返回值乘以2输出，所以第二次调用fun7的返回值为1

通过这种逆向分析，总共经过3次调用fun7，发现参数为22（地址0x603170）时，符合条件。

**所以secret\_phase的答案为：**

**22**

**综上，答案整理为：**

**Phase\_1:** **Houses will begat jobs, jobs will begat houses.**

**Phase\_2:** **1 2 4 7 11 16**

**Phase\_3:** **0 455**

**Phase\_4:**  **99 3 DrEvil**

**Phase\_5:** **5 115**

**Phase\_6:** **2 5 1 6 3 4**

**Secret\_phase:** **22**