tshering@TD:/mnt/d/one drive/OneDrive/Desktop/12190094/Assignment 1/bomb001$ gdb bomb

GNU gdb (Ubuntu 8.1.1-0ubuntu1) 8.1.1

Copyright (C) 2018 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-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\_2 **// set breakpoint for phase 2**

Breakpoint 1 at 0x400ea9

(gdb) b explode\_bomb

Breakpoint 2 at 0x40143d

(gdb) r

Starting program: /mnt/d/one drive/OneDrive/Desktop/12190094/Assignment 1/bomb001/bomb

Welcome to my fiendish little bomb. You have 6 phases with

which to blow yourself up. Have a nice day!

^V

^Z

Program received signal SIGTSTP, Stopped (user).

0x00007fffff110151 in \_\_GI\_\_\_libc\_read (fd=0, buf=0x605270, nbytes=4096) at ../sysdeps/unix/sysv/linux/read.c:27

27 ../sysdeps/unix/sysv/linux/read.c: No such file or directory.

(gdb) run

The program being debugged has been started already.

Start it from the beginning? (y or n) y

Starting program: /mnt/d/one drive/OneDrive/Desktop/12190094/Assignment 1/bomb001/bomb

Welcome to my fiendish little bomb. You have 6 phases with

which to blow yourself up. Have a nice day!

The moon unit will be divided into two divisions.

Phase 1 defused. How about the next one?

Apple **// test input**

Breakpoint 1, 0x0000000000400ea9 in phase\_2 ()

(gdb) diasa

Undefined command: "diasa". Try "help".

(gdb) disas

Dump of assembler code for function phase\_2:

=> 0x0000000000400ea9 <+0>: push %rbp

0x0000000000400eaa <+1>: push %rbx

0x0000000000400eab <+2>: sub $0x28,%rsp

0x0000000000400eaf <+6>: mov %fs:0x28,%rax

0x0000000000400eb8 <+15>: mov %rax,0x18(%rsp)

0x0000000000400ebd <+20>: xor %eax,%eax

0x0000000000400ebf <+22>: mov %rsp,%rsi

0x0000000000400ec2 <+25>: callq 0x40145f <read\_six\_numbers>

0x0000000000400ec7 <+30>: cmpl $0x0,(%rsp) **// from here we know that first integer is 0**

0x0000000000400ecb <+34>: jne 0x400ed4 <phase\_2+43>

0x0000000000400ecd <+36>: cmpl $0x1,0x4(%rsp) **// From here we know that the 2nd integer is 1**

0x0000000000400ed2 <+41>: je 0x400ed9 <phase\_2+48>

0x0000000000400ed4 <+43>: callq 0x40143d <explode\_bomb>

0x0000000000400ed9 <+48>: mov %rsp,%rbx

0x0000000000400edc <+51>: lea 0x10(%rsp),%rbp

0x0000000000400ee1 <+56>: mov 0x4(%rbx),%eax

0x0000000000400ee4 <+59>: add (%rbx),%eax

0x0000000000400ee6 <+61>: cmp %eax,0x8(%rbx)

0x0000000000400ee9 <+64>: je 0x400ef0 <phase\_2+71>

0x0000000000400eeb <+66>: callq 0x40143d <explode\_bomb>

0x0000000000400ef0 <+71>: add $0x4,%rbx

0x0000000000400ef4 <+75>: cmp %rbp,%rbx

0x0000000000400ef7 <+78>: jne 0x400ee1 <phase\_2+56>

0x0000000000400ef9 <+80>: mov 0x18(%rsp),%rax

0x0000000000400efe <+85>: xor %fs:0x28,%rax

0x0000000000400f07 <+94>: je 0x400f0e <phase\_2+101>

0x0000000000400f09 <+96>: callq 0x400b00 <\_\_stack\_chk\_fail@plt>

0x0000000000400f0e <+101>: add $0x28,%rsp

---Type <return> to continue, or q <return> to quit---

0x0000000000400f12 <+105>: pop %rbx

0x0000000000400f13 <+106>: pop %rbp

0x0000000000400f14 <+107>: retq

End of assembler dump.

(gdb) until \*0x0000000000400ec2

0x0000000000400ec2 in phase\_2 ()

(gdb) disas

Dump of assembler code for function phase\_2:

0x0000000000400ea9 <+0>: push %rbp

0x0000000000400eaa <+1>: push %rbx

0x0000000000400eab <+2>: sub $0x28,%rsp

0x0000000000400eaf <+6>: mov %fs:0x28,%rax

0x0000000000400eb8 <+15>: mov %rax,0x18(%rsp)

0x0000000000400ebd <+20>: xor %eax,%eax

0x0000000000400ebf <+22>: mov %rsp,%rsi

=> 0x0000000000400ec2 <+25>: callq 0x40145f <read\_six\_numbers>

0x0000000000400ec7 <+30>: cmpl $0x0,(%rsp)

0x0000000000400ecb <+34>: jne 0x400ed4 <phase\_2+43>

0x0000000000400ecd <+36>: cmpl $0x1,0x4(%rsp)

0x0000000000400ed2 <+41>: je 0x400ed9 <phase\_2+48>

0x0000000000400ed4 <+43>: callq 0x40143d <explode\_bomb>

0x0000000000400ed9 <+48>: mov %rsp,%rbx

0x0000000000400edc <+51>: lea 0x10(%rsp),%rbp

0x0000000000400ee1 <+56>: mov 0x4(%rbx),%eax

0x0000000000400ee4 <+59>: add (%rbx),%eax

0x0000000000400ee6 <+61>: cmp %eax,0x8(%rbx)

0x0000000000400ee9 <+64>: je 0x400ef0 <phase\_2+71>

0x0000000000400eeb <+66>: callq 0x40143d <explode\_bomb>

0x0000000000400ef0 <+71>: add $0x4,%rbx

0x0000000000400ef4 <+75>: cmp %rbp,%rbx

0x0000000000400ef7 <+78>: jne 0x400ee1 <phase\_2+56>

0x0000000000400ef9 <+80>: mov 0x18(%rsp),%rax

0x0000000000400efe <+85>: xor %fs:0x28,%rax

0x0000000000400f07 <+94>: je 0x400f0e <phase\_2+101>

0x0000000000400f09 <+96>: callq 0x400b00 <\_\_stack\_chk\_fail@plt>

0x0000000000400f0e <+101>: add $0x28,%rsp

---Type <return> to continue, or q <return> to quit---

0x0000000000400f12 <+105>: pop %rbx

0x0000000000400f13 <+106>: pop %rbp

0x0000000000400f14 <+107>: retq

End of assembler dump.

(gdb) si

0x000000000040145f in read\_six\_numbers ()

(gdb) disas

Dump of assembler code for function read\_six\_numbers:

=> 0x000000000040145f <+0>: sub $0x8,%rsp

0x0000000000401463 <+4>: mov %rsi,%rdx

0x0000000000401466 <+7>: lea 0x4(%rsi),%rcx

0x000000000040146a <+11>: lea 0x14(%rsi),%rax

0x000000000040146e <+15>: push %rax

0x000000000040146f <+16>: lea 0x10(%rsi),%rax

0x0000000000401473 <+20>: push %rax

0x0000000000401474 <+21>: lea 0xc(%rsi),%r9

0x0000000000401478 <+25>: lea 0x8(%rsi),%r8

0x000000000040147c <+29>: mov $0x4025c3,%esi

0x0000000000401481 <+34>: mov $0x0,%eax

0x0000000000401486 <+39>: callq 0x400bb0 <\_\_isoc99\_sscanf@plt>

0x000000000040148b <+44>: add $0x10,%rsp

0x000000000040148f <+48>: cmp $0x5,%eax

0x0000000000401492 <+51>: jg 0x401499 <read\_six\_numbers+58>

0x0000000000401494 <+53>: callq 0x40143d <explode\_bomb>

0x0000000000401499 <+58>: add $0x8,%rsp

0x000000000040149d <+62>: retq

End of assembler dump.

(gdb) x/s 0x4025c3

0x4025c3: "%d %d %d %d %d %d" **// it is the input format that is 6 integer**

(gdb) r

The program being debugged has been started already.

Start it from the beginning? (y or n) y

Starting program: /mnt/d/one drive/OneDrive/Desktop/12190094/Assignment 1/bomb001/bomb

Welcome to my fiendish little bomb. You have 6 phases with

which to blow yourself up. Have a nice day!

The moon unit will be divided into two divisions.

Phase 1 defused. How about the next one?

0 1 2 3 4 5 **// test input using integer**

Breakpoint 1, 0x0000000000400ea9 in phase\_2 ()

(gdb) disas

Dump of assembler code for function phase\_2:

=> 0x0000000000400ea9 <+0>: push %rbp

0x0000000000400eaa <+1>: push %rbx

0x0000000000400eab <+2>: sub $0x28,%rsp

0x0000000000400eaf <+6>: mov %fs:0x28,%rax

0x0000000000400eb8 <+15>: mov %rax,0x18(%rsp)

0x0000000000400ebd <+20>: xor %eax,%eax

0x0000000000400ebf <+22>: mov %rsp,%rsi

0x0000000000400ec2 <+25>: callq 0x40145f <read\_six\_numbers>

0x0000000000400ec7 <+30>: cmpl $0x0,(%rsp)

0x0000000000400ecb <+34>: jne 0x400ed4 <phase\_2+43>

0x0000000000400ecd <+36>: cmpl $0x1,0x4(%rsp)

0x0000000000400ed2 <+41>: je 0x400ed9 <phase\_2+48>

0x0000000000400ed4 <+43>: callq 0x40143d <explode\_bomb>

0x0000000000400ed9 <+48>: mov %rsp,%rbx

0x0000000000400edc <+51>: lea 0x10(%rsp),%rbp

0x0000000000400ee1 <+56>: mov 0x4(%rbx),%eax

0x0000000000400ee4 <+59>: add (%rbx),%eax

0x0000000000400ee6 <+61>: cmp %eax,0x8(%rbx)

0x0000000000400ee9 <+64>: je 0x400ef0 <phase\_2+71>

0x0000000000400eeb <+66>: callq 0x40143d <explode\_bomb>

0x0000000000400ef0 <+71>: add $0x4,%rbx

0x0000000000400ef4 <+75>: cmp %rbp,%rbx

0x0000000000400ef7 <+78>: jne 0x400ee1 <phase\_2+56>

0x0000000000400ef9 <+80>: mov 0x18(%rsp),%rax

0x0000000000400efe <+85>: xor %fs:0x28,%rax

0x0000000000400f07 <+94>: je 0x400f0e <phase\_2+101>

0x0000000000400f09 <+96>: callq 0x400b00 <\_\_stack\_chk\_fail@plt>

0x0000000000400f0e <+101>: add $0x28,%rsp

---Type <return> to continue, or q <return> to quit---

0x0000000000400f12 <+105>: pop %rbx

0x0000000000400f13 <+106>: pop %rbp

0x0000000000400f14 <+107>: retq

End of assembler dump.

