# CSAPP LAB之bomb，汇编、反汇编与gdb调试

不得不说，此题的创意很好，不仅全面的考察了阅读汇编，使用gdb调试的能力，而且还有一定的趣味性，谁不想成为一个成功的“拆弹专家”呢？

Bomb是一个二进制炸弹，拆除他需要分别输入六条命令，一旦输入错误就会爆炸。在bomb.c文件中可以看到有6个phase，每个phase的输入是一条字符串，所以关键就是通过分析phase的内容来判断我们应该输入的字符串是什么。

理论上通过阅读汇编代码就可以找到答案，前3题或许可以，后边的题目就有点难度了。熟练使用gdb也是这个lab很重要的一个目的。

官方提供了gdb常用命令手册：<http://csapp.cs.cmu.edu/3e/docs/gdbnotes-x86-64.pdf>

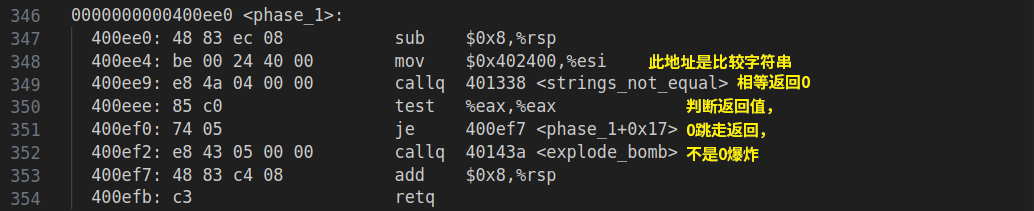
下面开始“拆弹“。

首先需要将bomb二进制文件反汇编，得到汇编代码。下面的命令将bomb反汇编并保存在file.s中。

objdump -d bomb> file.s

### phase1：

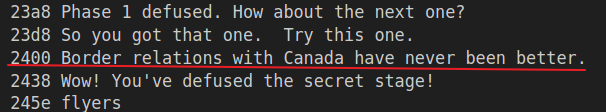
在file.s中搜索phase\_1找到第一个phase。



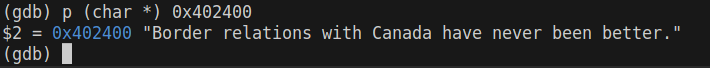
第348行的$0x402400就是我们要比较的字符串首地址。通过下面的命令可以获取bomb中的所有字符串：

