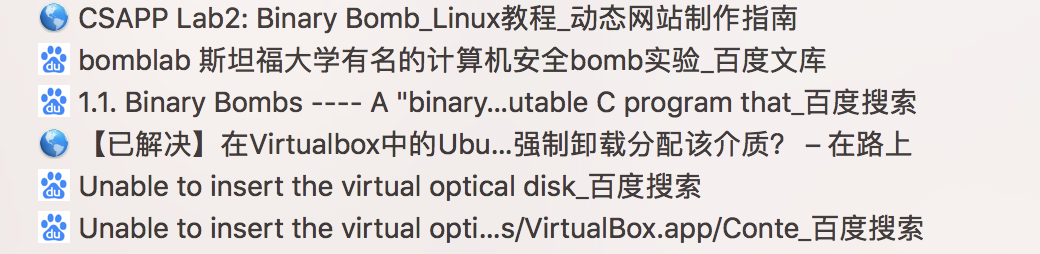
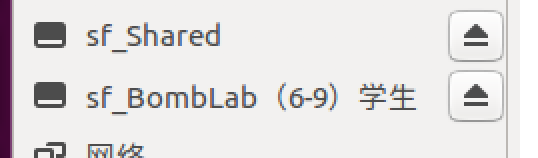
**第六周日志**

完成1、2、3关

2018年4 月 10日

第六周周二中午：把文件拖进ubuntu环境，由于不能直接拖进去，所以一节课都在查资料该怎么建立共享文件夹，设置挂载点等，最终成功。

以下是我的搜索记录和最终共享文件夹截图



2018年4 月 10日

第六周周二下午：

找到反汇编代码对应的phase1

08048f61 <phase\_1>:

8048f61: 55 push %ebp

8048f62: 89 e5 mov %esp,%ebp

8048f64: 83 ec 18 sub $0x18,%esp

8048f67: c7 44 24 04 5c a1 04 movl $0x804a15c,0x4(%esp)

8048f6e: 08

8048f6f: 8b 45 08 mov 0x8(%ebp),%eax

8048f72: 89 04 24 mov %eax,(%esp)

8048f75: e8 31 00 00 00 call 8048fab <strings\_not\_equal>

8048f7a: 85 c0 test %eax,%eax

8048f7c: 74 05 je 8048f83 <phase\_1+0x22>

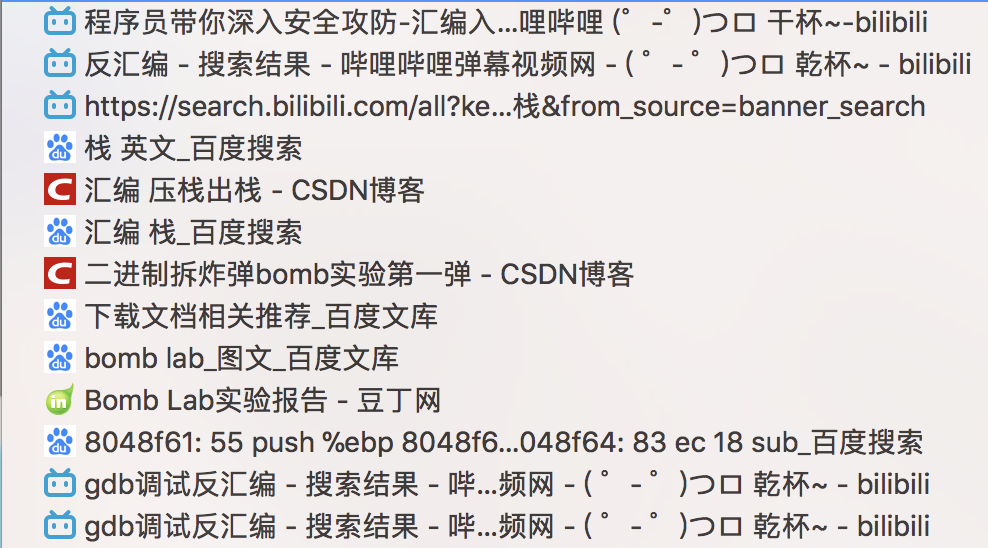
8048f7e: e8 4e 01 00 00 call 80490d1 <explode\_bomb>