(gdb) until \* 0x0000000000400ee6 **// To get into the address**

0x0000000000400ee6 in phase\_2 ()

(gdb) i r

rax 0x1 1 **//The third input is 1 because the input stored in rax 64 bits is 1 as eax 32 bit stores the input**

rbx 0x7ffffffee300 140737488282368

rcx 0x0 0

rdx 0x7ffffffee314 140737488282388

rsi 0x0 0

rdi 0x7ffffffedc70 140737488280688

rbp 0x7ffffffee310 0x7ffffffee310

rsp 0x7ffffffee300 0x7ffffffee300

r8 0x0 0

r9 0x0 0

r10 0x7fffff19ec40 140737473276992

r11 0x4025d4 4203988

r12 0x400c60 4197472

r13 0x7ffffffee420 140737488282656

r14 0x0 0

r15 0x0 0

rip 0x400ee6 0x400ee6 <phase\_2+61>

eflags 0x202 [ IF ]

cs 0x33 51

ss 0x2b 43

ds 0x0 0

es 0x0 0

fs 0x0 0

gs 0x0 0

(gdb) r

The program being debugged has been started already.

Start it from the beginning? (y or n) y

Starting program: /mnt/d/one drive/OneDrive/Desktop/12190094/Assignment 1/bomb001/bomb

Welcome to my fiendish little bomb. You have 6 phases with

which to blow yourself up. Have a nice day!

The moon unit will be divided into two divisions.

Phase 1 defused. How about the next one?

0 1 1 3 4 5 **// test input for fourth integer**

Breakpoint 1, 0x0000000000400ea9 in phase\_2 ()

(gdb) disas

Dump of assembler code for function phase\_2:

=> 0x0000000000400ea9 <+0>: push %rbp

0x0000000000400eaa <+1>: push %rbx

0x0000000000400eab <+2>: sub $0x28,%rsp

0x0000000000400eaf <+6>: mov %fs:0x28,%rax

0x0000000000400eb8 <+15>: mov %rax,0x18(%rsp)

0x0000000000400ebd <+20>: xor %eax,%eax

0x0000000000400ebf <+22>: mov %rsp,%rsi

0x0000000000400ec2 <+25>: callq 0x40145f <read\_six\_numbers>

0x0000000000400ec7 <+30>: cmpl $0x0,(%rsp)

0x0000000000400ecb <+34>: jne 0x400ed4 <phase\_2+43>

0x0000000000400ecd <+36>: cmpl $0x1,0x4(%rsp)

0x0000000000400ed2 <+41>: je 0x400ed9 <phase\_2+48>

0x0000000000400ed4 <+43>: callq 0x40143d <explode\_bomb>

0x0000000000400ed9 <+48>: mov %rsp,%rbx

0x0000000000400edc <+51>: lea 0x10(%rsp),%rbp

0x0000000000400ee1 <+56>: mov 0x4(%rbx),%eax

0x0000000000400ee4 <+59>: add (%rbx),%eax

0x0000000000400ee6 <+61>: cmp %eax,0x8(%rbx)

0x0000000000400ee9 <+64>: je 0x400ef0 <phase\_2+71>

0x0000000000400eeb <+66>: callq 0x40143d <explode\_bomb>

0x0000000000400ef0 <+71>: add $0x4,%rbx

0x0000000000400ef4 <+75>: cmp %rbp,%rbx

0x0000000000400ef7 <+78>: jne 0x400ee1 <phase\_2+56>

0x0000000000400ef9 <+80>: mov 0x18(%rsp),%rax

0x0000000000400efe <+85>: xor %fs:0x28,%rax

0x0000000000400f07 <+94>: je 0x400f0e <phase\_2+101>

0x0000000000400f09 <+96>: callq 0x400b00 <\_\_stack\_chk\_fail@plt>

0x0000000000400f0e <+101>: add $0x28,%rsp

---Type <return> to continue, or q <return> to quit---

0x0000000000400f12 <+105>: pop %rbx

0x0000000000400f13 <+106>: pop %rbp

0x0000000000400f14 <+107>: retq

End of assembler dump.

(gdb) until \*0x0000000000400ee6

0x0000000000400ee6 in phase\_2 ()

(gdb) disas

Dump of assembler code for function phase\_2:

0x0000000000400ea9 <+0>: push %rbp

0x0000000000400eaa <+1>: push %rbx

0x0000000000400eab <+2>: sub $0x28,%rsp

0x0000000000400eaf <+6>: mov %fs:0x28,%rax

0x0000000000400eb8 <+15>: mov %rax,0x18(%rsp)

0x0000000000400ebd <+20>: xor %eax,%eax

0x0000000000400ebf <+22>: mov %rsp,%rsi

0x0000000000400ec2 <+25>: callq 0x40145f <read\_six\_numbers>

0x0000000000400ec7 <+30>: cmpl $0x0,(%rsp)

0x0000000000400ecb <+34>: jne 0x400ed4 <phase\_2+43>

0x0000000000400ecd <+36>: cmpl $0x1,0x4(%rsp)

0x0000000000400ed2 <+41>: je 0x400ed9 <phase\_2+48>

0x0000000000400ed4 <+43>: callq 0x40143d <explode\_bomb>

0x0000000000400ed9 <+48>: mov %rsp,%rbx

0x0000000000400edc <+51>: lea 0x10(%rsp),%rbp

0x0000000000400ee1 <+56>: mov 0x4(%rbx),%eax

0x0000000000400ee4 <+59>: add (%rbx),%eax

=> 0x0000000000400ee6 <+61>: cmp %eax,0x8(%rbx)

0x0000000000400ee9 <+64>: je 0x400ef0 <phase\_2+71>

0x0000000000400eeb <+66>: callq 0x40143d <explode\_bomb>

0x0000000000400ef0 <+71>: add $0x4,%rbx

0x0000000000400ef4 <+75>: cmp %rbp,%rbx

0x0000000000400ef7 <+78>: jne 0x400ee1 <phase\_2+56>

0x0000000000400ef9 <+80>: mov 0x18(%rsp),%rax

0x0000000000400efe <+85>: xor %fs:0x28,%rax

0x0000000000400f07 <+94>: je 0x400f0e <phase\_2+101>

0x0000000000400f09 <+96>: callq 0x400b00 <\_\_stack\_chk\_fail@plt>

0x0000000000400f0e <+101>: add $0x28,%rsp

---Type <return> to continue, or q <return> to quit---

0x0000000000400f12 <+105>: pop %rbx

0x0000000000400f13 <+106>: pop %rbp

0x0000000000400f14 <+107>: retq

End of assembler dump.

(gdb) ni

0x0000000000400ee9 in phase\_2 ()

(gdb) ni

0x0000000000400ef0 in phase\_2 ()

(gdb) disas

Dump of assembler code for function phase\_2:

0x0000000000400ea9 <+0>: push %rbp

0x0000000000400eaa <+1>: push %rbx

0x0000000000400eab <+2>: sub $0x28,%rsp

0x0000000000400eaf <+6>: mov %fs:0x28,%rax

0x0000000000400eb8 <+15>: mov %rax,0x18(%rsp)

0x0000000000400ebd <+20>: xor %eax,%eax

0x0000000000400ebf <+22>: mov %rsp,%rsi

0x0000000000400ec2 <+25>: callq 0x40145f <read\_six\_numbers>

0x0000000000400ec7 <+30>: cmpl $0x0,(%rsp)

0x0000000000400ecb <+34>: jne 0x400ed4 <phase\_2+43>

0x0000000000400ecd <+36>: cmpl $0x1,0x4(%rsp)

0x0000000000400ed2 <+41>: je 0x400ed9 <phase\_2+48>

0x0000000000400ed4 <+43>: callq 0x40143d <explode\_bomb>

0x0000000000400ed9 <+48>: mov %rsp,%rbx

0x0000000000400edc <+51>: lea 0x10(%rsp),%rbp

0x0000000000400ee1 <+56>: mov 0x4(%rbx),%eax

0x0000000000400ee4 <+59>: add (%rbx),%eax

0x0000000000400ee6 <+61>: cmp %eax,0x8(%rbx)

0x0000000000400ee9 <+64>: je 0x400ef0 <phase\_2+71>

0x0000000000400eeb <+66>: callq 0x40143d <explode\_bomb>

=> 0x0000000000400ef0 <+71>: add $0x4,%rbx

0x0000000000400ef4 <+75>: cmp %rbp,%rbx

0x0000000000400ef7 <+78>: jne 0x400ee1 <phase\_2+56>

0x0000000000400ef9 <+80>: mov 0x18(%rsp),%rax

0x0000000000400efe <+85>: xor %fs:0x28,%rax

0x0000000000400f07 <+94>: je 0x400f0e <phase\_2+101>

0x0000000000400f09 <+96>: callq 0x400b00 <\_\_stack\_chk\_fail@plt>

0x0000000000400f0e <+101>: add $0x28,%rsp

---Type <return> to continue, or q <return> to quit---ni

0x0000000000400f12 <+105>: pop %rbx

0x0000000000400f13 <+106>: pop %rbp

0x0000000000400f14 <+107>: retq

End of assembler dump.

(gdb) ni

0x0000000000400ef4 in phase\_2 ()

(gdb) disas

Dump of assembler code for function phase\_2:

0x0000000000400ea9 <+0>: push %rbp

0x0000000000400eaa <+1>: push %rbx

0x0000000000400eab <+2>: sub $0x28,%rsp

0x0000000000400eaf <+6>: mov %fs:0x28,%rax

0x0000000000400eb8 <+15>: mov %rax,0x18(%rsp)

0x0000000000400ebd <+20>: xor %eax,%eax

0x0000000000400ebf <+22>: mov %rsp,%rsi

0x0000000000400ec2 <+25>: callq 0x40145f <read\_six\_numbers>

0x0000000000400ec7 <+30>: cmpl $0x0,(%rsp)

0x0000000000400ecb <+34>: jne 0x400ed4 <phase\_2+43>

0x0000000000400ecd <+36>: cmpl $0x1,0x4(%rsp)

0x0000000000400ed2 <+41>: je 0x400ed9 <phase\_2+48>

0x0000000000400ed4 <+43>: callq 0x40143d <explode\_bomb>

0x0000000000400ed9 <+48>: mov %rsp,%rbx

0x0000000000400edc <+51>: lea 0x10(%rsp),%rbp

0x0000000000400ee1 <+56>: mov 0x4(%rbx),%eax

0x0000000000400ee4 <+59>: add (%rbx),%eax

0x0000000000400ee6 <+61>: cmp %eax,0x8(%rbx)

0x0000000000400ee9 <+64>: je 0x400ef0 <phase\_2+71>

0x0000000000400eeb <+66>: callq 0x40143d <explode\_bomb>

0x0000000000400ef0 <+71>: add $0x4,%rbx

=> 0x0000000000400ef4 <+75>: cmp %rbp,%rbx

0x0000000000400ef7 <+78>: jne 0x400ee1 <phase\_2+56>

0x0000000000400ef9 <+80>: mov 0x18(%rsp),%rax

0x0000000000400efe <+85>: xor %fs:0x28,%rax

0x0000000000400f07 <+94>: je 0x400f0e <phase\_2+101>

0x0000000000400f09 <+96>: callq 0x400b00 <\_\_stack\_chk\_fail@plt>

0x0000000000400f0e <+101>: add $0x28,%rsp

---Type <return> to continue, or q <return> to quit---

0x0000000000400f12 <+105>: pop %rbx

0x0000000000400f13 <+106>: pop %rbp

0x0000000000400f14 <+107>: retq

End of assembler dump.