strings -t x bomb >string.txt

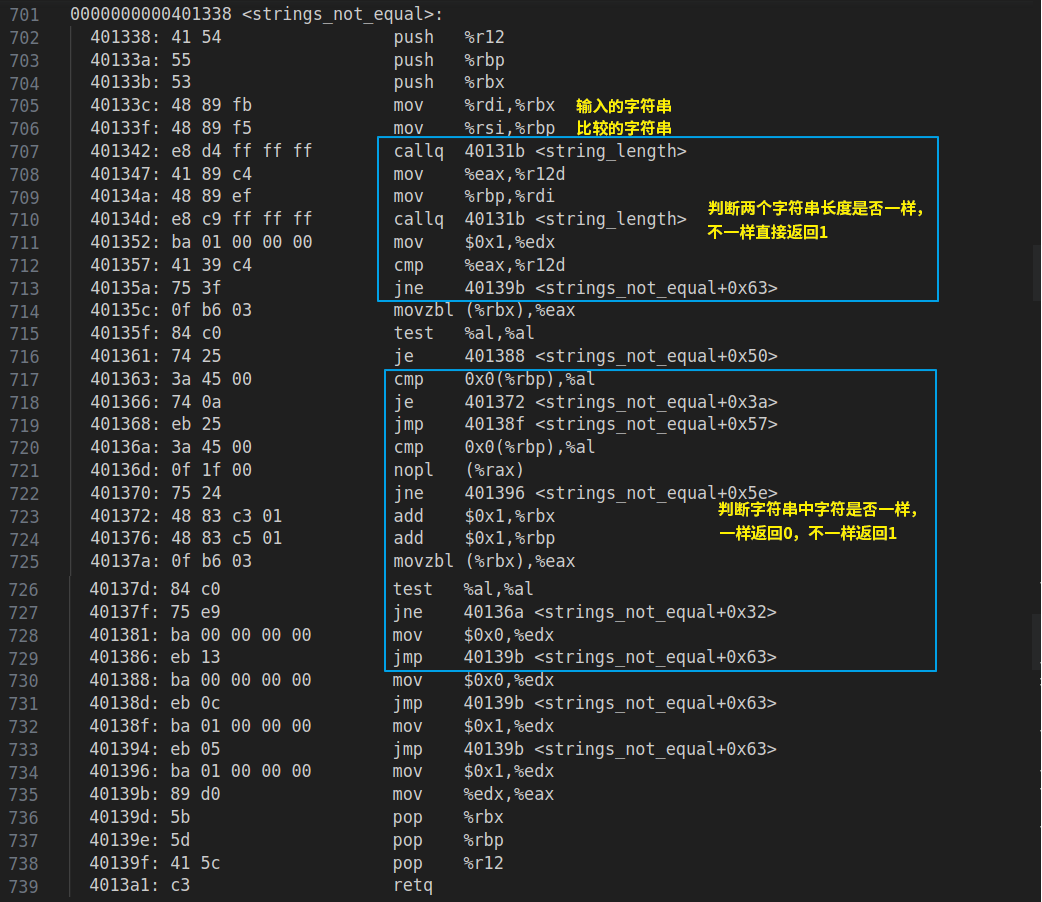
-t可以显示字符串离程序开始地址的偏移。找到2400：



我们也可以使用gdb打印出这个地址的字符串：

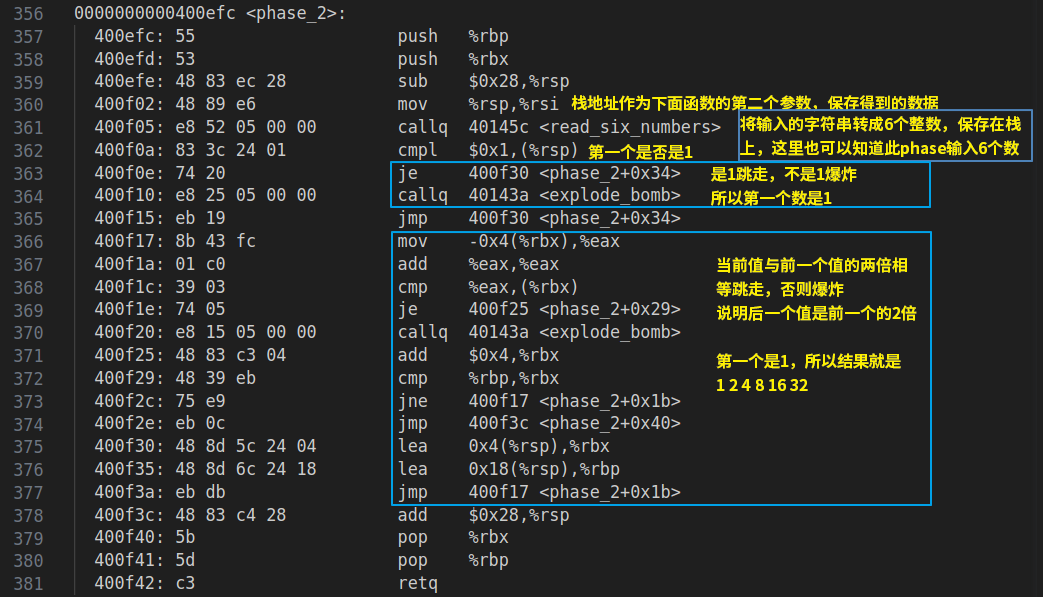