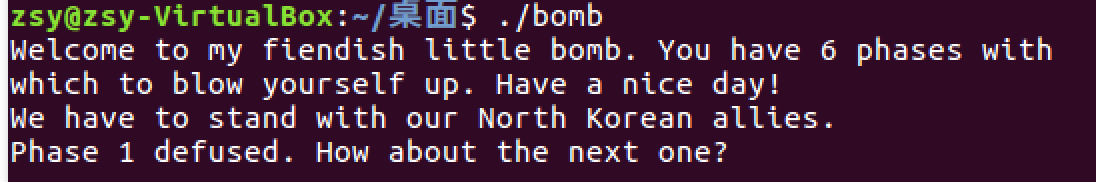
8048f83: c9 leave

8048f84: c3 ret

按照文件中对第一次汇编代码的分析来设置断点，查看密码。暂时不能自主解决，但是可以看懂。



在这之后我去网络上搜索了一些关于汇编的博客与慕课，在吃饭的时候看了一下



2018年4月 13日

第六周周五中午：开始研究第二题。

首先找到phase2代码：

08048d6a <phase\_2>:

8048d6a: 55 push %ebp

8048d6b: 89 e5 mov %esp,%ebp

8048d6d: 56 push %esi

8048d6e: 53 push %ebx

8048d6f: 83 ec 30 sub $0x30,%esp

8048d72: 8d 45 e0 lea -0x20(%ebp),%eax

8048d75: 89 44 24 04 mov %eax,0x4(%esp)

8048d79: 8b 45 08 mov 0x8(%ebp),%eax

8048d7c: 89 04 24 mov %eax,(%esp)

8048d7f: e8 87 03 00 00 call 804910b <read\_six\_numbers>

8048d84: 83 7d e0 00 cmpl $0x0,-0x20(%ebp) 0

8048d88: 75 06 jne 8048d90 <phase\_2+0x26>

8048d8a: 83 7d e4 01 cmpl $0x1,-0x1c(%ebp) 1

8048d8e: 74 05 je 8048d95 <phase\_2+0x2b>

8048d90: e8 3c 03 00 00 call 80490d1 <explode\_bomb>

8048d95: 8d 5d e8 lea -0x18(%ebp),%ebx ebx在最后

8048d98: 8d 75 f8 lea -0x8(%ebp),%esi

8048d9b: 8b 43 fc mov -0x4(%ebx),%eax 1 ebx前一位

8048d9e: 03 43 f8 add -0x8(%ebx),%eax 0 ebx前两位

8048da1: 39 03 cmp %eax,(%ebx)

