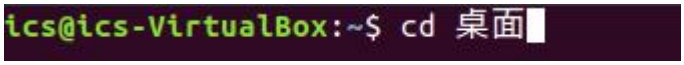
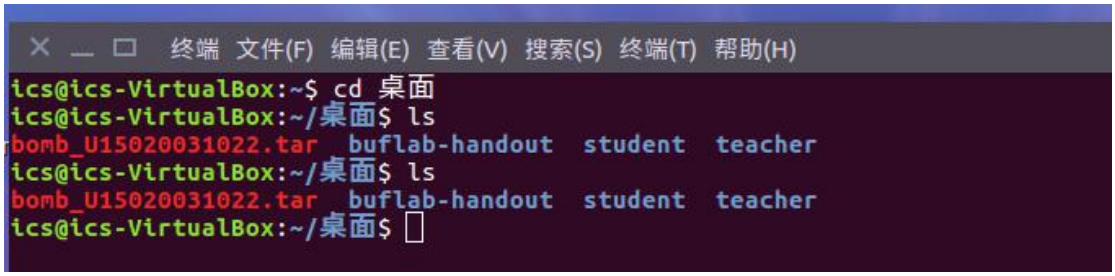
1 首先将下载的文件放到桌面，打开终端进入桌面
命令：cd 桌面