(gdb) i r

rax 0x1 1.

rbx 0x7ffffffee304 140737488282372

rcx 0x0 0

rdx 0x7ffffffee314 140737488282388

rsi 0x0 0

rdi 0x7ffffffedc70 140737488280688

rbp 0x7ffffffee310 0x7ffffffee310

rsp 0x7ffffffee300 0x7ffffffee300

r8 0x0 0

r9 0x0 0

r10 0x7fffff19ec40 140737473276992

r11 0x4025d4 4203988

r12 0x400c60 4197472

r13 0x7ffffffee420 140737488282656

r14 0x0 0

r15 0x0 0

rip 0x400ef4 0x400ef4 <phase\_2+75>

eflags 0x202 [ IF ]

cs 0x33 51

ss 0x2b 43

ds 0x0 0

es 0x0 0

fs 0x0 0

gs 0x0 0

(gdb) ni

0x0000000000400ef7 in phase\_2 ()

(gdb) disas

Dump of assembler code for function phase\_2:

0x0000000000400ea9 <+0>: push %rbp

0x0000000000400eaa <+1>: push %rbx

0x0000000000400eab <+2>: sub $0x28,%rsp

0x0000000000400eaf <+6>: mov %fs:0x28,%rax

0x0000000000400eb8 <+15>: mov %rax,0x18(%rsp)

0x0000000000400ebd <+20>: xor %eax,%eax

0x0000000000400ebf <+22>: mov %rsp,%rsi

0x0000000000400ec2 <+25>: callq 0x40145f <read\_six\_numbers>

0x0000000000400ec7 <+30>: cmpl $0x0,(%rsp)

0x0000000000400ecb <+34>: jne 0x400ed4 <phase\_2+43>

0x0000000000400ecd <+36>: cmpl $0x1,0x4(%rsp)

0x0000000000400ed2 <+41>: je 0x400ed9 <phase\_2+48>

0x0000000000400ed4 <+43>: callq 0x40143d <explode\_bomb>

0x0000000000400ed9 <+48>: mov %rsp,%rbx

0x0000000000400edc <+51>: lea 0x10(%rsp),%rbp

0x0000000000400ee1 <+56>: mov 0x4(%rbx),%eax

0x0000000000400ee4 <+59>: add (%rbx),%eax

0x0000000000400ee6 <+61>: cmp %eax,0x8(%rbx)

0x0000000000400ee9 <+64>: je 0x400ef0 <phase\_2+71>

0x0000000000400eeb <+66>: callq 0x40143d <explode\_bomb>

0x0000000000400ef0 <+71>: add $0x4,%rbx

0x0000000000400ef4 <+75>: cmp %rbp,%rbx

=> 0x0000000000400ef7 <+78>: jne 0x400ee1 <phase\_2+56>

0x0000000000400ef9 <+80>: mov 0x18(%rsp),%rax

0x0000000000400efe <+85>: xor %fs:0x28,%rax

0x0000000000400f07 <+94>: je 0x400f0e <phase\_2+101>

0x0000000000400f09 <+96>: callq 0x400b00 <\_\_stack\_chk\_fail@plt>

0x0000000000400f0e <+101>: add $0x28,%rsp

---Type <return> to continue, or q <return> to quit---

0x0000000000400f12 <+105>: pop %rbx

0x0000000000400f13 <+106>: pop %rbp

0x0000000000400f14 <+107>: retq

End of assembler dump.

(gdb) ni

0x0000000000400ee1 in phase\_2 ()

(gdb) disas

Dump of assembler code for function phase\_2:

0x0000000000400ea9 <+0>: push %rbp

0x0000000000400eaa <+1>: push %rbx

0x0000000000400eab <+2>: sub $0x28,%rsp

0x0000000000400eaf <+6>: mov %fs:0x28,%rax

0x0000000000400eb8 <+15>: mov %rax,0x18(%rsp)

0x0000000000400ebd <+20>: xor %eax,%eax

0x0000000000400ebf <+22>: mov %rsp,%rsi

0x0000000000400ec2 <+25>: callq 0x40145f <read\_six\_numbers>

0x0000000000400ec7 <+30>: cmpl $0x0,(%rsp)

0x0000000000400ecb <+34>: jne 0x400ed4 <phase\_2+43>

0x0000000000400ecd <+36>: cmpl $0x1,0x4(%rsp)

0x0000000000400ed2 <+41>: je 0x400ed9 <phase\_2+48>

0x0000000000400ed4 <+43>: callq 0x40143d <explode\_bomb>

0x0000000000400ed9 <+48>: mov %rsp,%rbx

0x0000000000400edc <+51>: lea 0x10(%rsp),%rbp

=> 0x0000000000400ee1 <+56>: mov 0x4(%rbx),%eax

0x0000000000400ee4 <+59>: add (%rbx),%eax

0x0000000000400ee6 <+61>: cmp %eax,0x8(%rbx)

0x0000000000400ee9 <+64>: je 0x400ef0 <phase\_2+71>

0x0000000000400eeb <+66>: callq 0x40143d <explode\_bomb>

0x0000000000400ef0 <+71>: add $0x4,%rbx

0x0000000000400ef4 <+75>: cmp %rbp,%rbx

0x0000000000400ef7 <+78>: jne 0x400ee1 <phase\_2+56>

0x0000000000400ef9 <+80>: mov 0x18(%rsp),%rax

0x0000000000400efe <+85>: xor %fs:0x28,%rax

0x0000000000400f07 <+94>: je 0x400f0e <phase\_2+101>

0x0000000000400f09 <+96>: callq 0x400b00 <\_\_stack\_chk\_fail@plt>

0x0000000000400f0e <+101>: add $0x28,%rsp

---Type <return> to continue, or q <return> to quit---

0x0000000000400f12 <+105>: pop %rbx

0x0000000000400f13 <+106>: pop %rbp

0x0000000000400f14 <+107>: retq

End of assembler dump.

(gdb) ni

0x0000000000400ee4 in phase\_2 ()

(gdb) disas

Dump of assembler code for function phase\_2:

0x0000000000400ea9 <+0>: push %rbp

0x0000000000400eaa <+1>: push %rbx

0x0000000000400eab <+2>: sub $0x28,%rsp

0x0000000000400eaf <+6>: mov %fs:0x28,%rax

0x0000000000400eb8 <+15>: mov %rax,0x18(%rsp)

0x0000000000400ebd <+20>: xor %eax,%eax

0x0000000000400ebf <+22>: mov %rsp,%rsi

0x0000000000400ec2 <+25>: callq 0x40145f <read\_six\_numbers>

0x0000000000400ec7 <+30>: cmpl $0x0,(%rsp)

0x0000000000400ecb <+34>: jne 0x400ed4 <phase\_2+43>

0x0000000000400ecd <+36>: cmpl $0x1,0x4(%rsp)

0x0000000000400ed2 <+41>: je 0x400ed9 <phase\_2+48>

0x0000000000400ed4 <+43>: callq 0x40143d <explode\_bomb>

0x0000000000400ed9 <+48>: mov %rsp,%rbx

0x0000000000400edc <+51>: lea 0x10(%rsp),%rbp

0x0000000000400ee1 <+56>: mov 0x4(%rbx),%eax

=> 0x0000000000400ee4 <+59>: add (%rbx),%eax

0x0000000000400ee6 <+61>: cmp %eax,0x8(%rbx)

0x0000000000400ee9 <+64>: je 0x400ef0 <phase\_2+71>

0x0000000000400eeb <+66>: callq 0x40143d <explode\_bomb>

0x0000000000400ef0 <+71>: add $0x4,%rbx

0x0000000000400ef4 <+75>: cmp %rbp,%rbx

0x0000000000400ef7 <+78>: jne 0x400ee1 <phase\_2+56>

0x0000000000400ef9 <+80>: mov 0x18(%rsp),%rax

0x0000000000400efe <+85>: xor %fs:0x28,%rax

0x0000000000400f07 <+94>: je 0x400f0e <phase\_2+101>

0x0000000000400f09 <+96>: callq 0x400b00 <\_\_stack\_chk\_fail@plt>

0x0000000000400f0e <+101>: add $0x28,%rsp

---Type <return> to continue, or q <return> to quit---

0x0000000000400f12 <+105>: pop %rbx

0x0000000000400f13 <+106>: pop %rbp

0x0000000000400f14 <+107>: retq

End of assembler dump.

(gdb) ni

0x0000000000400ee6 in phase\_2 ()

(gdb) i r

rax 0x2 2 **// the 4th input is 2 as 2 is stored in rax 64 bit which stores the input**

rbx 0x7ffffffee304 140737488282372

rcx 0x0 0

rdx 0x7ffffffee314 140737488282388

rsi 0x0 0

rdi 0x7ffffffedc70 140737488280688

rbp 0x7ffffffee310 0x7ffffffee310

rsp 0x7ffffffee300 0x7ffffffee300

r8 0x0 0

r9 0x0 0

r10 0x7fffff19ec40 140737473276992

r11 0x4025d4 4203988

r12 0x400c60 4197472

r13 0x7ffffffee420 140737488282656

r14 0x0 0

r15 0x0 0

rip 0x400ee6 0x400ee6 <phase\_2+61>

eflags 0x202 [ IF ]

cs 0x33 51

ss 0x2b 43

ds 0x0 0

es 0x0 0

fs 0x0 0

gs 0x0 0

(gdb) r

The program being debugged has been started already.

Start it from the beginning? (y or n) r

Please answer y or n.

The program being debugged has been started already.

Start it from the beginning? (y or n) y

Starting program: /mnt/d/one drive/OneDrive/Desktop/12190094/Assignment 1/bomb001/bomb

Welcome to my fiendish little bomb. You have 6 phases with

which to blow yourself up. Have a nice day!

The moon unit will be divided into two divisions.

Phase 1 defused. How about the next one?

0 1 1 3 4 5 **// test input for fourth integer again since bomb exploded in earlier one**

Breakpoint 1, 0x0000000000400ea9 in phase\_2 ()

(gdb) disas

Dump of assembler code for function phase\_2:

=> 0x0000000000400ea9 <+0>: push %rbp

0x0000000000400eaa <+1>: push %rbx

0x0000000000400eab <+2>: sub $0x28,%rsp

0x0000000000400eaf <+6>: mov %fs:0x28,%rax

0x0000000000400eb8 <+15>: mov %rax,0x18(%rsp)

0x0000000000400ebd <+20>: xor %eax,%eax

0x0000000000400ebf <+22>: mov %rsp,%rsi

0x0000000000400ec2 <+25>: callq 0x40145f <read\_six\_numbers>

0x0000000000400ec7 <+30>: cmpl $0x0,(%rsp)

0x0000000000400ecb <+34>: jne 0x400ed4 <phase\_2+43>

0x0000000000400ecd <+36>: cmpl $0x1,0x4(%rsp)

0x0000000000400ed2 <+41>: je 0x400ed9 <phase\_2+48>

0x0000000000400ed4 <+43>: callq 0x40143d <explode\_bomb>

0x0000000000400ed9 <+48>: mov %rsp,%rbx

0x0000000000400edc <+51>: lea 0x10(%rsp),%rbp

0x0000000000400ee1 <+56>: mov 0x4(%rbx),%eax

0x0000000000400ee4 <+59>: add (%rbx),%eax

0x0000000000400ee6 <+61>: cmp %eax,0x8(%rbx)

0x0000000000400ee9 <+64>: je 0x400ef0 <phase\_2+71>

0x0000000000400eeb <+66>: callq 0x40143d <explode\_bomb>

0x0000000000400ef0 <+71>: add $0x4,%rbx

0x0000000000400ef4 <+75>: cmp %rbp,%rbx

0x0000000000400ef7 <+78>: jne 0x400ee1 <phase\_2+56>

0x0000000000400ef9 <+80>: mov 0x18(%rsp),%rax

0x0000000000400efe <+85>: xor %fs:0x28,%rax

0x0000000000400f07 <+94>: je 0x400f0e <phase\_2+101>

0x0000000000400f09 <+96>: callq 0x400b00 <\_\_stack\_chk\_fail@plt>

0x0000000000400f0e <+101>: add $0x28,%rsp

---Type <return> to continue, or q <return> to quit---

0x0000000000400f12 <+105>: pop %rbx

0x0000000000400f13 <+106>: pop %rbp

0x0000000000400f14 <+107>: retq

End of assembler dump.

