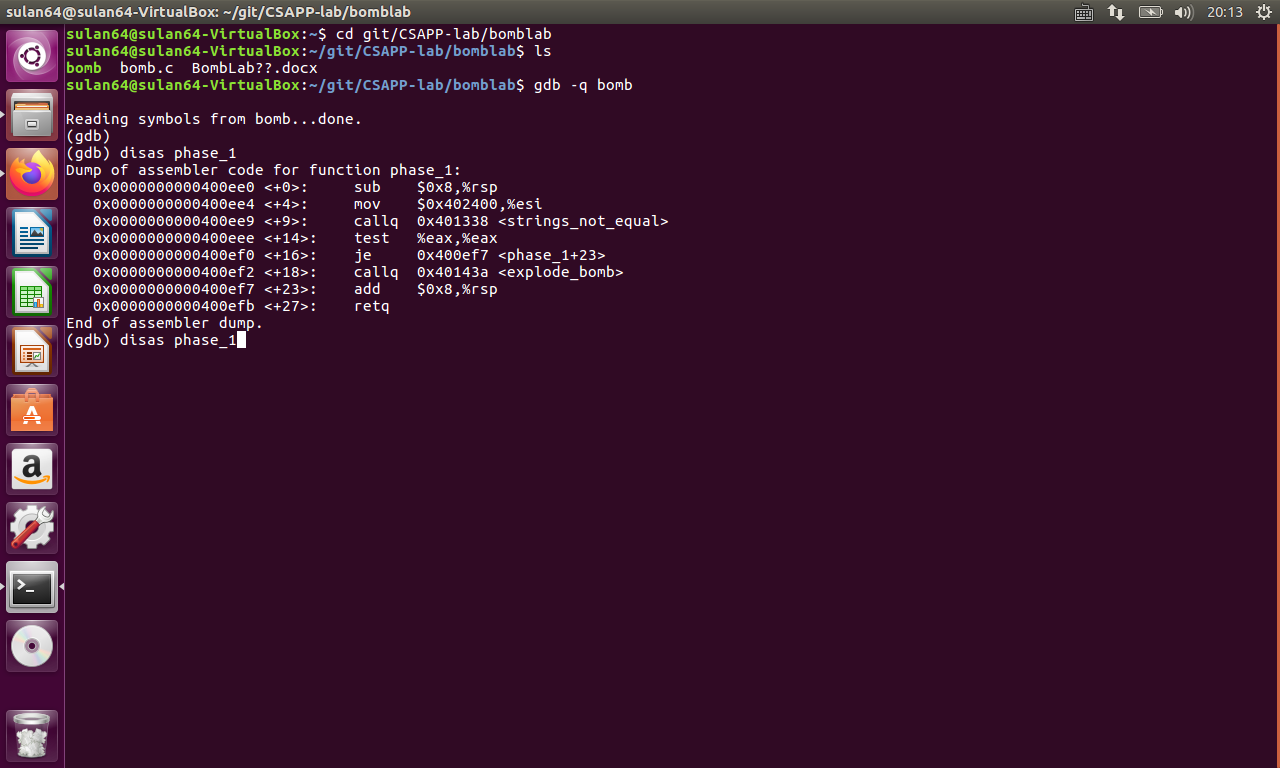
CSAPP-bomblab

思路：通过gdb获得汇编代码，通过课程学习的反汇编知识，观察寄存器和内存中的信息，得到密钥。

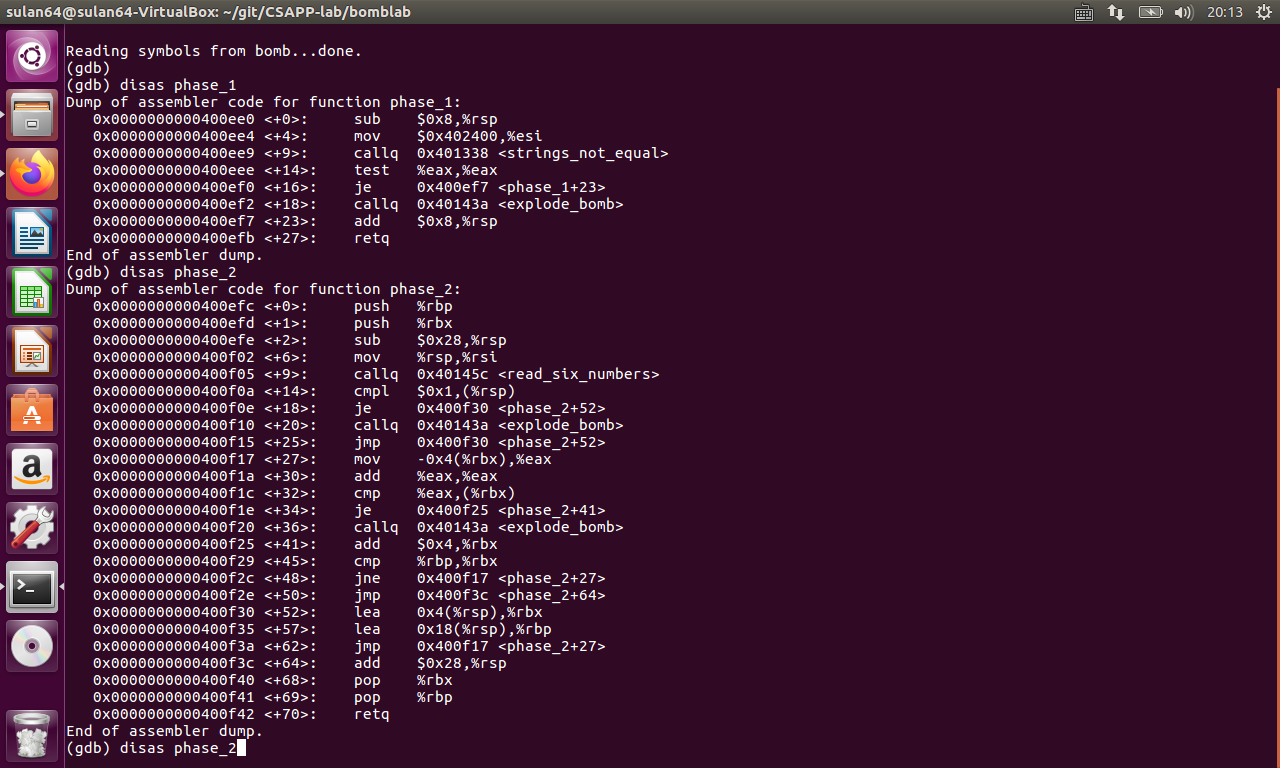