我们基本可以推测就是输入这个字符串，不过保险还是看看strings\_not\_equal函数。



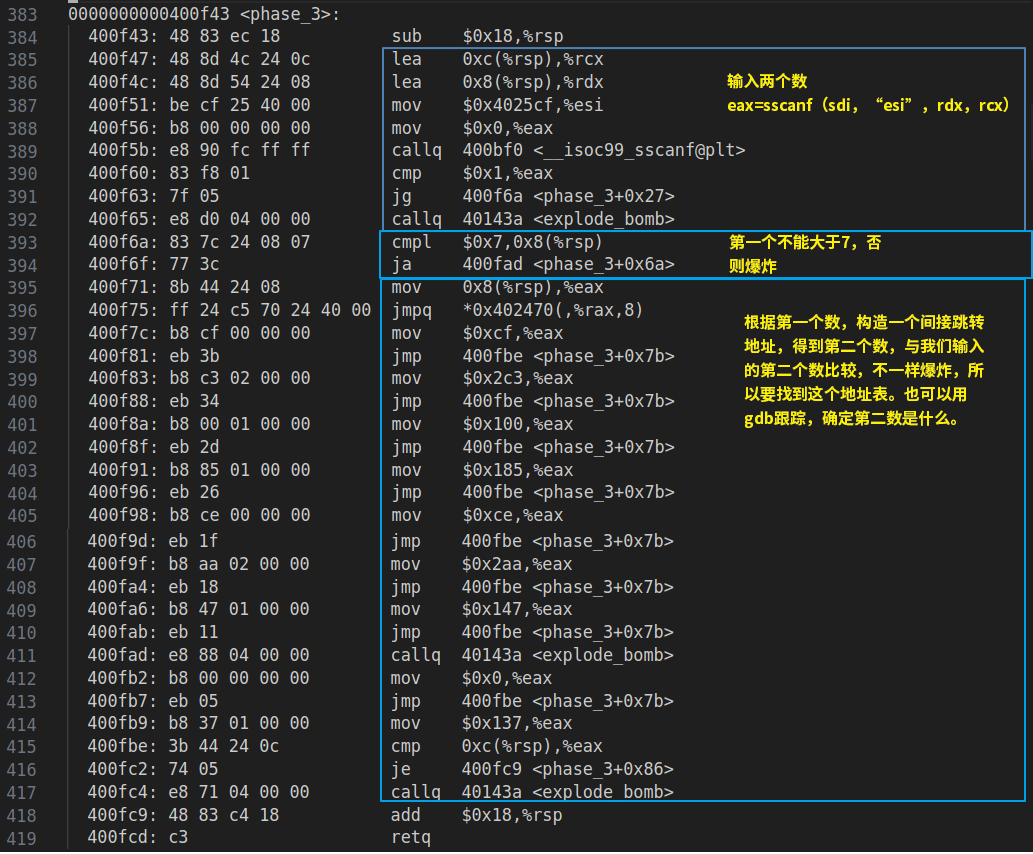
至此，我们得到phase1的密码就是我们在地址0x402400找到的那个字符串。