(gdb) until \* 0x0000000000400ef4

0x0000000000400ef4 in phase\_2 ()

(gdb) disas

Dump of assembler code for function phase\_2:

0x0000000000400ea9 <+0>: push %rbp

0x0000000000400eaa <+1>: push %rbx

0x0000000000400eab <+2>: sub $0x28,%rsp

0x0000000000400eaf <+6>: mov %fs:0x28,%rax

0x0000000000400eb8 <+15>: mov %rax,0x18(%rsp)

0x0000000000400ebd <+20>: xor %eax,%eax

0x0000000000400ebf <+22>: mov %rsp,%rsi

0x0000000000400ec2 <+25>: callq 0x40145f <read\_six\_numbers>

0x0000000000400ec7 <+30>: cmpl $0x0,(%rsp)

0x0000000000400ecb <+34>: jne 0x400ed4 <phase\_2+43>

0x0000000000400ecd <+36>: cmpl $0x1,0x4(%rsp)

0x0000000000400ed2 <+41>: je 0x400ed9 <phase\_2+48>

0x0000000000400ed4 <+43>: callq 0x40143d <explode\_bomb>

0x0000000000400ed9 <+48>: mov %rsp,%rbx

0x0000000000400edc <+51>: lea 0x10(%rsp),%rbp

0x0000000000400ee1 <+56>: mov 0x4(%rbx),%eax

0x0000000000400ee4 <+59>: add (%rbx),%eax

0x0000000000400ee6 <+61>: cmp %eax,0x8(%rbx)

0x0000000000400ee9 <+64>: je 0x400ef0 <phase\_2+71>

0x0000000000400eeb <+66>: callq 0x40143d <explode\_bomb>

0x0000000000400ef0 <+71>: add $0x4,%rbx

=> 0x0000000000400ef4 <+75>: cmp %rbp,%rbx

0x0000000000400ef7 <+78>: jne 0x400ee1 <phase\_2+56>

0x0000000000400ef9 <+80>: mov 0x18(%rsp),%rax

0x0000000000400efe <+85>: xor %fs:0x28,%rax

0x0000000000400f07 <+94>: je 0x400f0e <phase\_2+101>

0x0000000000400f09 <+96>: callq 0x400b00 <\_\_stack\_chk\_fail@plt>

0x0000000000400f0e <+101>: add $0x28,%rsp

---Type <return> to continue, or q <return> to quit---

0x0000000000400f12 <+105>: pop %rbx

0x0000000000400f13 <+106>: pop %rbp

0x0000000000400f14 <+107>: retq

End of assembler dump.

(gdb) i r

rax 0x1 1

rbx 0x7ffffffee304 140737488282372

rcx 0x0 0

rdx 0x7ffffffee314 140737488282388

rsi 0x0 0

rdi 0x7ffffffedc70 140737488280688

rbp 0x7ffffffee310 0x7ffffffee310

rsp 0x7ffffffee300 0x7ffffffee300

r8 0x0 0

r9 0x0 0

r10 0x7fffff19ec40 140737473276992

r11 0x4025d4 4203988

r12 0x400c60 4197472

r13 0x7ffffffee420 140737488282656

r14 0x0 0

r15 0x0 0

rip 0x400ef4 0x400ef4 <phase\_2+75>

eflags 0x202 [ IF ]

cs 0x33 51

ss 0x2b 43

ds 0x0 0

es 0x0 0

fs 0x0 0

gs 0x0 0

(gdb) ni

0x0000000000400ef7 in phase\_2 ()

(gdb) disas

Dump of assembler code for function phase\_2:

0x0000000000400ea9 <+0>: push %rbp

0x0000000000400eaa <+1>: push %rbx

0x0000000000400eab <+2>: sub $0x28,%rsp

0x0000000000400eaf <+6>: mov %fs:0x28,%rax

0x0000000000400eb8 <+15>: mov %rax,0x18(%rsp)

0x0000000000400ebd <+20>: xor %eax,%eax

0x0000000000400ebf <+22>: mov %rsp,%rsi

0x0000000000400ec2 <+25>: callq 0x40145f <read\_six\_numbers>

0x0000000000400ec7 <+30>: cmpl $0x0,(%rsp)

0x0000000000400ecb <+34>: jne 0x400ed4 <phase\_2+43>

0x0000000000400ecd <+36>: cmpl $0x1,0x4(%rsp)

0x0000000000400ed2 <+41>: je 0x400ed9 <phase\_2+48>

0x0000000000400ed4 <+43>: callq 0x40143d <explode\_bomb>

0x0000000000400ed9 <+48>: mov %rsp,%rbx

0x0000000000400edc <+51>: lea 0x10(%rsp),%rbp

0x0000000000400ee1 <+56>: mov 0x4(%rbx),%eax

0x0000000000400ee4 <+59>: add (%rbx),%eax

0x0000000000400ee6 <+61>: cmp %eax,0x8(%rbx)

0x0000000000400ee9 <+64>: je 0x400ef0 <phase\_2+71>

0x0000000000400eeb <+66>: callq 0x40143d <explode\_bomb>

0x0000000000400ef0 <+71>: add $0x4,%rbx

0x0000000000400ef4 <+75>: cmp %rbp,%rbx

=> 0x0000000000400ef7 <+78>: jne 0x400ee1 <phase\_2+56>

0x0000000000400ef9 <+80>: mov 0x18(%rsp),%rax

0x0000000000400efe <+85>: xor %fs:0x28,%rax

0x0000000000400f07 <+94>: je 0x400f0e <phase\_2+101>

0x0000000000400f09 <+96>: callq 0x400b00 <\_\_stack\_chk\_fail@plt>

0x0000000000400f0e <+101>: add $0x28,%rsp

---Type <return> to continue, or q <return> to quit---

0x0000000000400f12 <+105>: pop %rbx

0x0000000000400f13 <+106>: pop %rbp

0x0000000000400f14 <+107>: retq

End of assembler dump.

(gdb) ni

0x0000000000400ee1 in phase\_2 ()

(gdb) disas

Dump of assembler code for function phase\_2:

0x0000000000400ea9 <+0>: push %rbp

0x0000000000400eaa <+1>: push %rbx

0x0000000000400eab <+2>: sub $0x28,%rsp

0x0000000000400eaf <+6>: mov %fs:0x28,%rax

0x0000000000400eb8 <+15>: mov %rax,0x18(%rsp)

0x0000000000400ebd <+20>: xor %eax,%eax

0x0000000000400ebf <+22>: mov %rsp,%rsi

0x0000000000400ec2 <+25>: callq 0x40145f <read\_six\_numbers>

0x0000000000400ec7 <+30>: cmpl $0x0,(%rsp)

0x0000000000400ecb <+34>: jne 0x400ed4 <phase\_2+43>

0x0000000000400ecd <+36>: cmpl $0x1,0x4(%rsp)

0x0000000000400ed2 <+41>: je 0x400ed9 <phase\_2+48>

0x0000000000400ed4 <+43>: callq 0x40143d <explode\_bomb>

0x0000000000400ed9 <+48>: mov %rsp,%rbx

0x0000000000400edc <+51>: lea 0x10(%rsp),%rbp

=> 0x0000000000400ee1 <+56>: mov 0x4(%rbx),%eax

0x0000000000400ee4 <+59>: add (%rbx),%eax

0x0000000000400ee6 <+61>: cmp %eax,0x8(%rbx)

0x0000000000400ee9 <+64>: je 0x400ef0 <phase\_2+71>

0x0000000000400eeb <+66>: callq 0x40143d <explode\_bomb>

0x0000000000400ef0 <+71>: add $0x4,%rbx

0x0000000000400ef4 <+75>: cmp %rbp,%rbx

0x0000000000400ef7 <+78>: jne 0x400ee1 <phase\_2+56>

0x0000000000400ef9 <+80>: mov 0x18(%rsp),%rax

0x0000000000400efe <+85>: xor %fs:0x28,%rax

0x0000000000400f07 <+94>: je 0x400f0e <phase\_2+101>

0x0000000000400f09 <+96>: callq 0x400b00 <\_\_stack\_chk\_fail@plt>

0x0000000000400f0e <+101>: add $0x28,%rsp

---Type <return> to continue, or q <return> to quit---

0x0000000000400f12 <+105>: pop %rbx

0x0000000000400f13 <+106>: pop %rbp

0x0000000000400f14 <+107>: retq

End of assembler dump.

(gdb) ni

0x0000000000400ee4 in phase\_2 ()

(gdb) disas

Dump of assembler code for function phase\_2:

0x0000000000400ea9 <+0>: push %rbp

0x0000000000400eaa <+1>: push %rbx

0x0000000000400eab <+2>: sub $0x28,%rsp

0x0000000000400eaf <+6>: mov %fs:0x28,%rax

0x0000000000400eb8 <+15>: mov %rax,0x18(%rsp)

0x0000000000400ebd <+20>: xor %eax,%eax

0x0000000000400ebf <+22>: mov %rsp,%rsi

0x0000000000400ec2 <+25>: callq 0x40145f <read\_six\_numbers>

0x0000000000400ec7 <+30>: cmpl $0x0,(%rsp)

0x0000000000400ecb <+34>: jne 0x400ed4 <phase\_2+43>

0x0000000000400ecd <+36>: cmpl $0x1,0x4(%rsp)

0x0000000000400ed2 <+41>: je 0x400ed9 <phase\_2+48>

0x0000000000400ed4 <+43>: callq 0x40143d <explode\_bomb>

0x0000000000400ed9 <+48>: mov %rsp,%rbx

0x0000000000400edc <+51>: lea 0x10(%rsp),%rbp

0x0000000000400ee1 <+56>: mov 0x4(%rbx),%eax

=> 0x0000000000400ee4 <+59>: add (%rbx),%eax

0x0000000000400ee6 <+61>: cmp %eax,0x8(%rbx)

0x0000000000400ee9 <+64>: je 0x400ef0 <phase\_2+71>

0x0000000000400eeb <+66>: callq 0x40143d <explode\_bomb>

0x0000000000400ef0 <+71>: add $0x4,%rbx

0x0000000000400ef4 <+75>: cmp %rbp,%rbx

0x0000000000400ef7 <+78>: jne 0x400ee1 <phase\_2+56>

0x0000000000400ef9 <+80>: mov 0x18(%rsp),%rax

0x0000000000400efe <+85>: xor %fs:0x28,%rax

0x0000000000400f07 <+94>: je 0x400f0e <phase\_2+101>

0x0000000000400f09 <+96>: callq 0x400b00 <\_\_stack\_chk\_fail@plt>

0x0000000000400f0e <+101>: add $0x28,%rsp

---Type <return> to continue, or q <return> to quit---

0x0000000000400f12 <+105>: pop %rbx

0x0000000000400f13 <+106>: pop %rbp

0x0000000000400f14 <+107>: retq

End of assembler dump.