**Phase\_1**



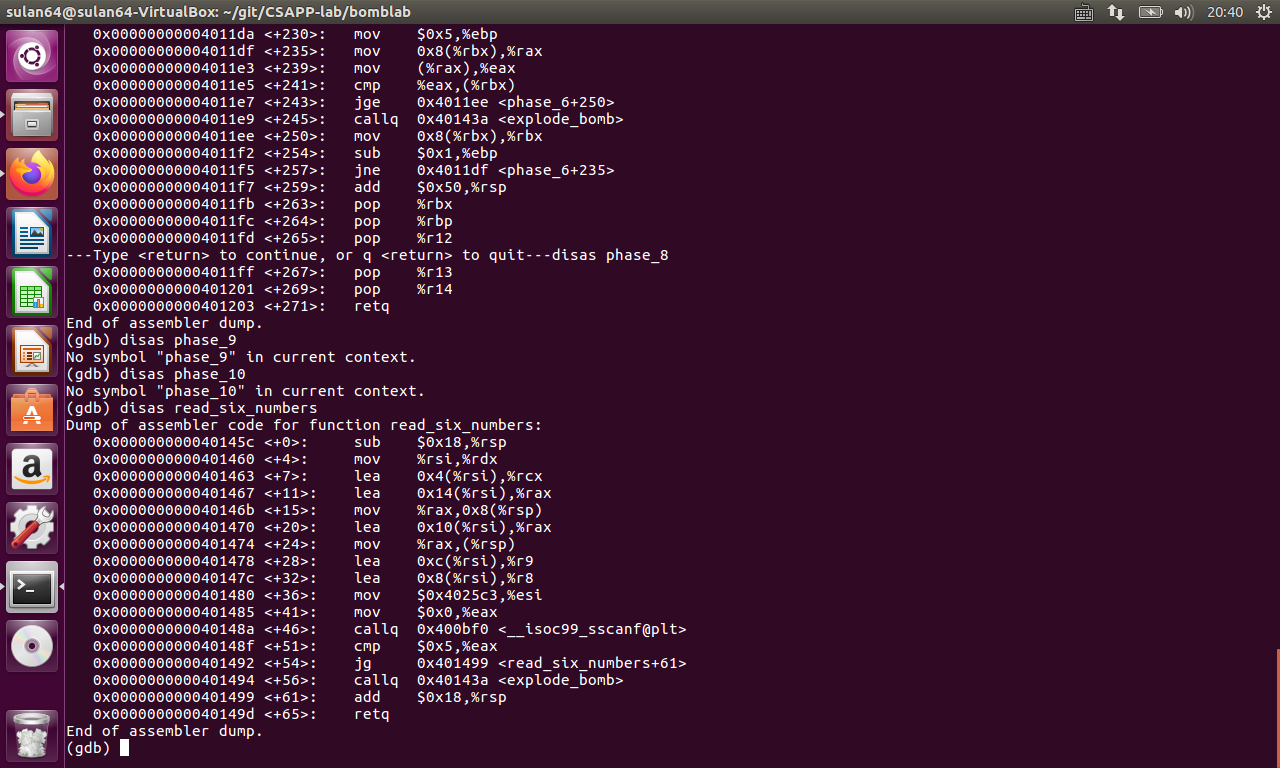
这个题可以猜到，调用函数strings not equal,判断字符串相等。于是我们输入x 0x402400 访问内置字符串，得到密钥：

Border relations with Canada have never been better.