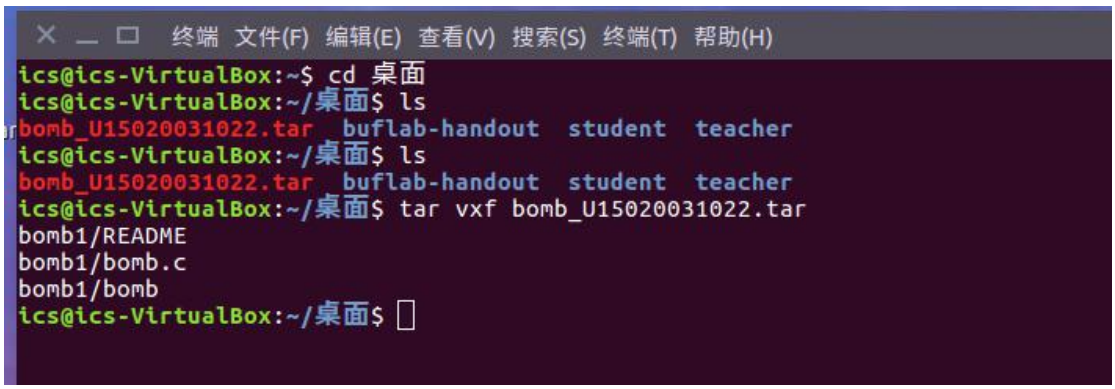


2.看一下桌面有什么
命令 ls

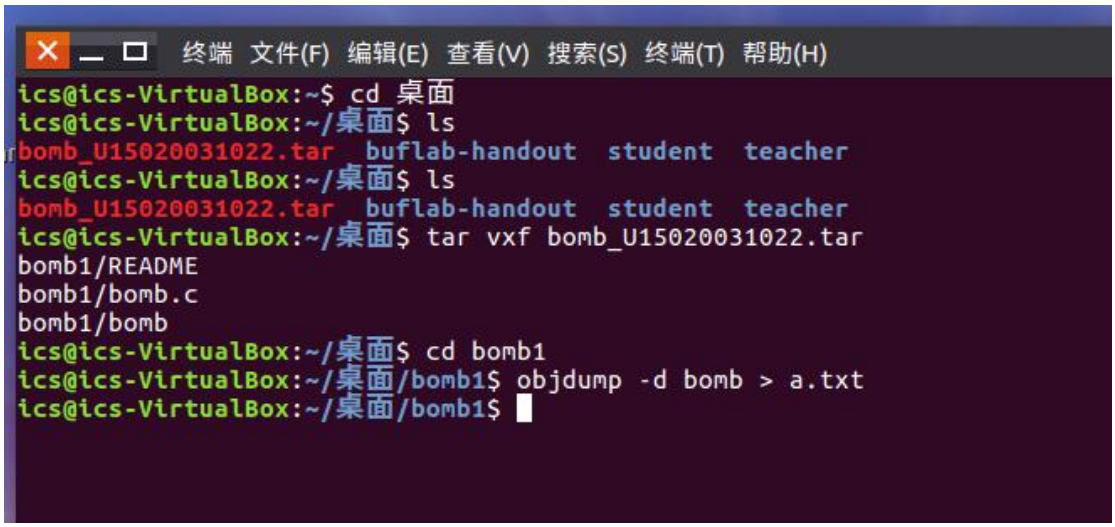


3 看见了放到桌面上的压缩文件 

对压缩文件解压
命令：tar vxf 压缩包名.rar



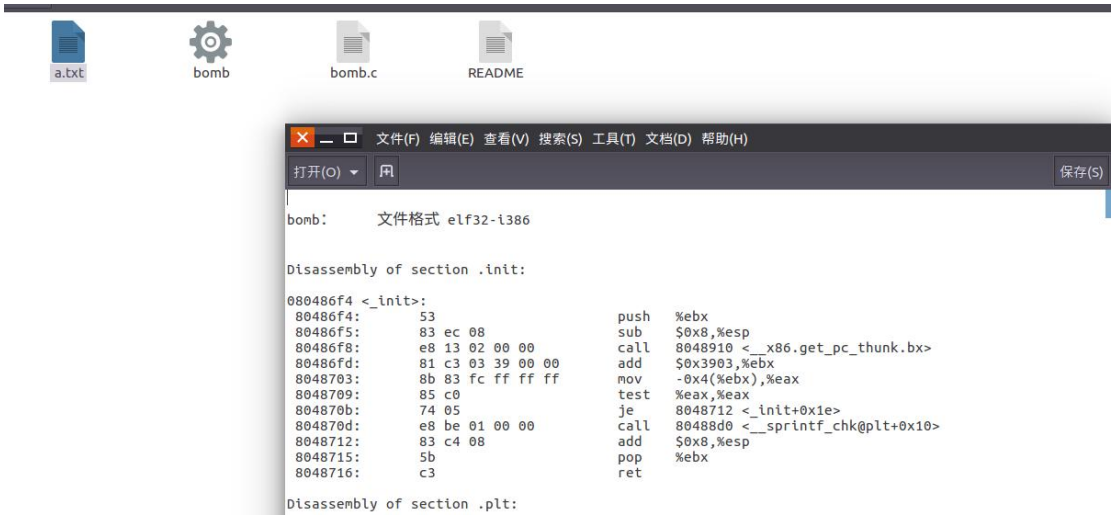
解压后发现 bomb1 中有 3 个文件，桌面上有了 bomb1 文件夹
4.进入 bomb1，输入 objdump -d bomb > a.txt 命令，对 bomb 进行反汇编得到反汇编代码并将存入在 a.txt 文件中



下面打开我们的 bomb1 文件夹发现多了一个文件

```
终端 文件(F) 编辑(E) 查看(V) 搜索(S) 终端(T) 帮助(H)
ics@ics-VirtualBox:~$ cd 桌面
ics@ics-VirtualBox:~/桌面$ ls
bomb_U15020031022.tar  buflab-handout  student  teacher
ics@ics-VirtualBox:~/桌面$ ls
bomb_U15020031022.tar  buflab-handout  student  teacher
ics@ics-VirtualBox:~/桌面$ tar vxf bomb_U15020031022.tar
bomb1/README
bomb1/bomb.c
bomb1/bomb
ics@ics-VirtualBox:~/桌面$ cd bomb1
ics@ics-VirtualBox:~/桌面/bomb1$ objdump -d bomb > a.txt
ics@ics-VirtualBox:~/桌面/bomb1$ ls
a.txt  bomb  bomb.c  README
ics@ics-VirtualBox:~/桌面/bomb1$
```