(gdb) ni

0x0000000000400ee6 in phase\_2 ()

(gdb) disas

Dump of assembler code for function phase\_2:

0x0000000000400ea9 <+0>: push %rbp

0x0000000000400eaa <+1>: push %rbx

0x0000000000400eab <+2>: sub $0x28,%rsp

0x0000000000400eaf <+6>: mov %fs:0x28,%rax

0x0000000000400eb8 <+15>: mov %rax,0x18(%rsp)

0x0000000000400ebd <+20>: xor %eax,%eax

0x0000000000400ebf <+22>: mov %rsp,%rsi

0x0000000000400ec2 <+25>: callq 0x40145f <read\_six\_numbers>

0x0000000000400ec7 <+30>: cmpl $0x0,(%rsp)

0x0000000000400ecb <+34>: jne 0x400ed4 <phase\_2+43>

0x0000000000400ecd <+36>: cmpl $0x1,0x4(%rsp)

0x0000000000400ed2 <+41>: je 0x400ed9 <phase\_2+48>

0x0000000000400ed4 <+43>: callq 0x40143d <explode\_bomb>

0x0000000000400ed9 <+48>: mov %rsp,%rbx

0x0000000000400edc <+51>: lea 0x10(%rsp),%rbp

0x0000000000400ee1 <+56>: mov 0x4(%rbx),%eax

0x0000000000400ee4 <+59>: add (%rbx),%eax

=> 0x0000000000400ee6 <+61>: cmp %eax,0x8(%rbx)

0x0000000000400ee9 <+64>: je 0x400ef0 <phase\_2+71>

0x0000000000400eeb <+66>: callq 0x40143d <explode\_bomb>

0x0000000000400ef0 <+71>: add $0x4,%rbx

0x0000000000400ef4 <+75>: cmp %rbp,%rbx

0x0000000000400ef7 <+78>: jne 0x400ee1 <phase\_2+56>

0x0000000000400ef9 <+80>: mov 0x18(%rsp),%rax

0x0000000000400efe <+85>: xor %fs:0x28,%rax

0x0000000000400f07 <+94>: je 0x400f0e <phase\_2+101>

0x0000000000400f09 <+96>: callq 0x400b00 <\_\_stack\_chk\_fail@plt>

0x0000000000400f0e <+101>: add $0x28,%rsp

---Type <return> to continue, or q <return> to quit---

0x0000000000400f12 <+105>: pop %rbx

0x0000000000400f13 <+106>: pop %rbp

0x0000000000400f14 <+107>: retq

End of assembler dump.

(gdb) i r

rax 0x2 2 **//The fourth input is 2 because the input stored in rax 64 bits is 1 as eax 32 bit stores the input and since the bomb exploded we had to get it again**

rbx 0x7ffffffee304 140737488282372

rcx 0x0 0

rdx 0x7ffffffee314 140737488282388

rsi 0x0 0

rdi 0x7ffffffedc70 140737488280688

rbp 0x7ffffffee310 0x7ffffffee310

rsp 0x7ffffffee300 0x7ffffffee300

r8 0x0 0

r9 0x0 0

r10 0x7fffff19ec40 140737473276992

r11 0x4025d4 4203988

r12 0x400c60 4197472

r13 0x7ffffffee420 140737488282656

r14 0x0 0

r15 0x0 0

rip 0x400ee6 0x400ee6 <phase\_2+61>

eflags 0x202 [ IF ]

cs 0x33 51

ss 0x2b 43

ds 0x0 0

es 0x0 0

fs 0x0 0

gs 0x0 0

(gdb) ni

0x0000000000400ee9 in phase\_2 ()

(gdb) disas

Dump of assembler code for function phase\_2:

0x0000000000400ea9 <+0>: push %rbp

0x0000000000400eaa <+1>: push %rbx

0x0000000000400eab <+2>: sub $0x28,%rsp

0x0000000000400eaf <+6>: mov %fs:0x28,%rax

0x0000000000400eb8 <+15>: mov %rax,0x18(%rsp)

0x0000000000400ebd <+20>: xor %eax,%eax

0x0000000000400ebf <+22>: mov %rsp,%rsi

0x0000000000400ec2 <+25>: callq 0x40145f <read\_six\_numbers>

0x0000000000400ec7 <+30>: cmpl $0x0,(%rsp)

0x0000000000400ecb <+34>: jne 0x400ed4 <phase\_2+43>

0x0000000000400ecd <+36>: cmpl $0x1,0x4(%rsp)

0x0000000000400ed2 <+41>: je 0x400ed9 <phase\_2+48>

0x0000000000400ed4 <+43>: callq 0x40143d <explode\_bomb>

0x0000000000400ed9 <+48>: mov %rsp,%rbx

0x0000000000400edc <+51>: lea 0x10(%rsp),%rbp

0x0000000000400ee1 <+56>: mov 0x4(%rbx),%eax

0x0000000000400ee4 <+59>: add (%rbx),%eax

0x0000000000400ee6 <+61>: cmp %eax,0x8(%rbx)

=> 0x0000000000400ee9 <+64>: je 0x400ef0 <phase\_2+71>

0x0000000000400eeb <+66>: callq 0x40143d <explode\_bomb>

0x0000000000400ef0 <+71>: add $0x4,%rbx

0x0000000000400ef4 <+75>: cmp %rbp,%rbx

0x0000000000400ef7 <+78>: jne 0x400ee1 <phase\_2+56>

0x0000000000400ef9 <+80>: mov 0x18(%rsp),%rax

0x0000000000400efe <+85>: xor %fs:0x28,%rax

0x0000000000400f07 <+94>: je 0x400f0e <phase\_2+101>

0x0000000000400f09 <+96>: callq 0x400b00 <\_\_stack\_chk\_fail@plt>

0x0000000000400f0e <+101>: add $0x28,%rsp

---Type <return> to continue, or q <return> to quit---

0x0000000000400f12 <+105>: pop %rbx

0x0000000000400f13 <+106>: pop %rbp

0x0000000000400f14 <+107>: retq

End of assembler dump.

(gdb) ni

0x0000000000400eeb in phase\_2 ()

(gdb) disas

Dump of assembler code for function phase\_2:

0x0000000000400ea9 <+0>: push %rbp

0x0000000000400eaa <+1>: push %rbx

0x0000000000400eab <+2>: sub $0x28,%rsp

0x0000000000400eaf <+6>: mov %fs:0x28,%rax

0x0000000000400eb8 <+15>: mov %rax,0x18(%rsp)

0x0000000000400ebd <+20>: xor %eax,%eax

0x0000000000400ebf <+22>: mov %rsp,%rsi

0x0000000000400ec2 <+25>: callq 0x40145f <read\_six\_numbers>

0x0000000000400ec7 <+30>: cmpl $0x0,(%rsp)

0x0000000000400ecb <+34>: jne 0x400ed4 <phase\_2+43>

0x0000000000400ecd <+36>: cmpl $0x1,0x4(%rsp)

0x0000000000400ed2 <+41>: je 0x400ed9 <phase\_2+48>

0x0000000000400ed4 <+43>: callq 0x40143d <explode\_bomb>

0x0000000000400ed9 <+48>: mov %rsp,%rbx

0x0000000000400edc <+51>: lea 0x10(%rsp),%rbp

0x0000000000400ee1 <+56>: mov 0x4(%rbx),%eax

0x0000000000400ee4 <+59>: add (%rbx),%eax

0x0000000000400ee6 <+61>: cmp %eax,0x8(%rbx)

0x0000000000400ee9 <+64>: je 0x400ef0 <phase\_2+71>

=> 0x0000000000400eeb <+66>: callq 0x40143d <explode\_bomb>

0x0000000000400ef0 <+71>: add $0x4,%rbx

0x0000000000400ef4 <+75>: cmp %rbp,%rbx

0x0000000000400ef7 <+78>: jne 0x400ee1 <phase\_2+56>

0x0000000000400ef9 <+80>: mov 0x18(%rsp),%rax

0x0000000000400efe <+85>: xor %fs:0x28,%rax

0x0000000000400f07 <+94>: je 0x400f0e <phase\_2+101>

0x0000000000400f09 <+96>: callq 0x400b00 <\_\_stack\_chk\_fail@plt>

0x0000000000400f0e <+101>: add $0x28,%rsp

---Type <return> to continue, or q <return> to quit---

0x0000000000400f12 <+105>: pop %rbx

0x0000000000400f13 <+106>: pop %rbp

0x0000000000400f14 <+107>: retq

End of assembler dump.

(gdb) ni

Breakpoint 2, 0x000000000040143d in explode\_bomb ()

(gdb) disas

Dump of assembler code for function explode\_bomb:

=> 0x000000000040143d <+0>: sub $0x8,%rsp

0x0000000000401441 <+4>: mov $0x4025a3,%edi

0x0000000000401446 <+9>: callq 0x400ae0 <puts@plt>

0x000000000040144b <+14>: mov $0x4025ac,%edi

0x0000000000401450 <+19>: callq 0x400ae0 <puts@plt>

0x0000000000401455 <+24>: mov $0x8,%edi

0x000000000040145a <+29>: callq 0x400be0 <exit@plt>

End of assembler dump.

(gdb) r

The program being debugged has been started already.

Start it from the beginning? (y or n) y

Starting program: /mnt/d/one drive/OneDrive/Desktop/12190094/Assignment 1/bomb001/bomb

Welcome to my fiendish little bomb. You have 6 phases with

which to blow yourself up. Have a nice day!

The moon unit will be divided into two divisions.

Phase 1 defused. How about the next one?

0 1 1 2 4 5 **// test input for fifth integer**

Breakpoint 1, 0x0000000000400ea9 in phase\_2 ()

(gdb) disas

Dump of assembler code for function phase\_2:

=> 0x0000000000400ea9 <+0>: push %rbp

0x0000000000400eaa <+1>: push %rbx

0x0000000000400eab <+2>: sub $0x28,%rsp

0x0000000000400eaf <+6>: mov %fs:0x28,%rax

0x0000000000400eb8 <+15>: mov %rax,0x18(%rsp)

0x0000000000400ebd <+20>: xor %eax,%eax

0x0000000000400ebf <+22>: mov %rsp,%rsi

0x0000000000400ec2 <+25>: callq 0x40145f <read\_six\_numbers>

0x0000000000400ec7 <+30>: cmpl $0x0,(%rsp)

0x0000000000400ecb <+34>: jne 0x400ed4 <phase\_2+43>

0x0000000000400ecd <+36>: cmpl $0x1,0x4(%rsp)

0x0000000000400ed2 <+41>: je 0x400ed9 <phase\_2+48>

0x0000000000400ed4 <+43>: callq 0x40143d <explode\_bomb>

0x0000000000400ed9 <+48>: mov %rsp,%rbx

0x0000000000400edc <+51>: lea 0x10(%rsp),%rbp

0x0000000000400ee1 <+56>: mov 0x4(%rbx),%eax

0x0000000000400ee4 <+59>: add (%rbx),%eax

0x0000000000400ee6 <+61>: cmp %eax,0x8(%rbx)

0x0000000000400ee9 <+64>: je 0x400ef0 <phase\_2+71>

0x0000000000400eeb <+66>: callq 0x40143d <explode\_bomb>

0x0000000000400ef0 <+71>: add $0x4,%rbx

0x0000000000400ef4 <+75>: cmp %rbp,%rbx

0x0000000000400ef7 <+78>: jne 0x400ee1 <phase\_2+56>

0x0000000000400ef9 <+80>: mov 0x18(%rsp),%rax

0x0000000000400efe <+85>: xor %fs:0x28,%rax

0x0000000000400f07 <+94>: je 0x400f0e <phase\_2+101>

0x0000000000400f09 <+96>: callq 0x400b00 <\_\_stack\_chk\_fail@plt>

0x0000000000400f0e <+101>: add $0x28,%rsp

---Type <return> to continue, or q <return> to quit---

0x0000000000400f12 <+105>: pop %rbx

0x0000000000400f13 <+106>: pop %rbp

0x0000000000400f14 <+107>: retq

End of assembler dump.

