

- phase_1
 - 字符串相等
- phase_2
 - 循环
 - +delta, delta++
- phase_3
 - a[8] 数组标识目标代码位置
- phase_4
 - $0 \leq in1 \leq 14$
 - $in2 == 3$
 - 枚举
- phase_5
 - 6 长度字符串
 - 每位 % 5 后, 映射到下面数组, 其和为 64
 - 2, 10, 6, 1, 12, 16, 9, 3, 4, 7, 14, 5, 11, 8, 15, 13
- phase_6
 - 6 长度链表
 - 830, 1
 - 699, 2
 - 970, 3
 - 549, 4
 - 937, 5
 - 585, 6
 - 值, 索引
 - 输入 num[1...6]
 - $1 \leq num[i] \leq 6$, 且互不相同
 - $num[i] = 7 - num[i]$, 作为索引的顺序, 对链表进行排序
 - 按照索引排序的结果, 其值应降序排序才不会爆炸
- secret_phase
 - 二叉树
 - 36
 - 8
 - 6
 - 1
 - 7
 - 22
 - 20
 - 39
 - 50

- 45
 - 40
 - 47
- 107
 - 99
 - 1001

```
[G1300012703@ics14 Bomblab]$ gdb bomb
GNU gdb (GDB) Red Hat Enterprise Linux (7.2-56.el6)
Copyright (C) 2010 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 "x86_64-redhat-linux-gnu".
For bug reporting instructions, please see:
<http://www.gnu.org/software/gdb/bugs/>...
Reading symbols from /home/ics2014G/G1300012703/Bomblab/bomb...done.
(gdb) b explode_bomb
Breakpoint 1 at 0x4013fa
(gdb) i b

```

Num	Type	Disp	Enb	Address	What
1	breakpoint	keep	y	0x00000000004013fa	<explode_bomb>

```
(gdb) run ans
Starting program: /home/ics2014G/G1300012703/Bomblab/bomb ans
Welcome to my fiendish little bomb. You have 6 phases with
which to blow yourself up. Have a nice day!
Phase 1 defused. How about the next one?
That's number 2. Keep going!
Halfway there!
So you got that one. Try this one.
Good work! On to the next...
Curses, you've found the secret phase!
But finding it and solving it are quite different...
Wow! You've defused the secret stage!
Congratulations! You've defused the bomb!
Your instructor has been notified and will verify your solution.

Program exited normally.
Missing separate debuginfos, use: debuginfo-install glibc-2.12-1.80.el6.x86_64
(gdb) █
```