**Phase\_2**

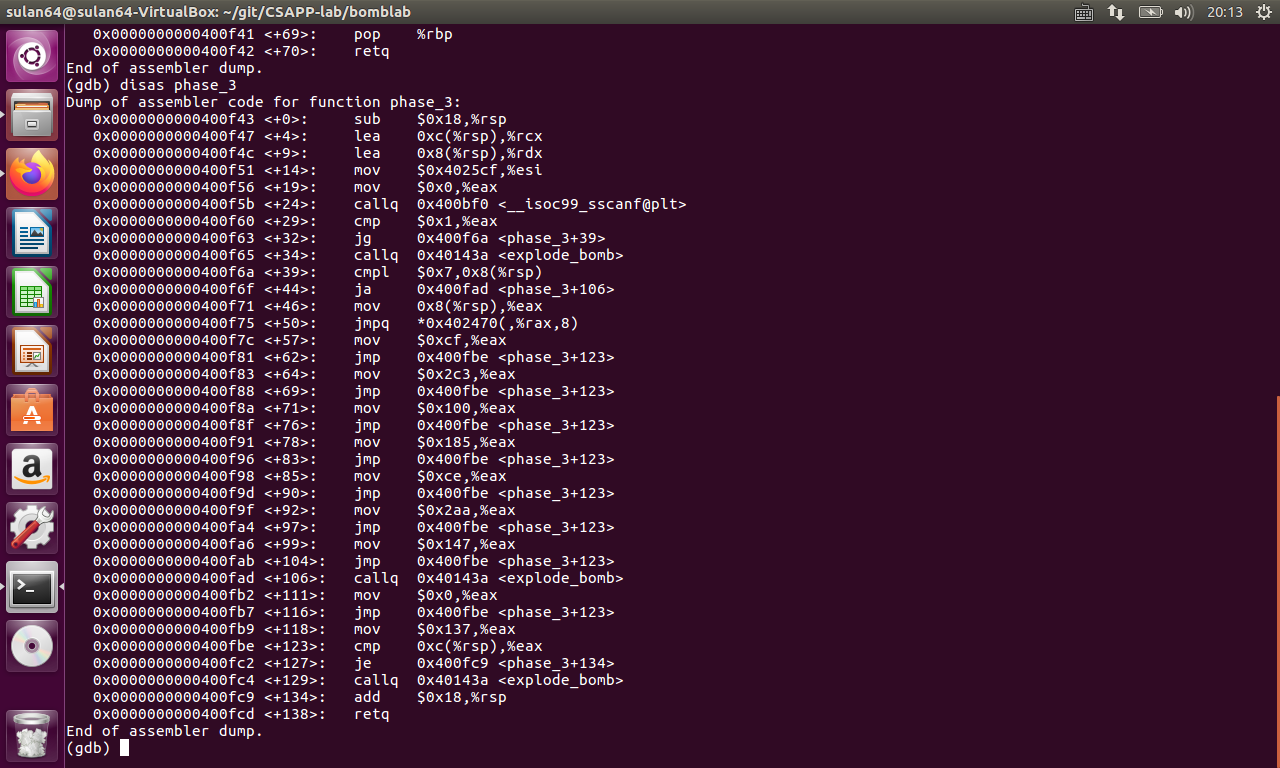


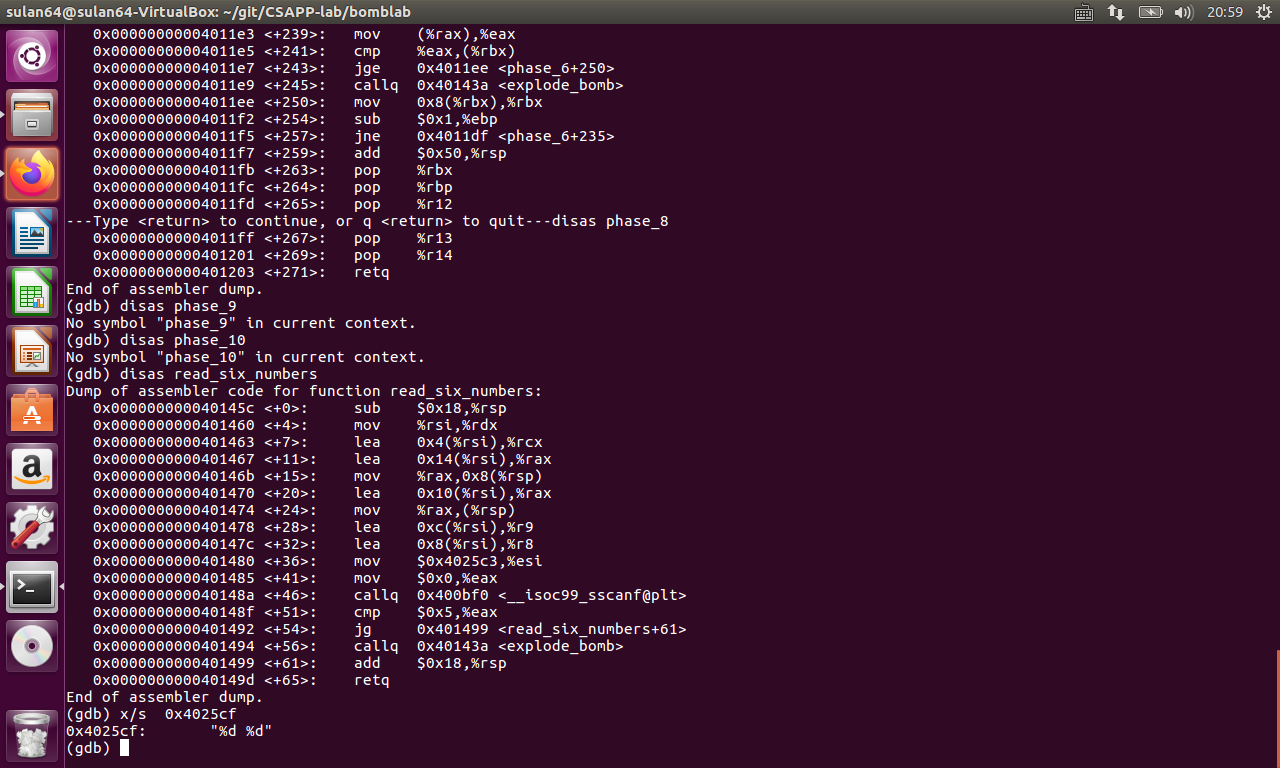
上一题取巧了。。其实没事，反正所有题都是一样的套路，无非是考反汇编能力。这道题调用了一个函数，目测是输入六个数字。看看read\_six\_numbers的代码：