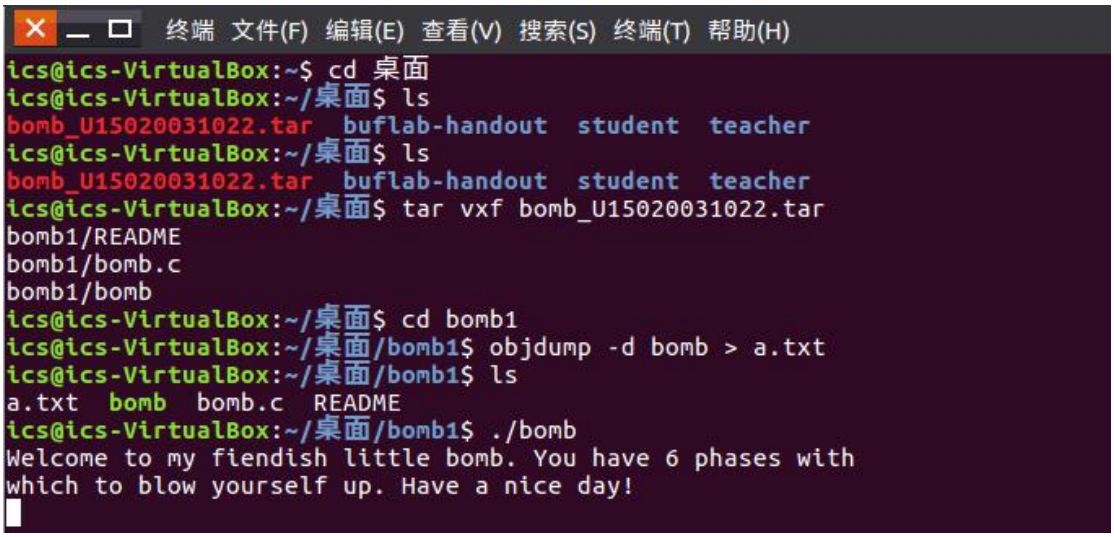
下面打开看一下这个文件进行分析了，可以直接打开也可以用 cat a.txt 命令进行查看。直接打开看一下。