8048da3: 74 05 je 8048daa <phase\_2+0x40>

8048da5: e8 27 03 00 00 call 80490d1 <explode\_bomb>

8048daa: 83 c3 04 add $0x4,%ebx ebx往下移动一位

8048dad: 39 f3 cmp %esi,%ebx

8048daf: 75 ea jne 8048d9b <phase\_2+0x31>

8048db1: 83 c4 30 add $0x30,%esp

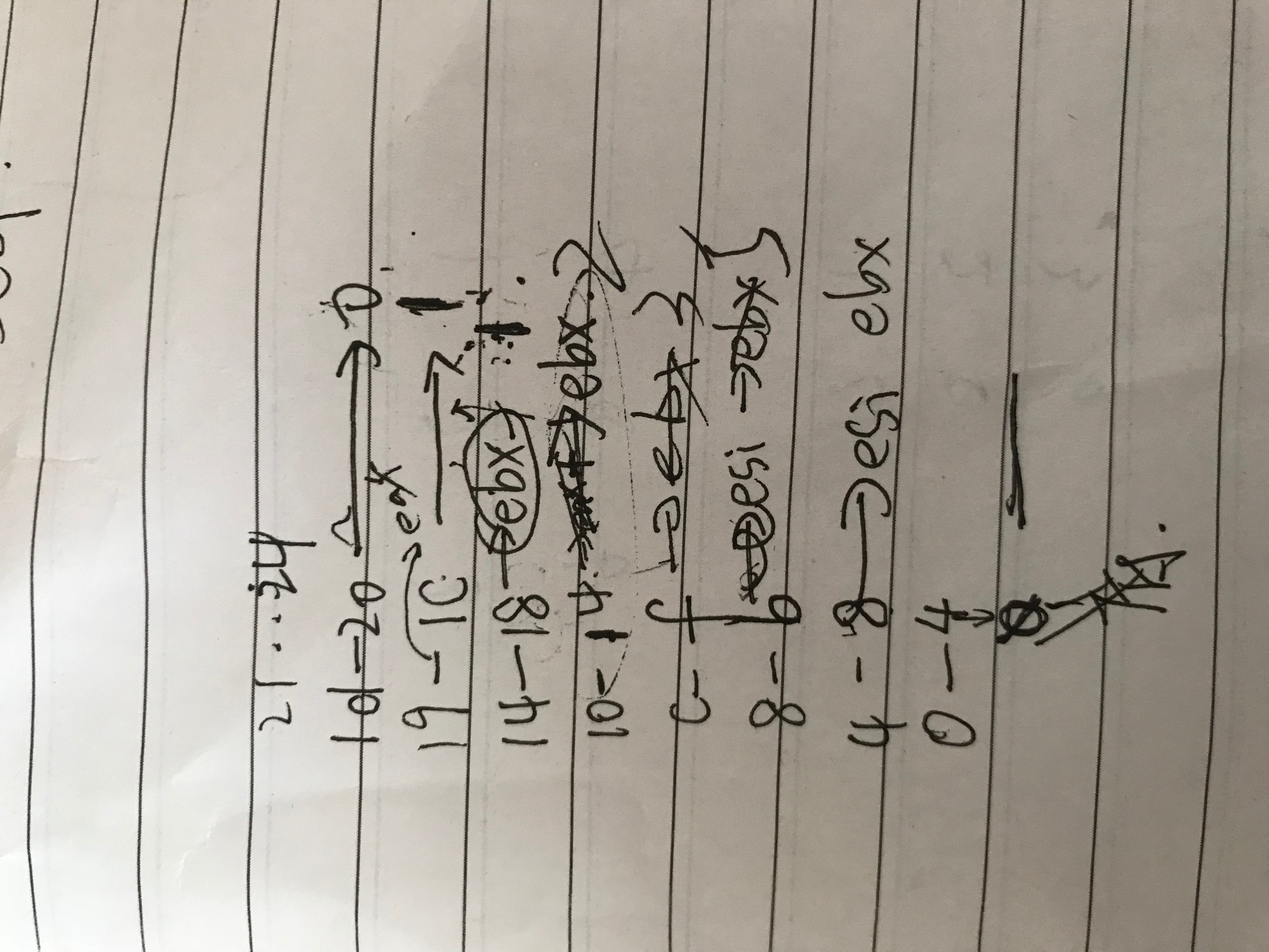
8048db4: 5b pop %ebx

8048db5: 5e pop %esi

8048db6: 5d pop %ebp

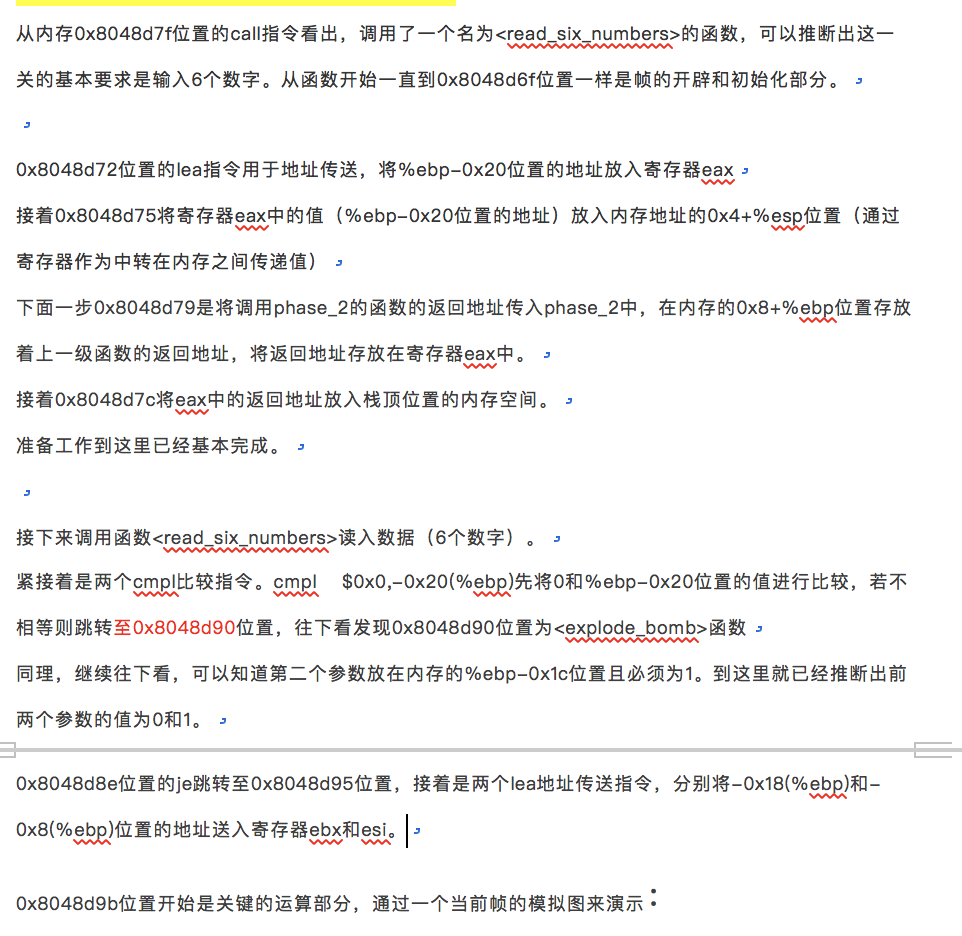
8048db7: c3 ret

为了看得明白，我画了一张图。



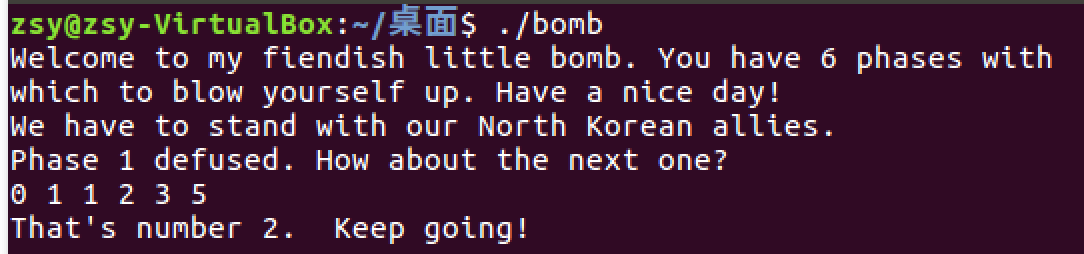


并且在代码中用不同颜色做标注。