****所以和我之前想的一样，是输入六个整型数字，回到phase\_2,黄框表示第一个数字是1，绿框表示把寄存器内数自身相加，所以是二倍。很简单，输入一个由1开头的等比数列：

1 2 4 8 16 32

**Phase\_3**

****

从这道题开始复杂起来了。和phase\_2一样，是接收数字，看一下：****

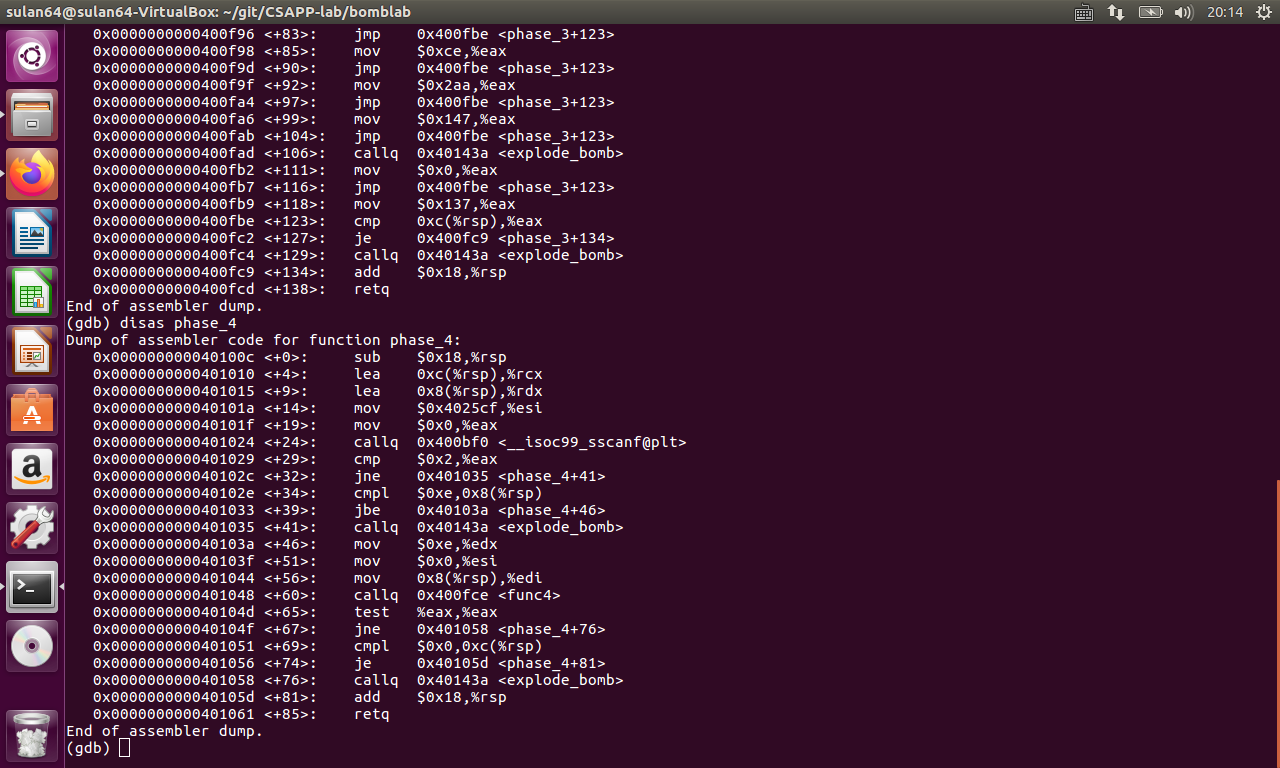
嗯，两个整型。

黄框是课上讲的选择语句，所以意思是，先给出一个值，再比较%eax中的值和他给定的内存中的值是否相等，只要不相等，就call explode\_bomb。一共有七种情况。所以我们找到任一个内存中的值就行：