### Phase2：

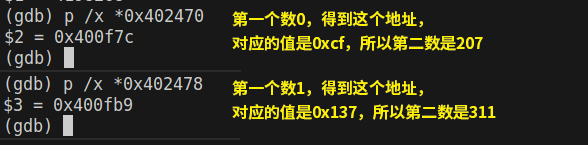


这个也不难，答案是1 2 4 8 16 32。

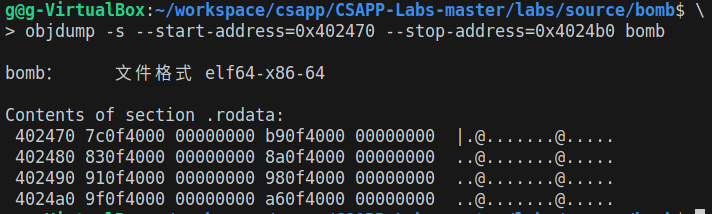
### Phase3：



可以用gdb打印地址处的内容，如下



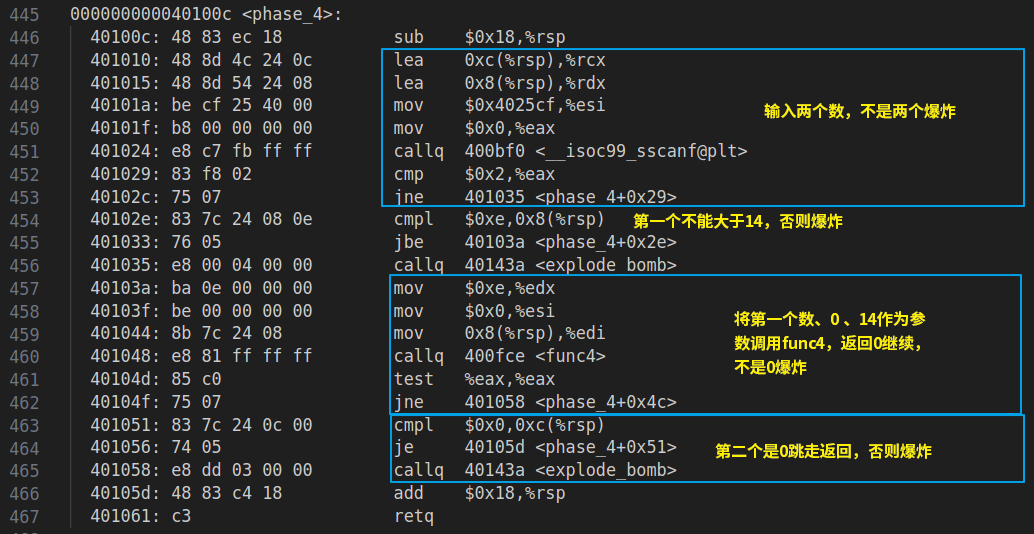
也可以用objdump把整个区域的内容打印出来



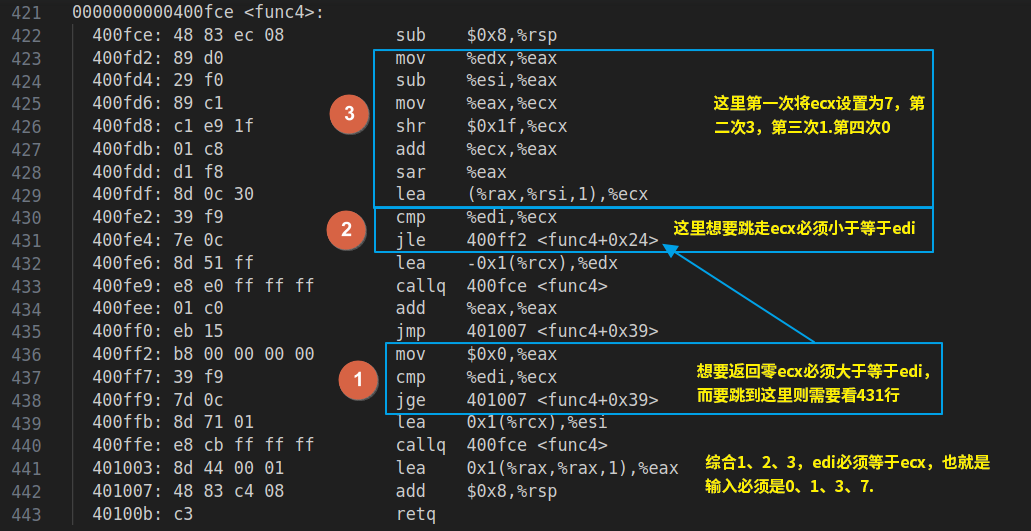
一查就知道跳哪了，所以此题有8组答案。