(gdb) until \* 0x0000000000400ee6

0x0000000000400ee6 in phase\_2 ()

(gdb) disas

Dump of assembler code for function phase\_2:

0x0000000000400ea9 <+0>: push %rbp

0x0000000000400eaa <+1>: push %rbx

0x0000000000400eab <+2>: sub $0x28,%rsp

0x0000000000400eaf <+6>: mov %fs:0x28,%rax

0x0000000000400eb8 <+15>: mov %rax,0x18(%rsp)

0x0000000000400ebd <+20>: xor %eax,%eax

0x0000000000400ebf <+22>: mov %rsp,%rsi

0x0000000000400ec2 <+25>: callq 0x40145f <read\_six\_numbers>

0x0000000000400ec7 <+30>: cmpl $0x0,(%rsp)

0x0000000000400ecb <+34>: jne 0x400ed4 <phase\_2+43>

0x0000000000400ecd <+36>: cmpl $0x1,0x4(%rsp)

0x0000000000400ed2 <+41>: je 0x400ed9 <phase\_2+48>

0x0000000000400ed4 <+43>: callq 0x40143d <explode\_bomb>

0x0000000000400ed9 <+48>: mov %rsp,%rbx

0x0000000000400edc <+51>: lea 0x10(%rsp),%rbp

0x0000000000400ee1 <+56>: mov 0x4(%rbx),%eax

0x0000000000400ee4 <+59>: add (%rbx),%eax

=> 0x0000000000400ee6 <+61>: cmp %eax,0x8(%rbx)

0x0000000000400ee9 <+64>: je 0x400ef0 <phase\_2+71>

0x0000000000400eeb <+66>: callq 0x40143d <explode\_bomb>

0x0000000000400ef0 <+71>: add $0x4,%rbx

0x0000000000400ef4 <+75>: cmp %rbp,%rbx

0x0000000000400ef7 <+78>: jne 0x400ee1 <phase\_2+56>

0x0000000000400ef9 <+80>: mov 0x18(%rsp),%rax

0x0000000000400efe <+85>: xor %fs:0x28,%rax

0x0000000000400f07 <+94>: je 0x400f0e <phase\_2+101>

0x0000000000400f09 <+96>: callq 0x400b00 <\_\_stack\_chk\_fail@plt>

0x0000000000400f0e <+101>: add $0x28,%rsp

---Type <return> to continue, or q <return> to quit---

0x0000000000400f12 <+105>: pop %rbx

0x0000000000400f13 <+106>: pop %rbp

0x0000000000400f14 <+107>: retq

End of assembler dump.

(gdb) i r

rax 0x1 1

rbx 0x7ffffffee300 140737488282368

rcx 0x0 0

rdx 0x7ffffffee314 140737488282388

rsi 0x0 0

rdi 0x7ffffffedc70 140737488280688

rbp 0x7ffffffee310 0x7ffffffee310

rsp 0x7ffffffee300 0x7ffffffee300

r8 0x0 0

r9 0x0 0

r10 0x7fffff19ec40 140737473276992

r11 0x4025d4 4203988

r12 0x400c60 4197472

r13 0x7ffffffee420 140737488282656

r14 0x0 0

r15 0x0 0

rip 0x400ee6 0x400ee6 <phase\_2+61>

eflags 0x202 [ IF ]

cs 0x33 51

ss 0x2b 43

ds 0x0 0

es 0x0 0

fs 0x0 0

gs 0x0 0

(gdb) ni

0x0000000000400ee9 in phase\_2 ()

(gdb) disas

Dump of assembler code for function phase\_2:

0x0000000000400ea9 <+0>: push %rbp

0x0000000000400eaa <+1>: push %rbx

0x0000000000400eab <+2>: sub $0x28,%rsp

0x0000000000400eaf <+6>: mov %fs:0x28,%rax

0x0000000000400eb8 <+15>: mov %rax,0x18(%rsp)

0x0000000000400ebd <+20>: xor %eax,%eax

0x0000000000400ebf <+22>: mov %rsp,%rsi

0x0000000000400ec2 <+25>: callq 0x40145f <read\_six\_numbers>

0x0000000000400ec7 <+30>: cmpl $0x0,(%rsp)

0x0000000000400ecb <+34>: jne 0x400ed4 <phase\_2+43>

0x0000000000400ecd <+36>: cmpl $0x1,0x4(%rsp)

0x0000000000400ed2 <+41>: je 0x400ed9 <phase\_2+48>

0x0000000000400ed4 <+43>: callq 0x40143d <explode\_bomb>

0x0000000000400ed9 <+48>: mov %rsp,%rbx

0x0000000000400edc <+51>: lea 0x10(%rsp),%rbp

0x0000000000400ee1 <+56>: mov 0x4(%rbx),%eax

0x0000000000400ee4 <+59>: add (%rbx),%eax

0x0000000000400ee6 <+61>: cmp %eax,0x8(%rbx)

=> 0x0000000000400ee9 <+64>: je 0x400ef0 <phase\_2+71>

0x0000000000400eeb <+66>: callq 0x40143d <explode\_bomb>

0x0000000000400ef0 <+71>: add $0x4,%rbx

0x0000000000400ef4 <+75>: cmp %rbp,%rbx

0x0000000000400ef7 <+78>: jne 0x400ee1 <phase\_2+56>

0x0000000000400ef9 <+80>: mov 0x18(%rsp),%rax

0x0000000000400efe <+85>: xor %fs:0x28,%rax

0x0000000000400f07 <+94>: je 0x400f0e <phase\_2+101>

0x0000000000400f09 <+96>: callq 0x400b00 <\_\_stack\_chk\_fail@plt>

0x0000000000400f0e <+101>: add $0x28,%rsp

---Type <return> to continue, or q <return> to quit---

0x0000000000400f12 <+105>: pop %rbx

0x0000000000400f13 <+106>: pop %rbp

0x0000000000400f14 <+107>: retq

End of assembler dump.

(gdb) ni

0x0000000000400ef0 in phase\_2 ()

(gdb) disas

Dump of assembler code for function phase\_2:

0x0000000000400ea9 <+0>: push %rbp

0x0000000000400eaa <+1>: push %rbx

0x0000000000400eab <+2>: sub $0x28,%rsp

0x0000000000400eaf <+6>: mov %fs:0x28,%rax

0x0000000000400eb8 <+15>: mov %rax,0x18(%rsp)

0x0000000000400ebd <+20>: xor %eax,%eax

0x0000000000400ebf <+22>: mov %rsp,%rsi

0x0000000000400ec2 <+25>: callq 0x40145f <read\_six\_numbers>

0x0000000000400ec7 <+30>: cmpl $0x0,(%rsp)

0x0000000000400ecb <+34>: jne 0x400ed4 <phase\_2+43>

0x0000000000400ecd <+36>: cmpl $0x1,0x4(%rsp)

0x0000000000400ed2 <+41>: je 0x400ed9 <phase\_2+48>

0x0000000000400ed4 <+43>: callq 0x40143d <explode\_bomb>

0x0000000000400ed9 <+48>: mov %rsp,%rbx

0x0000000000400edc <+51>: lea 0x10(%rsp),%rbp

0x0000000000400ee1 <+56>: mov 0x4(%rbx),%eax

0x0000000000400ee4 <+59>: add (%rbx),%eax

0x0000000000400ee6 <+61>: cmp %eax,0x8(%rbx)

0x0000000000400ee9 <+64>: je 0x400ef0 <phase\_2+71>

0x0000000000400eeb <+66>: callq 0x40143d <explode\_bomb>

=> 0x0000000000400ef0 <+71>: add $0x4,%rbx

0x0000000000400ef4 <+75>: cmp %rbp,%rbx

0x0000000000400ef7 <+78>: jne 0x400ee1 <phase\_2+56>

0x0000000000400ef9 <+80>: mov 0x18(%rsp),%rax

0x0000000000400efe <+85>: xor %fs:0x28,%rax

0x0000000000400f07 <+94>: je 0x400f0e <phase\_2+101>

0x0000000000400f09 <+96>: callq 0x400b00 <\_\_stack\_chk\_fail@plt>

0x0000000000400f0e <+101>: add $0x28,%rsp

---Type <return> to continue, or q <return> to quit---

0x0000000000400f12 <+105>: pop %rbx

0x0000000000400f13 <+106>: pop %rbp

0x0000000000400f14 <+107>: retq

End of assembler dump.

(gdb) ni

0x0000000000400ef4 in phase\_2 ()

(gdb) disas

Dump of assembler code for function phase\_2:

0x0000000000400ea9 <+0>: push %rbp

0x0000000000400eaa <+1>: push %rbx

0x0000000000400eab <+2>: sub $0x28,%rsp

0x0000000000400eaf <+6>: mov %fs:0x28,%rax

0x0000000000400eb8 <+15>: mov %rax,0x18(%rsp)

0x0000000000400ebd <+20>: xor %eax,%eax

0x0000000000400ebf <+22>: mov %rsp,%rsi

0x0000000000400ec2 <+25>: callq 0x40145f <read\_six\_numbers>

0x0000000000400ec7 <+30>: cmpl $0x0,(%rsp)

0x0000000000400ecb <+34>: jne 0x400ed4 <phase\_2+43>

0x0000000000400ecd <+36>: cmpl $0x1,0x4(%rsp)

0x0000000000400ed2 <+41>: je 0x400ed9 <phase\_2+48>

0x0000000000400ed4 <+43>: callq 0x40143d <explode\_bomb>

0x0000000000400ed9 <+48>: mov %rsp,%rbx

0x0000000000400edc <+51>: lea 0x10(%rsp),%rbp

0x0000000000400ee1 <+56>: mov 0x4(%rbx),%eax

0x0000000000400ee4 <+59>: add (%rbx),%eax

0x0000000000400ee6 <+61>: cmp %eax,0x8(%rbx)

0x0000000000400ee9 <+64>: je 0x400ef0 <phase\_2+71>

0x0000000000400eeb <+66>: callq 0x40143d <explode\_bomb>

0x0000000000400ef0 <+71>: add $0x4,%rbx

=> 0x0000000000400ef4 <+75>: cmp %rbp,%rbx

0x0000000000400ef7 <+78>: jne 0x400ee1 <phase\_2+56>

0x0000000000400ef9 <+80>: mov 0x18(%rsp),%rax

0x0000000000400efe <+85>: xor %fs:0x28,%rax

0x0000000000400f07 <+94>: je 0x400f0e <phase\_2+101>

0x0000000000400f09 <+96>: callq 0x400b00 <\_\_stack\_chk\_fail@plt>

0x0000000000400f0e <+101>: add $0x28,%rsp

---Type <return> to continue, or q <return> to quit---

0x0000000000400f12 <+105>: pop %rbx

0x0000000000400f13 <+106>: pop %rbp

0x0000000000400f14 <+107>: retq

End of assembler dump.