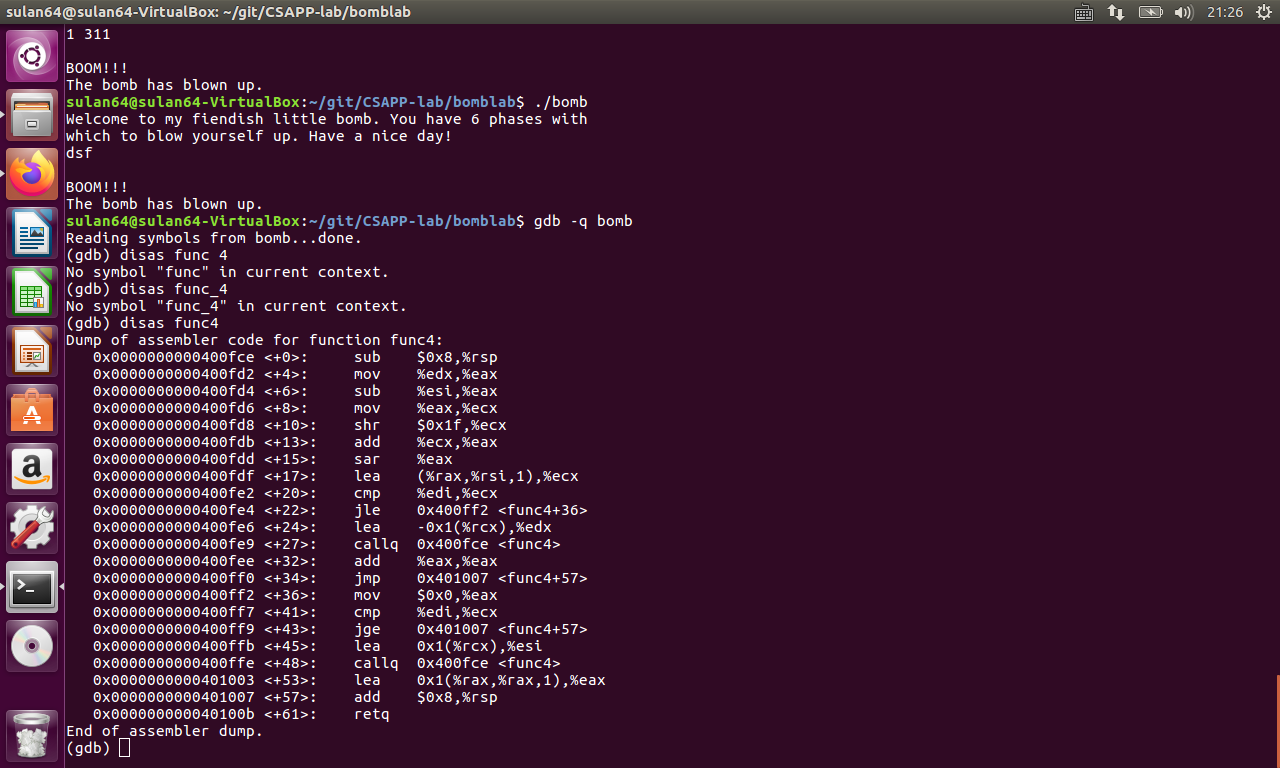
Print 0x2c3

1=707 但是好像不对？难道哪里出错了？

**Phase\_4**



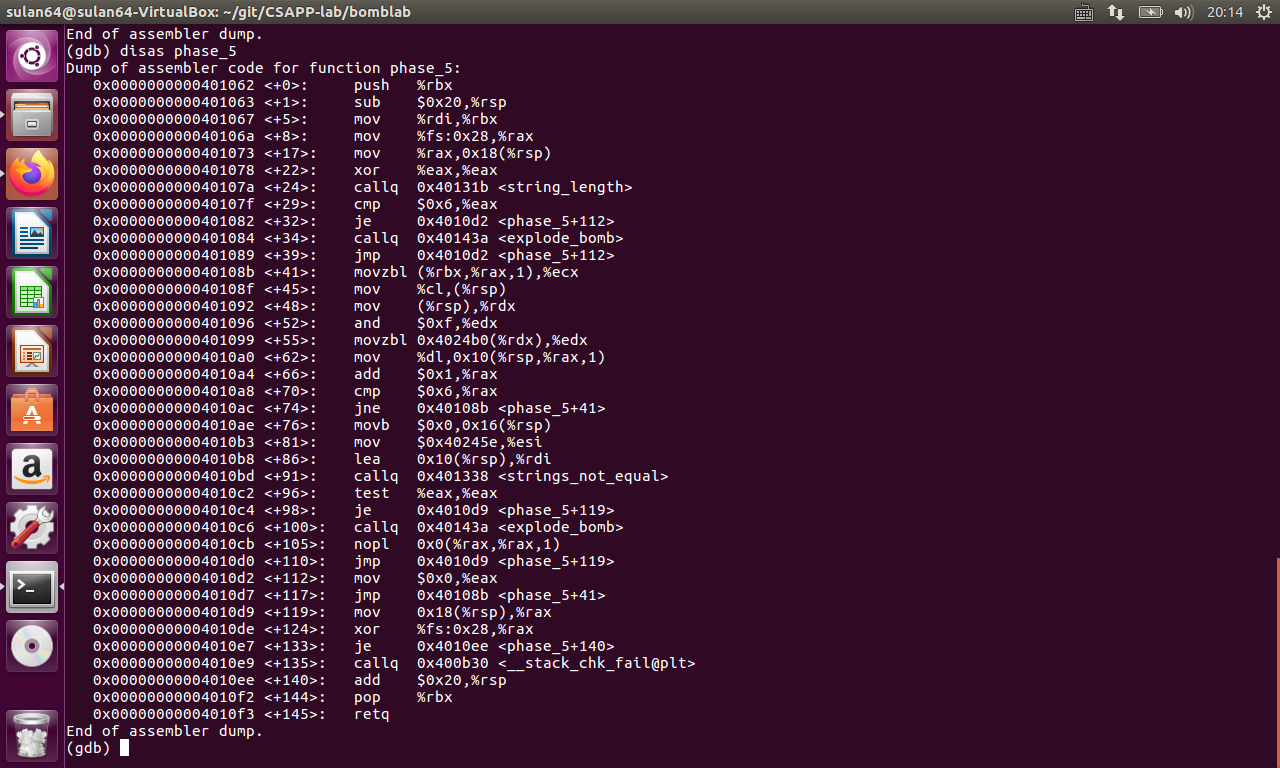