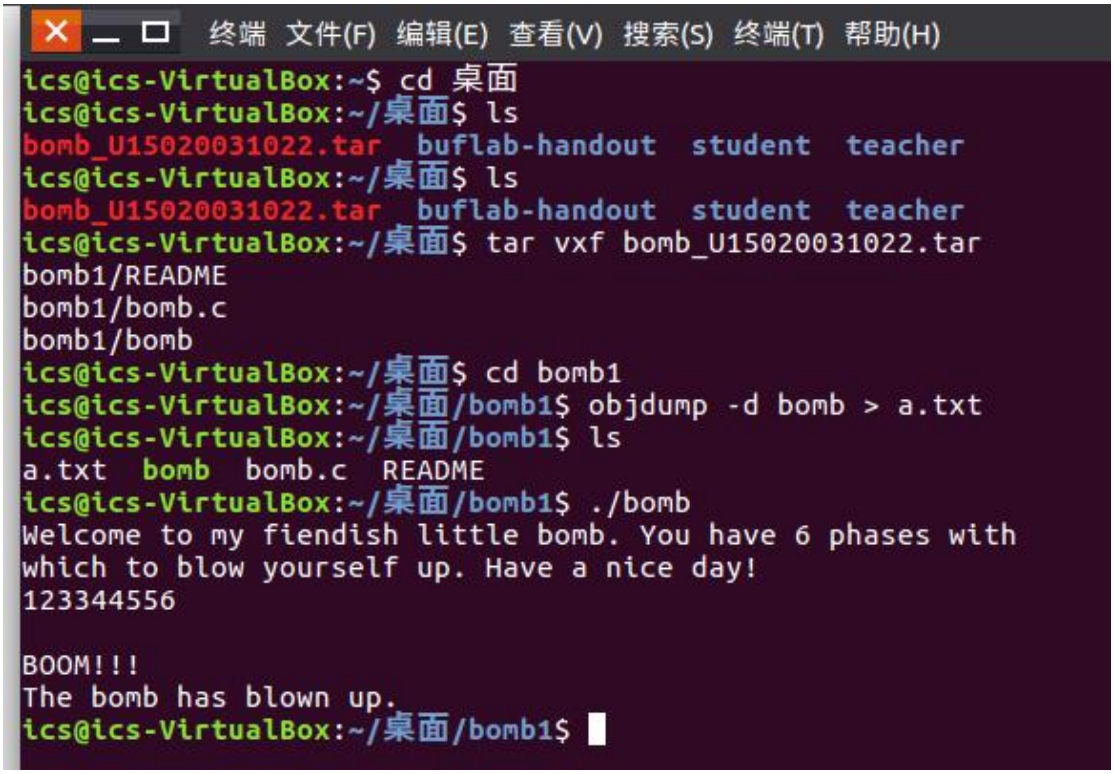
有好多汇编代码，下面可以开始做题了

以第一关为例：

1.先运行一下看一看显示如下：



随便再输入点什么吧



发现 boom!!!，炸弹炸了，我们需要找到一个正确的输入使得拆弹成功。

2.这时候需要分析我们的 a.txt 为文件了

查看汇编源代码 a.txt 文件，在 main 函数中找到如下语句（这里为 phase1 函数在 main()函数中被调用的位置）：

```
 8048a41:       ff 33                   pushl  (%ebx)
 8048a43:       68 9f 9e 04 08          push   $0x8049e9f
 8048a48:       6a 01                   push   $0x1
 8048a4a:       e8 f1 fd ff ff          call   8048840 <__printf_chk@plt>
 8048a4f:       c7 04 24 08 00 00 00    movl   $0x8,(%esp)
 8048a56:       e8 85 fd ff ff          call   80487e0 <exit@plt>
 8048a5b:       e8 c5 05 00 00          call   8049025 <initialize_bomb>
 8048a60:       83 ec 0c                sub    $0xc,%esp
 8048a63:       68 04 9f 04 08          push   $0x8049f04
 8048a68:       e8 53 fd ff ff          call   80487c0 <puts@plt>
 8048a6d:       c7 04 24 40 9f 04 08    movl   $0x8049f40,(%esp)
 8048a74:       e8 47 fd ff ff          call   80487c0 <puts@plt>
 8048a79:       e8 99 06 00 00          call   8049117 <read_line>
 8048a7e:       89 04 24                mov    %eax,(%esp)
 8048a81:       e8 ad 00 00 00          call   8048b33 <phase_1>
 8048a86:       e8 85 07 00 00          call   8049210 <phase_defused>
 8048a8b:       c7 04 24 6c 9f 04 08    movl   $0x8049f6c,(%esp)
```

在反汇编文件中继续查找 phase_1 的位置，如：

```
08048b33 <phase_1>:
 8048b33:       83 ec 14                sub    $0x14,%esp
 8048b36:       68 bc 9f 04 08          push   $0x8049fbc
 8048b3b:       ff 74 24 1c             pushl  0x1c(%esp)
 8048b3f:       e8 7c 04 00 00          call   8048fc0 <strings_not_equal>
 8048b44:       83 c4 10                add    $0x10,%esp
 8048b47:       85 c0                   test   %eax,%eax
 8048b49:       74 05                   je     8048b50 <phase_1+0x1d>
 8048b4b:       e8 67 05 00 00          call   80490b7 <explode_bomb>
 8048b50:       83 c4 0c                add    $0xc,%esp
 8048b53:       c3                      ret
```