0 207、1 311、2 707、3 256、4 389、5 206、6 682、7 327

### Phase4:

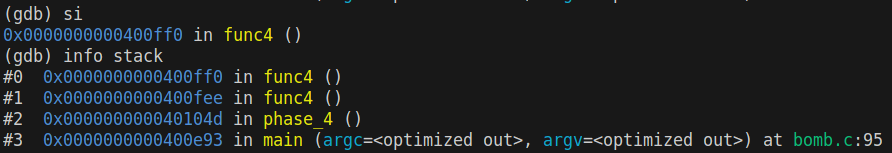


从上边可知第二个数必为0，第一个数要看func4函数，如下，这个函数有递归调用，还是比较绕的，粗暴的方法是用0-14试，找出返回0的输入，非常锻炼gdb的使用。

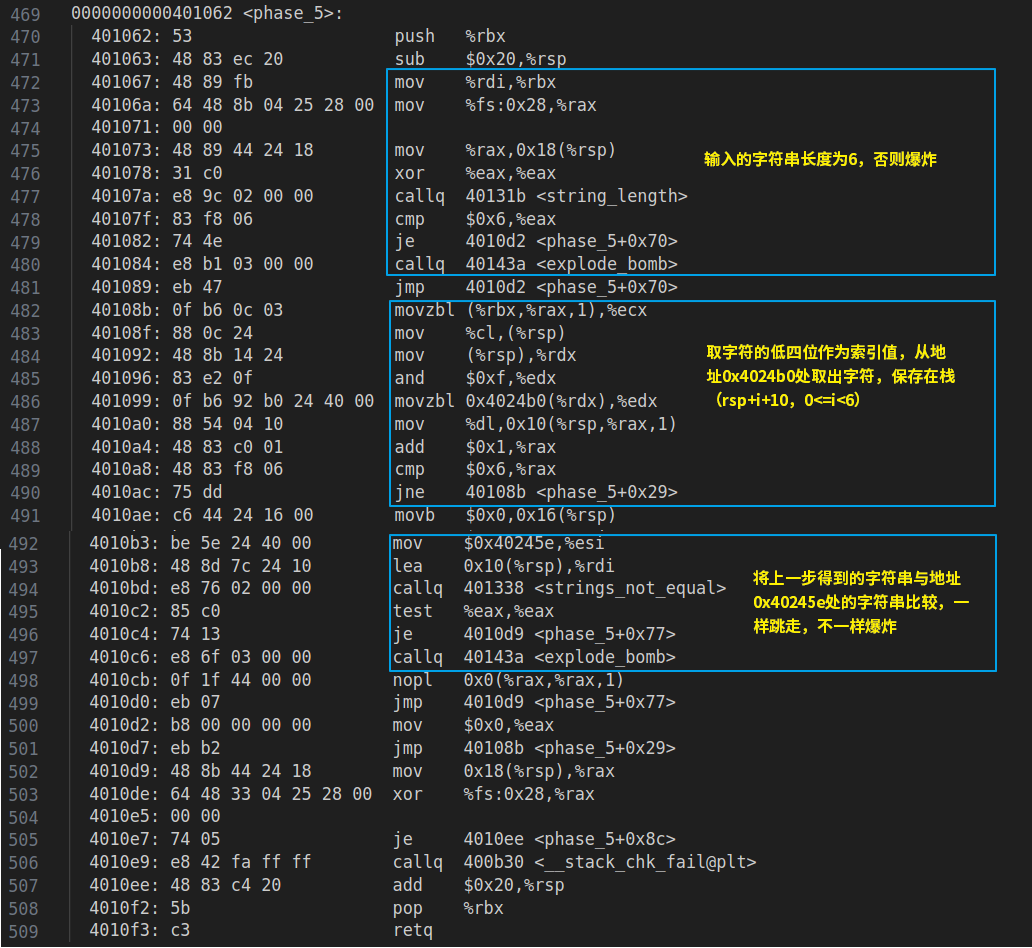


所以此题的答案是0 0、1 0、3 0、7 0。

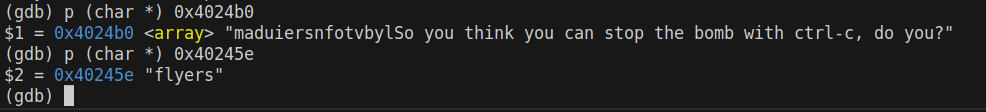