在我的反汇编代码下面写下思路

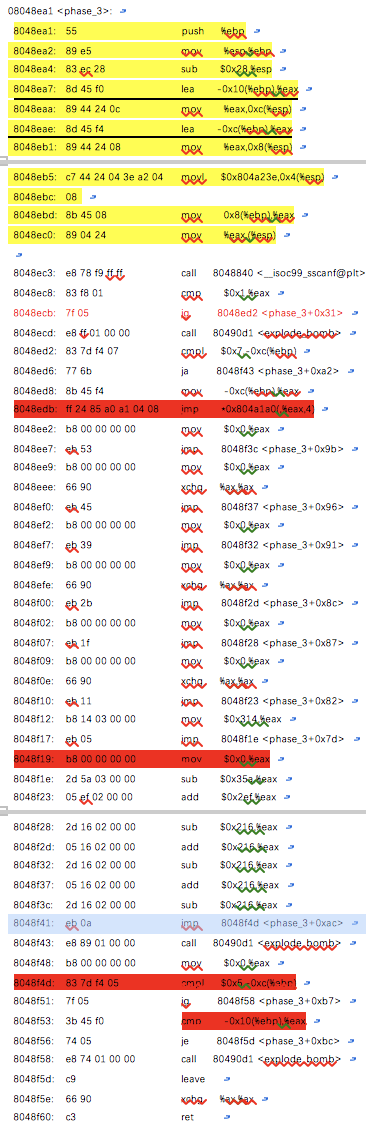
最后解决得出0 1 1 2 3 5



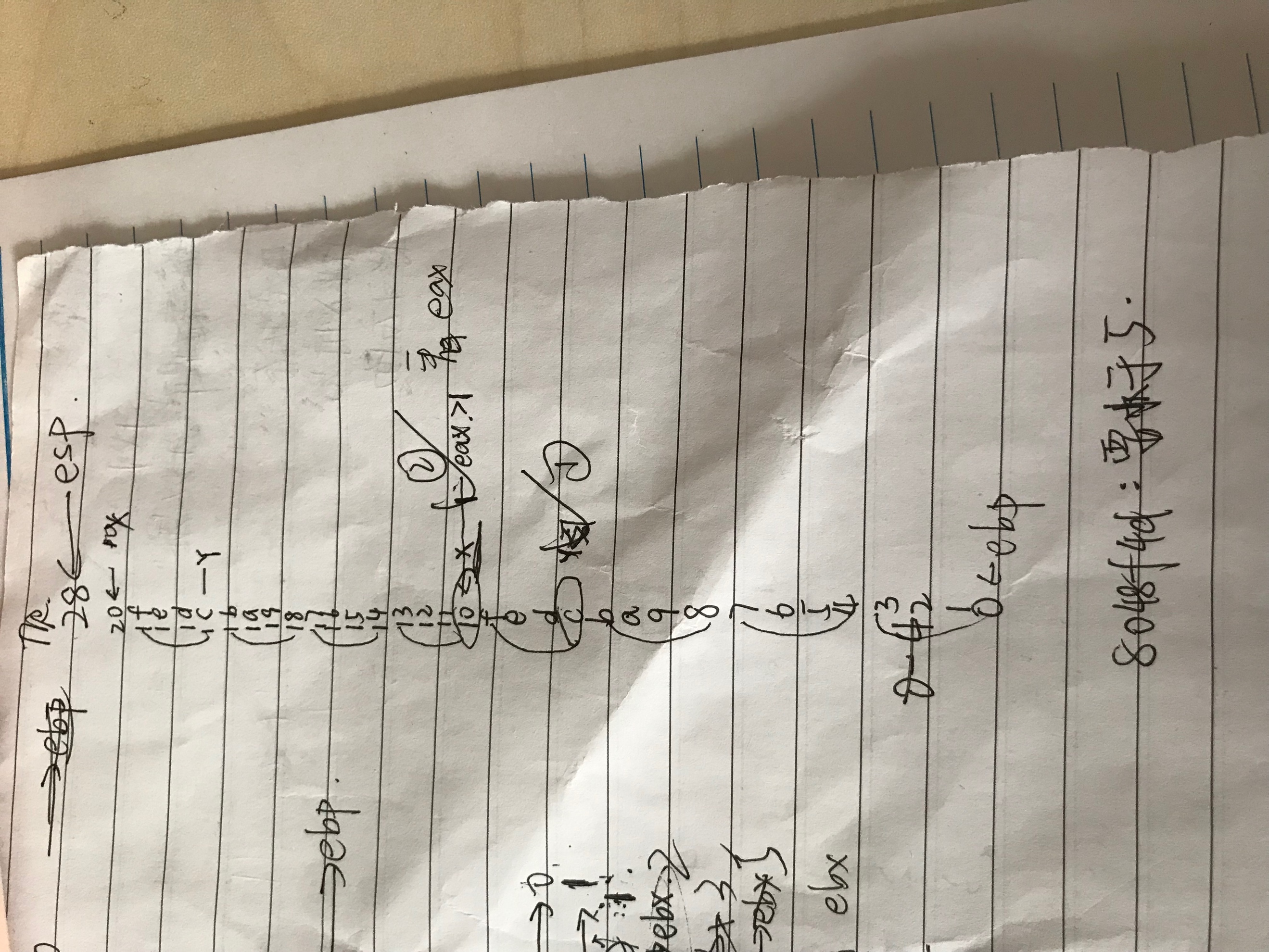
2018年4 月 15日

第六周周日下午： 开始解决第三题

找出对应的代码：



简易画个图图～

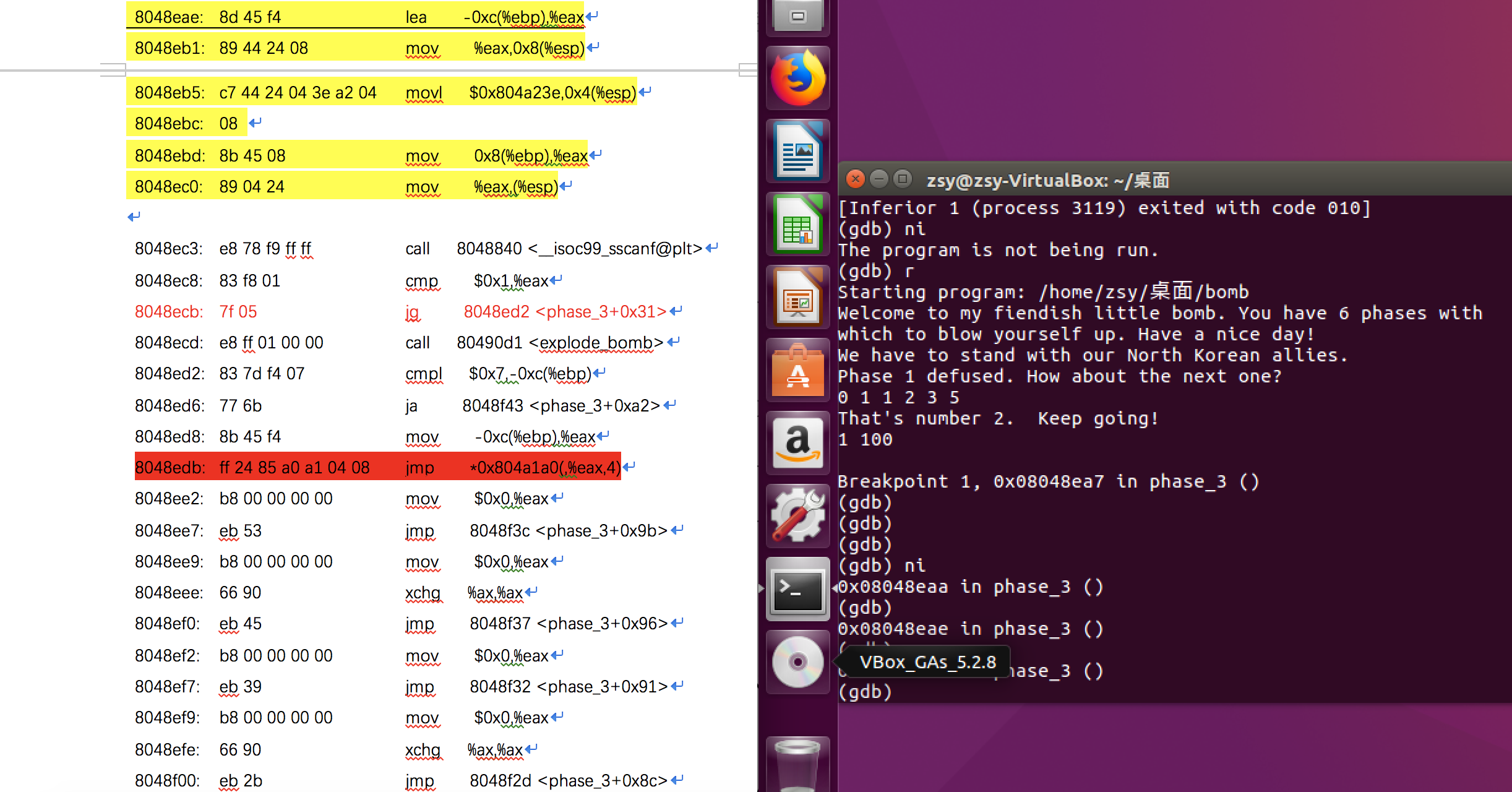


我认为这道题的关键是能分辨出有两个数字是我要输入的。