一样，先找func4：

****

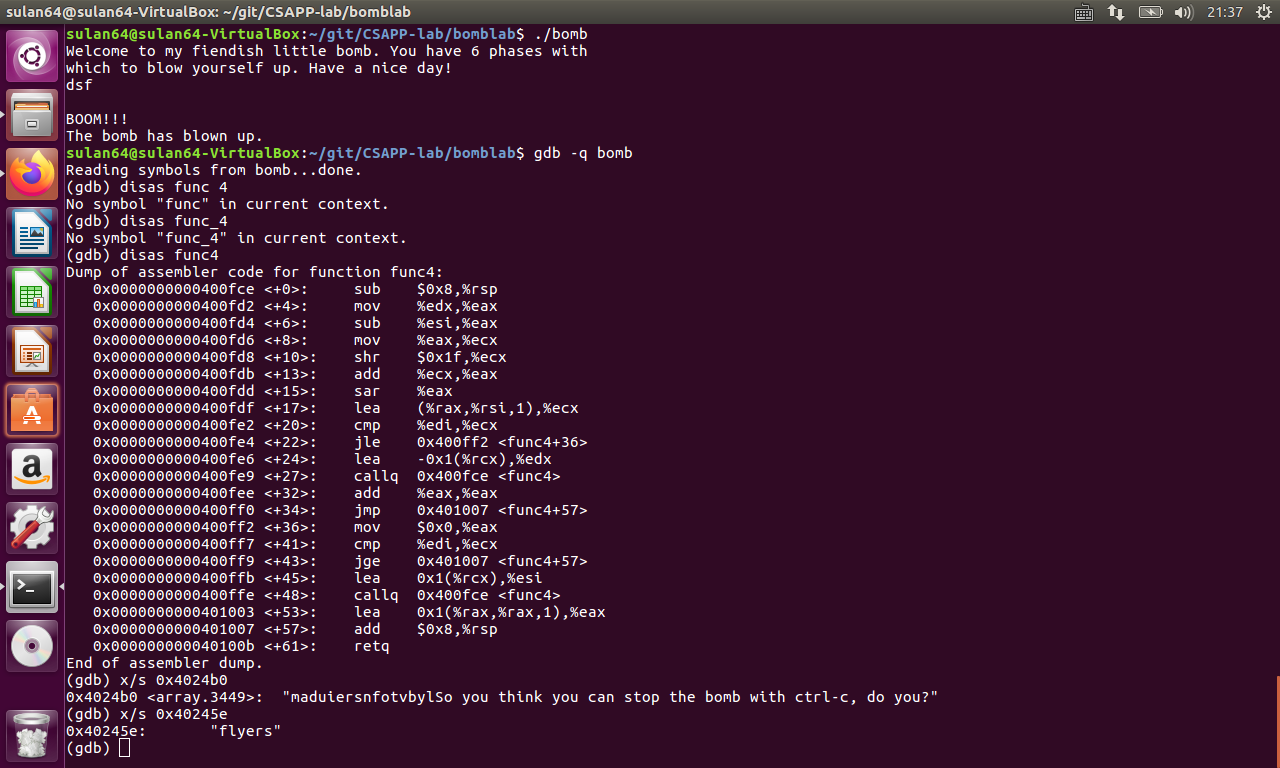
和3一样，输入两个数，这个题有点搞，看我框的地方，意思是，只要返回值不为零，就call bomb,那么我们只要输入的第二个值为0就行了？输入1 0，好像通过了