想要讲清楚调用逻辑比较麻烦，这里就不罗嗦了，感兴趣的大家自己琢磨吧。可以用gdb的si一步步走，用info stack查看调用栈的情况。



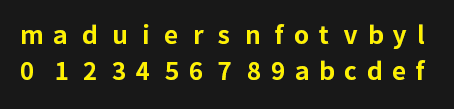
### Phase5：



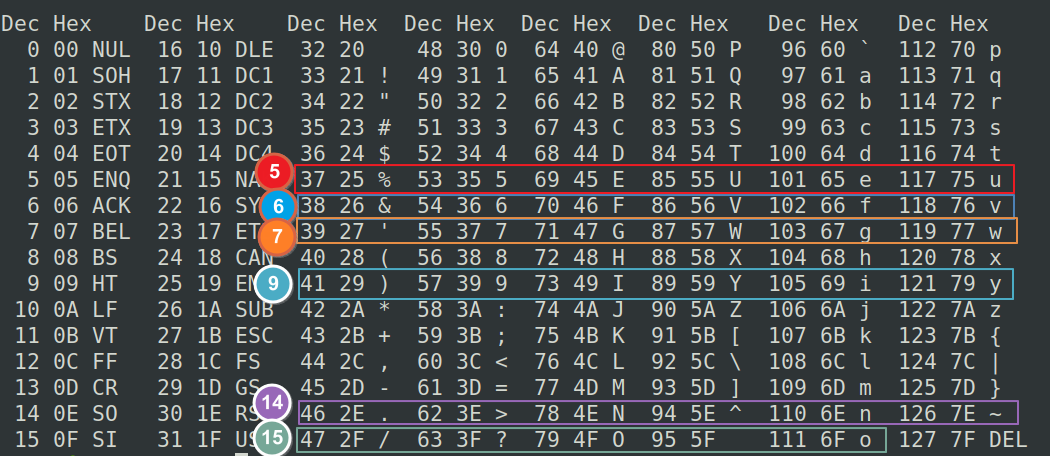
我们先看看两个地址的字符串是什么



这里看到比较的字符串是“flyers”，我们要从上面那堆“maduiersnfotvbyl中把这几个字符按照”flyers“的顺序提取出来，可以看到索引值是9 15 14 5 6 7