(gdb) i r

rax 0x1 1

rbx 0x7ffffffee304 140737488282372

rcx 0x0 0

rdx 0x7ffffffee314 140737488282388

rsi 0x0 0

rdi 0x7ffffffedc70 140737488280688

rbp 0x7ffffffee310 0x7ffffffee310

rsp 0x7ffffffee300 0x7ffffffee300

r8 0x0 0

r9 0x0 0

r10 0x7fffff19ec40 140737473276992

r11 0x4025d4 4203988

r12 0x400c60 4197472

r13 0x7ffffffee420 140737488282656

r14 0x0 0

r15 0x0 0

rip 0x400ef4 0x400ef4 <phase\_2+75>

eflags 0x202 [ IF ]

cs 0x33 51

ss 0x2b 43

ds 0x0 0

es 0x0 0

fs 0x0 0

gs 0x0 0

(gdb) ni

0x0000000000400ef7 in phase\_2 ()

(gdb) disas

Dump of assembler code for function phase\_2:

0x0000000000400ea9 <+0>: push %rbp

0x0000000000400eaa <+1>: push %rbx

0x0000000000400eab <+2>: sub $0x28,%rsp

0x0000000000400eaf <+6>: mov %fs:0x28,%rax

0x0000000000400eb8 <+15>: mov %rax,0x18(%rsp)

0x0000000000400ebd <+20>: xor %eax,%eax

0x0000000000400ebf <+22>: mov %rsp,%rsi

0x0000000000400ec2 <+25>: callq 0x40145f <read\_six\_numbers>

0x0000000000400ec7 <+30>: cmpl $0x0,(%rsp)

0x0000000000400ecb <+34>: jne 0x400ed4 <phase\_2+43>

0x0000000000400ecd <+36>: cmpl $0x1,0x4(%rsp)

0x0000000000400ed2 <+41>: je 0x400ed9 <phase\_2+48>

0x0000000000400ed4 <+43>: callq 0x40143d <explode\_bomb>

0x0000000000400ed9 <+48>: mov %rsp,%rbx

0x0000000000400edc <+51>: lea 0x10(%rsp),%rbp

0x0000000000400ee1 <+56>: mov 0x4(%rbx),%eax

0x0000000000400ee4 <+59>: add (%rbx),%eax

0x0000000000400ee6 <+61>: cmp %eax,0x8(%rbx)

0x0000000000400ee9 <+64>: je 0x400ef0 <phase\_2+71>

0x0000000000400eeb <+66>: callq 0x40143d <explode\_bomb>

0x0000000000400ef0 <+71>: add $0x4,%rbx

0x0000000000400ef4 <+75>: cmp %rbp,%rbx

=> 0x0000000000400ef7 <+78>: jne 0x400ee1 <phase\_2+56>

0x0000000000400ef9 <+80>: mov 0x18(%rsp),%rax

0x0000000000400efe <+85>: xor %fs:0x28,%rax

0x0000000000400f07 <+94>: je 0x400f0e <phase\_2+101>

0x0000000000400f09 <+96>: callq 0x400b00 <\_\_stack\_chk\_fail@plt>

0x0000000000400f0e <+101>: add $0x28,%rsp

---Type <return> to continue, or q <return> to quit---

0x0000000000400f12 <+105>: pop %rbx

0x0000000000400f13 <+106>: pop %rbp

0x0000000000400f14 <+107>: retq

End of assembler dump.

(gdb) ni

0x0000000000400ee1 in phase\_2 ()

(gdb) disas

Dump of assembler code for function phase\_2:

0x0000000000400ea9 <+0>: push %rbp

0x0000000000400eaa <+1>: push %rbx

0x0000000000400eab <+2>: sub $0x28,%rsp

0x0000000000400eaf <+6>: mov %fs:0x28,%rax

0x0000000000400eb8 <+15>: mov %rax,0x18(%rsp)

0x0000000000400ebd <+20>: xor %eax,%eax

0x0000000000400ebf <+22>: mov %rsp,%rsi

0x0000000000400ec2 <+25>: callq 0x40145f <read\_six\_numbers>

0x0000000000400ec7 <+30>: cmpl $0x0,(%rsp)

0x0000000000400ecb <+34>: jne 0x400ed4 <phase\_2+43>

0x0000000000400ecd <+36>: cmpl $0x1,0x4(%rsp)

0x0000000000400ed2 <+41>: je 0x400ed9 <phase\_2+48>

0x0000000000400ed4 <+43>: callq 0x40143d <explode\_bomb>

0x0000000000400ed9 <+48>: mov %rsp,%rbx

0x0000000000400edc <+51>: lea 0x10(%rsp),%rbp

=> 0x0000000000400ee1 <+56>: mov 0x4(%rbx),%eax

0x0000000000400ee4 <+59>: add (%rbx),%eax

0x0000000000400ee6 <+61>: cmp %eax,0x8(%rbx)

0x0000000000400ee9 <+64>: je 0x400ef0 <phase\_2+71>

0x0000000000400eeb <+66>: callq 0x40143d <explode\_bomb>

0x0000000000400ef0 <+71>: add $0x4,%rbx

0x0000000000400ef4 <+75>: cmp %rbp,%rbx

0x0000000000400ef7 <+78>: jne 0x400ee1 <phase\_2+56>

0x0000000000400ef9 <+80>: mov 0x18(%rsp),%rax

0x0000000000400efe <+85>: xor %fs:0x28,%rax

0x0000000000400f07 <+94>: je 0x400f0e <phase\_2+101>

0x0000000000400f09 <+96>: callq 0x400b00 <\_\_stack\_chk\_fail@plt>

0x0000000000400f0e <+101>: add $0x28,%rsp

---Type <return> to continue, or q <return> to quit---

0x0000000000400f12 <+105>: pop %rbx

0x0000000000400f13 <+106>: pop %rbp

0x0000000000400f14 <+107>: retq

End of assembler dump.

(gdb) ni

0x0000000000400ee4 in phase\_2 ()

(gdb) disas

Dump of assembler code for function phase\_2:

0x0000000000400ea9 <+0>: push %rbp

0x0000000000400eaa <+1>: push %rbx

0x0000000000400eab <+2>: sub $0x28,%rsp

0x0000000000400eaf <+6>: mov %fs:0x28,%rax

0x0000000000400eb8 <+15>: mov %rax,0x18(%rsp)

0x0000000000400ebd <+20>: xor %eax,%eax

0x0000000000400ebf <+22>: mov %rsp,%rsi

0x0000000000400ec2 <+25>: callq 0x40145f <read\_six\_numbers>

0x0000000000400ec7 <+30>: cmpl $0x0,(%rsp)

0x0000000000400ecb <+34>: jne 0x400ed4 <phase\_2+43>

0x0000000000400ecd <+36>: cmpl $0x1,0x4(%rsp)

0x0000000000400ed2 <+41>: je 0x400ed9 <phase\_2+48>

0x0000000000400ed4 <+43>: callq 0x40143d <explode\_bomb>

0x0000000000400ed9 <+48>: mov %rsp,%rbx

0x0000000000400edc <+51>: lea 0x10(%rsp),%rbp

0x0000000000400ee1 <+56>: mov 0x4(%rbx),%eax

=> 0x0000000000400ee4 <+59>: add (%rbx),%eax

0x0000000000400ee6 <+61>: cmp %eax,0x8(%rbx)

0x0000000000400ee9 <+64>: je 0x400ef0 <phase\_2+71>

0x0000000000400eeb <+66>: callq 0x40143d <explode\_bomb>

0x0000000000400ef0 <+71>: add $0x4,%rbx

0x0000000000400ef4 <+75>: cmp %rbp,%rbx

0x0000000000400ef7 <+78>: jne 0x400ee1 <phase\_2+56>

0x0000000000400ef9 <+80>: mov 0x18(%rsp),%rax

0x0000000000400efe <+85>: xor %fs:0x28,%rax

0x0000000000400f07 <+94>: je 0x400f0e <phase\_2+101>

0x0000000000400f09 <+96>: callq 0x400b00 <\_\_stack\_chk\_fail@plt>

0x0000000000400f0e <+101>: add $0x28,%rsp

---Type <return> to continue, or q <return> to quit---

0x0000000000400f12 <+105>: pop %rbx

0x0000000000400f13 <+106>: pop %rbp

0x0000000000400f14 <+107>: retq

End of assembler dump.

(gdb) ni

0x0000000000400ee6 in phase\_2 ()

(gdb) disas

Dump of assembler code for function phase\_2:

0x0000000000400ea9 <+0>: push %rbp

0x0000000000400eaa <+1>: push %rbx

0x0000000000400eab <+2>: sub $0x28,%rsp

0x0000000000400eaf <+6>: mov %fs:0x28,%rax

0x0000000000400eb8 <+15>: mov %rax,0x18(%rsp)

0x0000000000400ebd <+20>: xor %eax,%eax

0x0000000000400ebf <+22>: mov %rsp,%rsi

0x0000000000400ec2 <+25>: callq 0x40145f <read\_six\_numbers>

0x0000000000400ec7 <+30>: cmpl $0x0,(%rsp)

0x0000000000400ecb <+34>: jne 0x400ed4 <phase\_2+43>

0x0000000000400ecd <+36>: cmpl $0x1,0x4(%rsp)

0x0000000000400ed2 <+41>: je 0x400ed9 <phase\_2+48>

0x0000000000400ed4 <+43>: callq 0x40143d <explode\_bomb>

0x0000000000400ed9 <+48>: mov %rsp,%rbx

0x0000000000400edc <+51>: lea 0x10(%rsp),%rbp

0x0000000000400ee1 <+56>: mov 0x4(%rbx),%eax

0x0000000000400ee4 <+59>: add (%rbx),%eax

=> 0x0000000000400ee6 <+61>: cmp %eax,0x8(%rbx)

0x0000000000400ee9 <+64>: je 0x400ef0 <phase\_2+71>

0x0000000000400eeb <+66>: callq 0x40143d <explode\_bomb>

0x0000000000400ef0 <+71>: add $0x4,%rbx

0x0000000000400ef4 <+75>: cmp %rbp,%rbx

0x0000000000400ef7 <+78>: jne 0x400ee1 <phase\_2+56>

0x0000000000400ef9 <+80>: mov 0x18(%rsp),%rax

0x0000000000400efe <+85>: xor %fs:0x28,%rax

0x0000000000400f07 <+94>: je 0x400f0e <phase\_2+101>

0x0000000000400f09 <+96>: callq 0x400b00 <\_\_stack\_chk\_fail@plt>

0x0000000000400f0e <+101>: add $0x28,%rsp

---Type <return> to continue, or q <return> to quit---

0x0000000000400f12 <+105>: pop %rbx

0x0000000000400f13 <+106>: pop %rbp

0x0000000000400f14 <+107>: retq

End of assembler dump.