1 0

**Phase\_5**

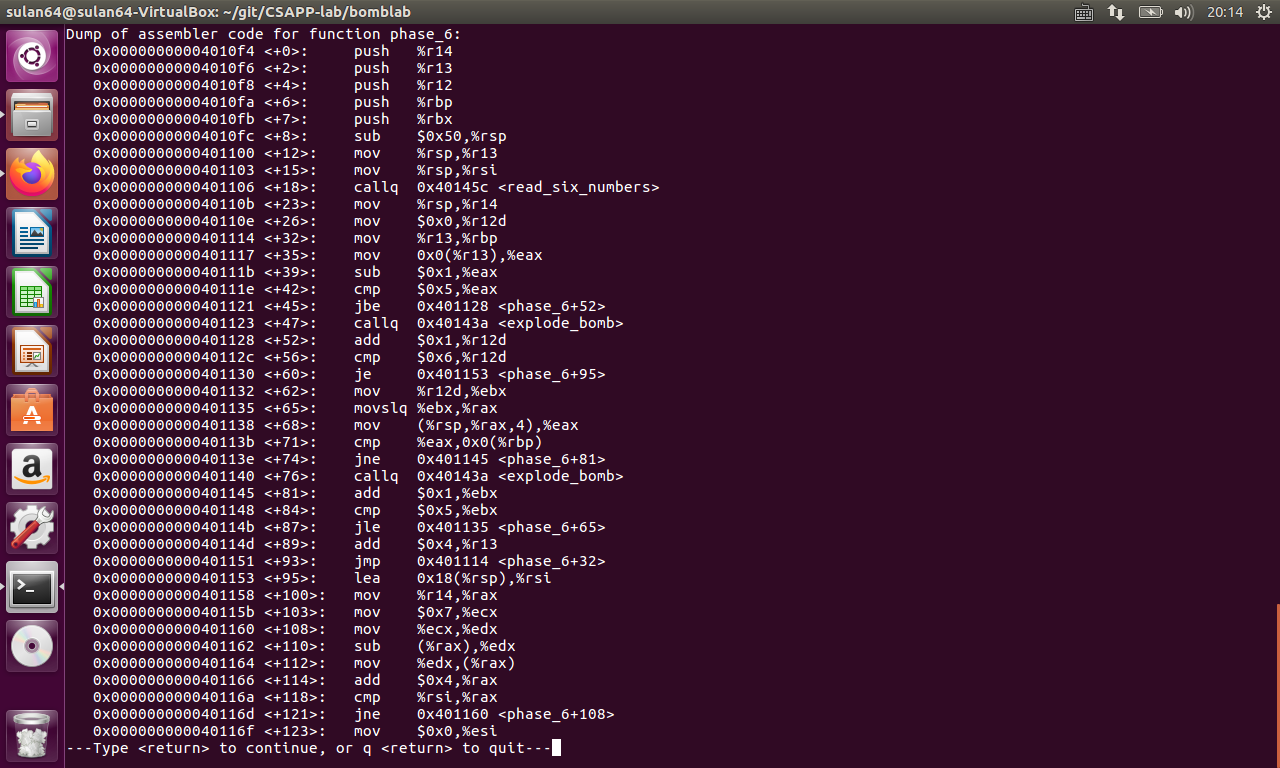
****

如图，调函数string-length，收到6位字符串；同时看到两个地址：

****

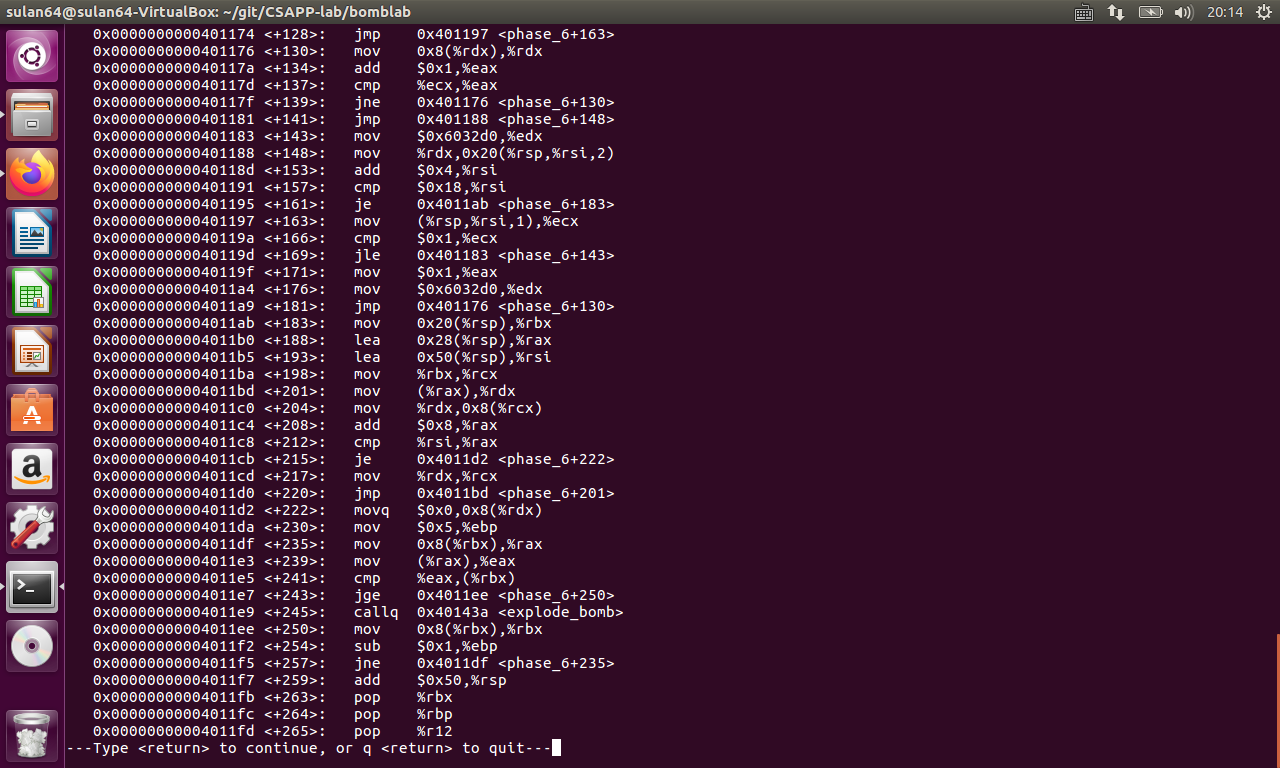
Ok,题目提示我了，要能够把输入的每个字符的低４位组合起来，使其能够成为flyers。查ASCII码，抽取出这个数字序列：9,f,e,5,6,7，再反过来找6个字符，over：  
ionefg