但是输入的字符串必须是6个字符，取字符低四位作为索引，所以查看ascii码表



9可以是 )、9、I、Y、i、y；

15可以是 /、？、O、\_、o；

14可以是 .、>、N、^、n、~l

5可以是 %、5、E、U、e、u；

6可以是 &、6、F、V、f、v；

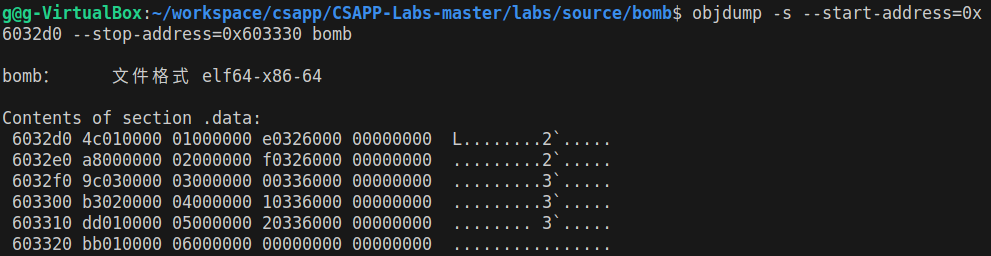
7可以是 ’、7、G、W、g、w；

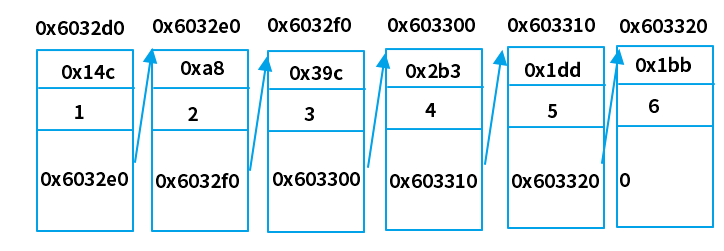
所以答案有很多种组合，每个数字挑一个字符就可以了，比如9?>567，这是最简单直观的。

### Phase6：

这个还是难度很大的，代码很长，而且各种指针，还有链表结构，搞得人很头疼。第4部分的理解非常重要也最难。画了个简要的流程图供大家参考，希望对理解有所帮助。

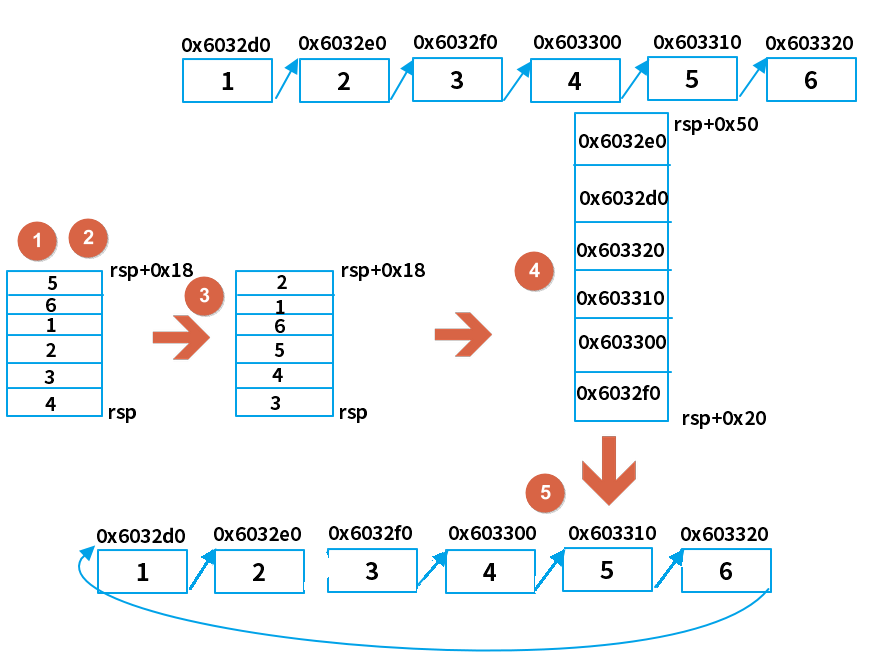
我们先来看看这个链表，如下：





链表头是0x6032d0，0x8(0x6032d0)就是下一个成员的地址（\*next指针）。理解了这里理解第4部分会容易一些。





此phase目的是让链表按照其第一个成员的大小从大到小的顺序排列，顺序如下：

0x6032f0，链表第3个成员，7-3则对应输入4 ；

0x603300，链表第4个成员，7-4则对应输入3；

0x603310，链表第5个成员，7-5则对应输入2 ；

0x603320，链表第6个成员，7-6则对应输入1 ；

0x6032d0，链表第1个成员，7-1则对应输入6 ；

0x6032e0，链表第2个成员，7-2则对应输入5；

所以此题的密码是4 3 2 1 6 5。

这个lab就像解密游戏一样，挺好玩的，不过个人表达能力有限，很难把代码逻辑讲的很调理，敬请谅解。最后，欢迎大家交流讨论。