后面的一大串运算公式能把他们打包在一起，不能让它们扰乱思维，产生抵触情绪（比如昨天的我）

设置断点单步运行，对照着代码来看地址运行到哪一步了

最最最重要的是要格外注意跳转！！！！！！

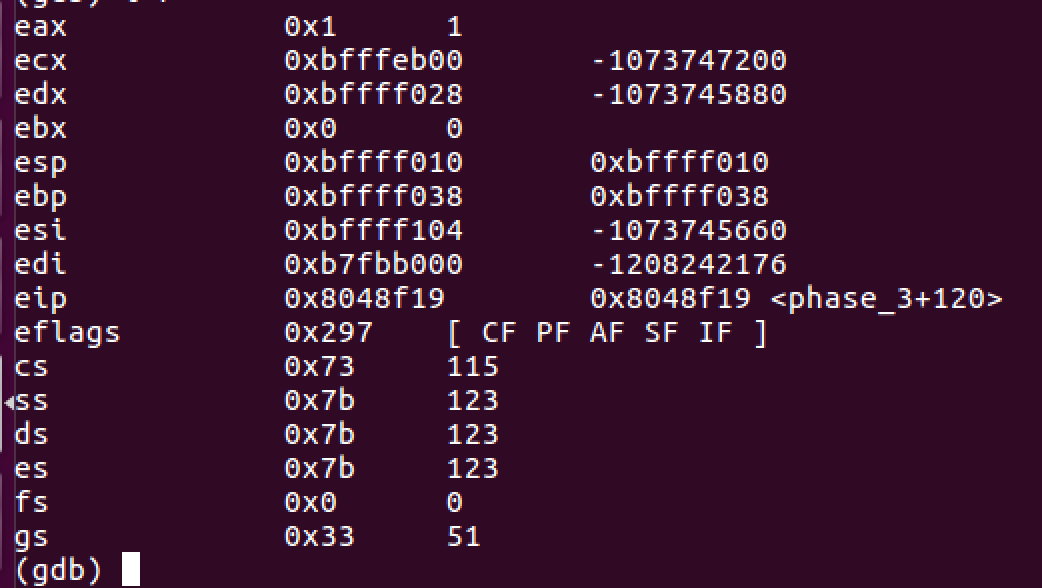


从地址08048eaa开始，两个数我就先设置为1，100，100是我瞎设置的，而1是我根据地址【8048ed2】的值第一个参数需要<7，于是我选择了1.

继续往下看