**Phase\_6**

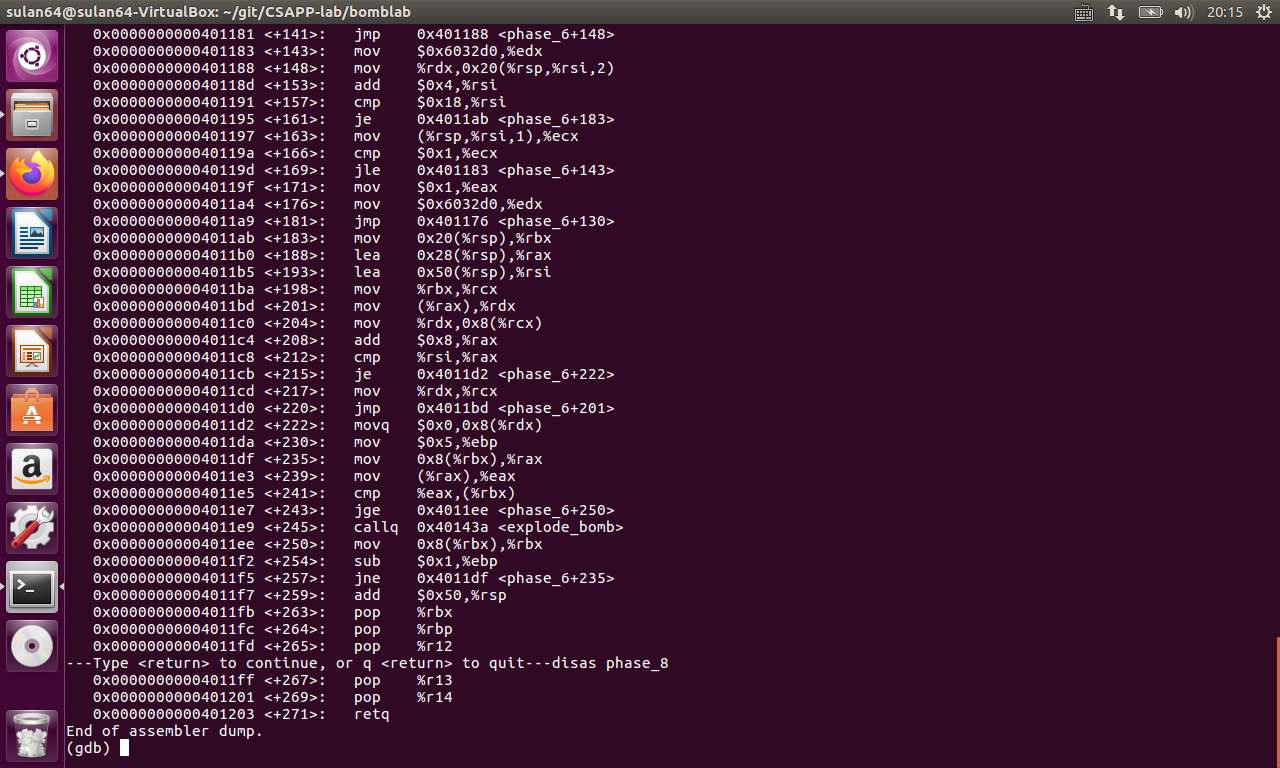


这个代码好像和链表有关，我得去看看数据结构了。。实在太复杂，参考答案都看不懂，先把作业交了，之后我再慢慢反汇编吧。

**Phase\_7**



**Phase\_8**



还有一个函数，但是不知道怎么开启。