分析一下：

push　$0x8049fbc 参数

pushl　0x1c(%esp)　参数

call　8048fc0 <strings_not_equal>调用函数比较字符串是否相等

add　$0x10,%esp

test　%eax,%eax 比较是否相同

je　8048b50 <phase_1+0x1d>相等则会跳到此处

call　80490b7 <explode_bomb>不相等则爆炸

add　$0xc,%esp

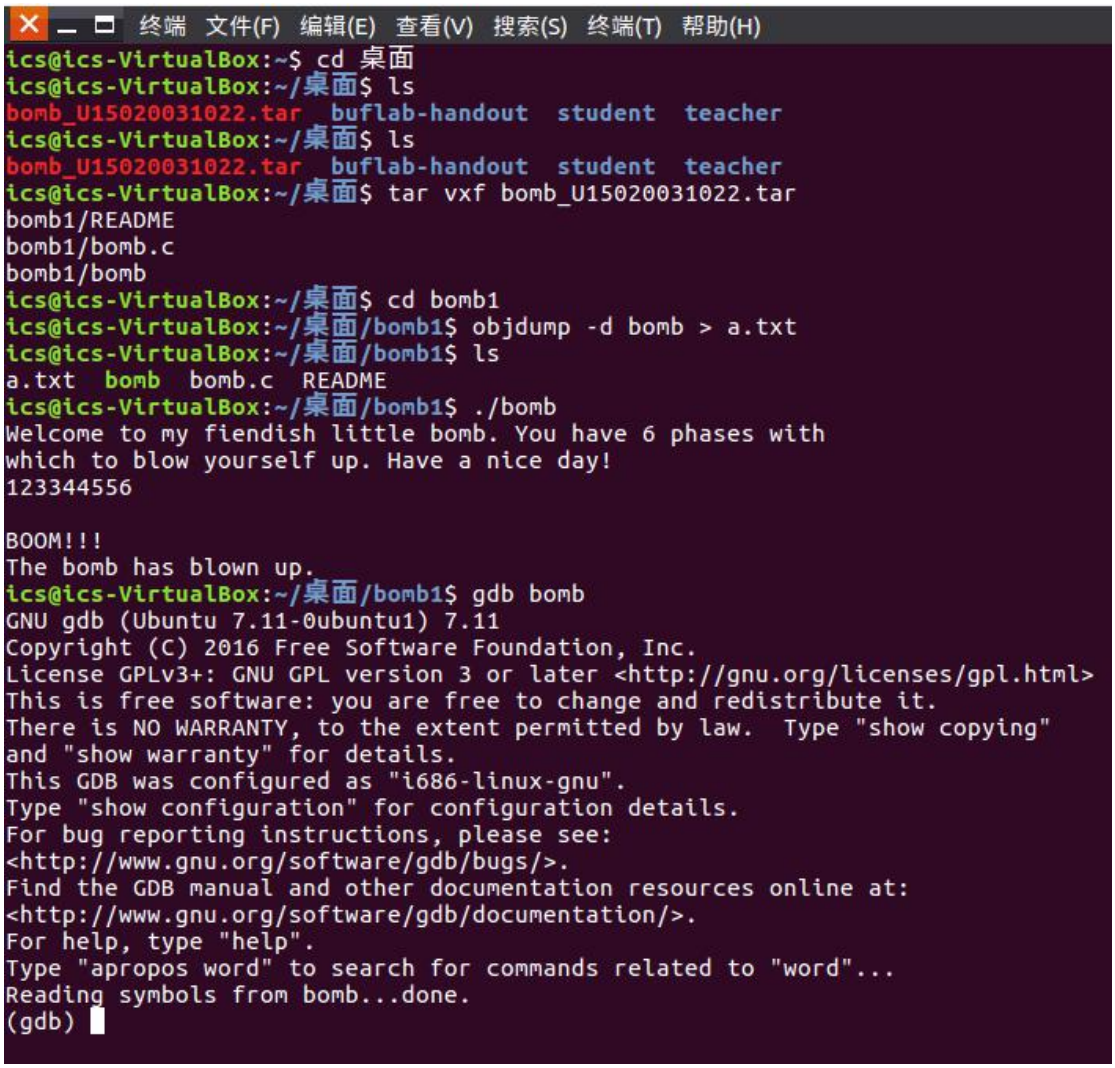
ret

<strings_not_equal>函数两个参数存在于%esp 所指向的堆栈存储单元里。

从上述汇编代码可以看出，在调用 strings_not_equal 函数之前，栈给它准备了两个参数，一个是从 0x8049fbc 里拿的，一个是从 0x1C(%esp)中拿的，由此可知，第一个是原先在内存中已经有的用来比较的，另一个是函数准备的参数，就是自己输入的值。

下面查看一下里面有什么

3.输入 gdb bomb 命令进入调试



设置断点

命令：b phase_1(在 phase_1 函数设断点)

```
终端 文件(F) 编辑(E) 查看(V) 搜索(S) 终端(T) 帮助(H)
ics@ics-VirtualBox:~$ cd 桌面
ics@ics-VirtualBox:~/桌面$ ls
bomb_U15020031022.tar  buflab-handout  student  teacher
ics@ics-VirtualBox:~/桌面$ ls
bomb_U15020031022.tar  buflab-handout  student  teacher
ics@ics-VirtualBox:~/桌面$ tar vxf bomb_U15020031022.tar
bomb1/README
bomb1/bomb.c
bomb1/bomb
ics@ics-VirtualBox:~/桌面$ cd bomb1
ics@ics-VirtualBox:~/桌面/bomb1$ objdump -d bomb > a.txt
ics@ics-VirtualBox:~/桌面/bomb1$ ls
a.txt  bomb  bomb.c  README
ics@ics-VirtualBox:~/桌面/bomb1$ ./bomb
Welcome to my fiendish little bomb. You have 6 phases with
which to blow yourself up. Have a nice day!
123344556

BOOM!!!
The bomb has blown up.
ics@ics-VirtualBox:~/桌面/bomb1$ gdb bomb
GNU gdb (Ubuntu 7.11-0ubuntu1) 7.11
Copyright (C) 2016 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.  Type "show copying"
and "show warranty" for details.
This GDB was configured as "i686-linux-gnu".
Type "show configuration" for configuration details.
For bug reporting instructions, please see:
<http://www.gnu.org/software/gdb/bugs/>.
Find the GDB manual and other documentation resources online at:
<http://www.gnu.org/software/gdb/documentation/>.
For help, type "help".
Type "apropos word" to search for commands related to "word"...
Reading symbols from bomb...done.
(gdb) b phase_1
Breakpoint 1 at 0x8048b33
(gdb)
```