到edb的时候停止，因为有jmp了。看一下寄存器：



eax已经把我的第一个参数1存好啦

继续往下，到地址【8048f53】停下，查看寄存器

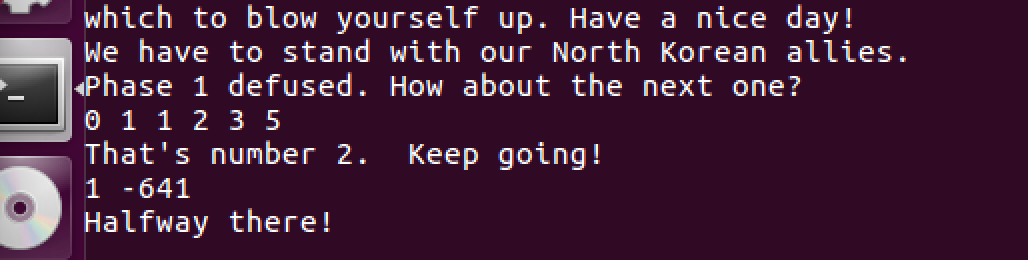
因为那个地址进行的是比较，ebp-10是不是等于eax里的值？？

Ebp-10是我第二个参数，我之前输入的是100，而看下图的eax：



是-641

所以下次我要是输入1 -641的话，就不会爆炸啦



第一个数可以取 0 1 2 3 4 5，分别有对应的第二个数