输入 r 运行程序到断点处停止（若没有断点，就一直执行下去直至结束。）

```
(gdb) b phase_1
Breakpoint 1 at 0x8048b33
(gdb) r
Starting program: /home/ics/桌面/bomb1/bomb
Welcome to my fiendish little bomb. You have 6 phases with
which to blow yourself up. Have a nice day!
```

单步执行 ni 命令就是机器指令

执行完 0x08048b3b 查看一下

```
08048b33 <phase_1>:
 8048b33:       83 ec 14                sub    $0x14,%esp
 8048b36:       68 bc 9f 04 08          push   $0x8049fbc
 8048b3b:       ff 74 24 1c             pushl  0x1c(%esp)
```

单步执行汇编语句到 0x08048b3b 如下所示（只有 ni 了之后才会执行）：

```
(gdb) r
Starting program: /home/ics/桌面/bomb1/bomb
Welcome to my fiendish little bomb. You have
which to blow yourself up. Have a nice day!
ni

Breakpoint 1, 0x08048b33 in phase_1 ()
(gdb) ni
0x08048b36 in phase_1 ()
(gdb) ni
0x08048b3b in phase_1 ()
(gdb) ni
0x08048b3f in phase_1 ()
(gdb)
```

查看一下里面有什什么吧！

首先看一下 `push $0x8049fbc` 里面有什么？

命令：x/2s 0x8049fbc（查看地址 0x8049fbc 处两个字符串）

```
(gdb) x/2s 0x8049fbc
0x8049fbc:      "When a problem comes along, you must zip it!"
0x8049fe9:      ""
(gdb)
```

其实这就是答案。检查一下对不对

q 命令退出 gdb 模式

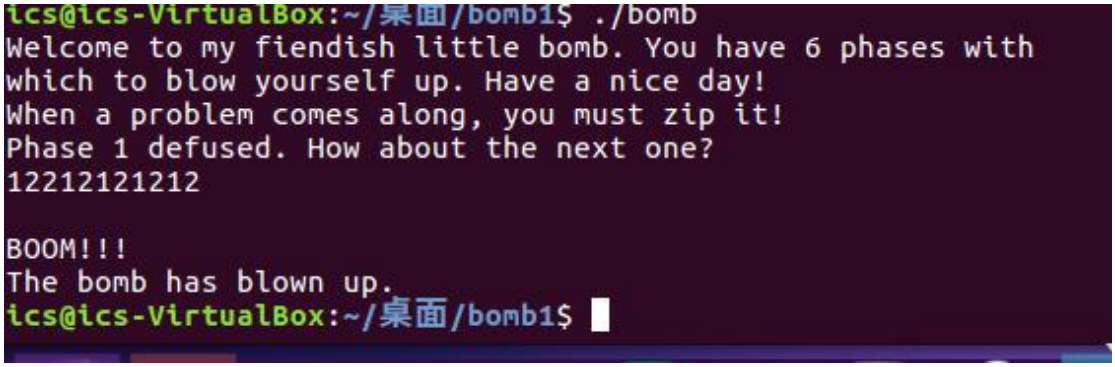
```
(gdb) x/2s 0x8049fbc
0x8049fbc:      "When a problem comes along, you must zip it!"
0x8049fe9:      ""
(gdb) q
A debugging session is active.

        Inferior 1 [process 3711] will be killed.

Quit anyway? (y or n) y
```

检查一下：

```
ics@ics-VirtualBox:~/桌面/bomb1$ ./bomb
Welcome to my fiendish little bomb. You have 6 phases with
which to blow yourself up. Have a nice day!
When a problem comes along, you must zip it!
Phase 1 defused. How about the next one?
```

第一关成功!!

只有过了第一关才能做第二关，继续输入第二关答案即可检测，下面随便输一下



第二关答案错了，大家动手自己做吧，加油！