(gdb) i r

rax 0x2 2

rbx 0x7ffffffee304 140737488282372

rcx 0x0 0

rdx 0x7ffffffee314 140737488282388

rsi 0x0 0

rdi 0x7ffffffedc70 140737488280688

rbp 0x7ffffffee310 0x7ffffffee310

rsp 0x7ffffffee300 0x7ffffffee300

r8 0x0 0

r9 0x0 0

r10 0x7fffff19ec40 140737473276992

r11 0x4025d4 4203988

r12 0x400c60 4197472

r13 0x7ffffffee420 140737488282656

r14 0x0 0

r15 0x0 0

rip 0x400ee6 0x400ee6 <phase\_2+61>

eflags 0x202 [ IF ]

cs 0x33 51

ss 0x2b 43

ds 0x0 0

es 0x0 0

fs 0x0 0

gs 0x0 0

(gdb) ni

0x0000000000400ee9 in phase\_2 ()

(gdb) disas

Dump of assembler code for function phase\_2:

0x0000000000400ea9 <+0>: push %rbp

0x0000000000400eaa <+1>: push %rbx

0x0000000000400eab <+2>: sub $0x28,%rsp

0x0000000000400eaf <+6>: mov %fs:0x28,%rax

0x0000000000400eb8 <+15>: mov %rax,0x18(%rsp)

0x0000000000400ebd <+20>: xor %eax,%eax

0x0000000000400ebf <+22>: mov %rsp,%rsi

0x0000000000400ec2 <+25>: callq 0x40145f <read\_six\_numbers>

0x0000000000400ec7 <+30>: cmpl $0x0,(%rsp)

0x0000000000400ecb <+34>: jne 0x400ed4 <phase\_2+43>

0x0000000000400ecd <+36>: cmpl $0x1,0x4(%rsp)

0x0000000000400ed2 <+41>: je 0x400ed9 <phase\_2+48>

0x0000000000400ed4 <+43>: callq 0x40143d <explode\_bomb>

0x0000000000400ed9 <+48>: mov %rsp,%rbx

0x0000000000400edc <+51>: lea 0x10(%rsp),%rbp

0x0000000000400ee1 <+56>: mov 0x4(%rbx),%eax

0x0000000000400ee4 <+59>: add (%rbx),%eax

0x0000000000400ee6 <+61>: cmp %eax,0x8(%rbx)

=> 0x0000000000400ee9 <+64>: je 0x400ef0 <phase\_2+71>

0x0000000000400eeb <+66>: callq 0x40143d <explode\_bomb>

0x0000000000400ef0 <+71>: add $0x4,%rbx

0x0000000000400ef4 <+75>: cmp %rbp,%rbx

0x0000000000400ef7 <+78>: jne 0x400ee1 <phase\_2+56>

0x0000000000400ef9 <+80>: mov 0x18(%rsp),%rax

0x0000000000400efe <+85>: xor %fs:0x28,%rax

0x0000000000400f07 <+94>: je 0x400f0e <phase\_2+101>

0x0000000000400f09 <+96>: callq 0x400b00 <\_\_stack\_chk\_fail@plt>

0x0000000000400f0e <+101>: add $0x28,%rsp

---Type <return> to continue, or q <return> to quit---

0x0000000000400f12 <+105>: pop %rbx

0x0000000000400f13 <+106>: pop %rbp

0x0000000000400f14 <+107>: retq

End of assembler dump.

(gdb) ni

0x0000000000400ef0 in phase\_2 ()

(gdb) disas

Dump of assembler code for function phase\_2:

0x0000000000400ea9 <+0>: push %rbp

0x0000000000400eaa <+1>: push %rbx

0x0000000000400eab <+2>: sub $0x28,%rsp

0x0000000000400eaf <+6>: mov %fs:0x28,%rax

0x0000000000400eb8 <+15>: mov %rax,0x18(%rsp)

0x0000000000400ebd <+20>: xor %eax,%eax

0x0000000000400ebf <+22>: mov %rsp,%rsi

0x0000000000400ec2 <+25>: callq 0x40145f <read\_six\_numbers>

0x0000000000400ec7 <+30>: cmpl $0x0,(%rsp)

0x0000000000400ecb <+34>: jne 0x400ed4 <phase\_2+43>

0x0000000000400ecd <+36>: cmpl $0x1,0x4(%rsp)

0x0000000000400ed2 <+41>: je 0x400ed9 <phase\_2+48>

0x0000000000400ed4 <+43>: callq 0x40143d <explode\_bomb>

0x0000000000400ed9 <+48>: mov %rsp,%rbx

0x0000000000400edc <+51>: lea 0x10(%rsp),%rbp

0x0000000000400ee1 <+56>: mov 0x4(%rbx),%eax

0x0000000000400ee4 <+59>: add (%rbx),%eax

0x0000000000400ee6 <+61>: cmp %eax,0x8(%rbx)

0x0000000000400ee9 <+64>: je 0x400ef0 <phase\_2+71>

0x0000000000400eeb <+66>: callq 0x40143d <explode\_bomb>

=> 0x0000000000400ef0 <+71>: add $0x4,%rbx

0x0000000000400ef4 <+75>: cmp %rbp,%rbx

0x0000000000400ef7 <+78>: jne 0x400ee1 <phase\_2+56>

0x0000000000400ef9 <+80>: mov 0x18(%rsp),%rax

0x0000000000400efe <+85>: xor %fs:0x28,%rax

0x0000000000400f07 <+94>: je 0x400f0e <phase\_2+101>

0x0000000000400f09 <+96>: callq 0x400b00 <\_\_stack\_chk\_fail@plt>

0x0000000000400f0e <+101>: add $0x28,%rsp

---Type <return> to continue, or q <return> to quit---

0x0000000000400f12 <+105>: pop %rbx

0x0000000000400f13 <+106>: pop %rbp

0x0000000000400f14 <+107>: retq

End of assembler dump.

(gdb) ni

0x0000000000400ef4 in phase\_2 ()

(gdb) disas

Dump of assembler code for function phase\_2:

0x0000000000400ea9 <+0>: push %rbp

0x0000000000400eaa <+1>: push %rbx

0x0000000000400eab <+2>: sub $0x28,%rsp

0x0000000000400eaf <+6>: mov %fs:0x28,%rax

0x0000000000400eb8 <+15>: mov %rax,0x18(%rsp)

0x0000000000400ebd <+20>: xor %eax,%eax

0x0000000000400ebf <+22>: mov %rsp,%rsi

0x0000000000400ec2 <+25>: callq 0x40145f <read\_six\_numbers>

0x0000000000400ec7 <+30>: cmpl $0x0,(%rsp)

0x0000000000400ecb <+34>: jne 0x400ed4 <phase\_2+43>

0x0000000000400ecd <+36>: cmpl $0x1,0x4(%rsp)

0x0000000000400ed2 <+41>: je 0x400ed9 <phase\_2+48>

0x0000000000400ed4 <+43>: callq 0x40143d <explode\_bomb>

0x0000000000400ed9 <+48>: mov %rsp,%rbx

0x0000000000400edc <+51>: lea 0x10(%rsp),%rbp

0x0000000000400ee1 <+56>: mov 0x4(%rbx),%eax

0x0000000000400ee4 <+59>: add (%rbx),%eax

0x0000000000400ee6 <+61>: cmp %eax,0x8(%rbx)

0x0000000000400ee9 <+64>: je 0x400ef0 <phase\_2+71>

0x0000000000400eeb <+66>: callq 0x40143d <explode\_bomb>

0x0000000000400ef0 <+71>: add $0x4,%rbx

=> 0x0000000000400ef4 <+75>: cmp %rbp,%rbx

0x0000000000400ef7 <+78>: jne 0x400ee1 <phase\_2+56>

0x0000000000400ef9 <+80>: mov 0x18(%rsp),%rax

0x0000000000400efe <+85>: xor %fs:0x28,%rax

0x0000000000400f07 <+94>: je 0x400f0e <phase\_2+101>

0x0000000000400f09 <+96>: callq 0x400b00 <\_\_stack\_chk\_fail@plt>

0x0000000000400f0e <+101>: add $0x28,%rsp

---Type <return> to continue, or q <return> to quit---

0x0000000000400f12 <+105>: pop %rbx

0x0000000000400f13 <+106>: pop %rbp

0x0000000000400f14 <+107>: retq

End of assembler dump.

(gdb) i r

rax 0x2 2

rbx 0x7ffffffee308 140737488282376

rcx 0x0 0

rdx 0x7ffffffee314 140737488282388

rsi 0x0 0

rdi 0x7ffffffedc70 140737488280688

rbp 0x7ffffffee310 0x7ffffffee310

rsp 0x7ffffffee300 0x7ffffffee300

r8 0x0 0

r9 0x0 0

r10 0x7fffff19ec40 140737473276992

r11 0x4025d4 4203988

r12 0x400c60 4197472

r13 0x7ffffffee420 140737488282656

r14 0x0 0

r15 0x0 0

rip 0x400ef4 0x400ef4 <phase\_2+75>

eflags 0x202 [ IF ]

cs 0x33 51

ss 0x2b 43

ds 0x0 0

es 0x0 0

fs 0x0 0

gs 0x0 0

(gdb) ni

0x0000000000400ef7 in phase\_2 ()

(gdb) disas

Dump of assembler code for function phase\_2:

0x0000000000400ea9 <+0>: push %rbp

0x0000000000400eaa <+1>: push %rbx

0x0000000000400eab <+2>: sub $0x28,%rsp

0x0000000000400eaf <+6>: mov %fs:0x28,%rax

0x0000000000400eb8 <+15>: mov %rax,0x18(%rsp)

0x0000000000400ebd <+20>: xor %eax,%eax

0x0000000000400ebf <+22>: mov %rsp,%rsi

0x0000000000400ec2 <+25>: callq 0x40145f <read\_six\_numbers>

0x0000000000400ec7 <+30>: cmpl $0x0,(%rsp)

0x0000000000400ecb <+34>: jne 0x400ed4 <phase\_2+43>

0x0000000000400ecd <+36>: cmpl $0x1,0x4(%rsp)

0x0000000000400ed2 <+41>: je 0x400ed9 <phase\_2+48>

0x0000000000400ed4 <+43>: callq 0x40143d <explode\_bomb>

0x0000000000400ed9 <+48>: mov %rsp,%rbx

0x0000000000400edc <+51>: lea 0x10(%rsp),%rbp

0x0000000000400ee1 <+56>: mov 0x4(%rbx),%eax

0x0000000000400ee4 <+59>: add (%rbx),%eax

0x0000000000400ee6 <+61>: cmp %eax,0x8(%rbx)

0x0000000000400ee9 <+64>: je 0x400ef0 <phase\_2+71>

0x0000000000400eeb <+66>: callq 0x40143d <explode\_bomb>

0x0000000000400ef0 <+71>: add $0x4,%rbx

0x0000000000400ef4 <+75>: cmp %rbp,%rbx

=> 0x0000000000400ef7 <+78>: jne 0x400ee1 <phase\_2+56>

0x0000000000400ef9 <+80>: mov 0x18(%rsp),%rax

0x0000000000400efe <+85>: xor %fs:0x28,%rax

0x0000000000400f07 <+94>: je 0x400f0e <phase\_2+101>

0x0000000000400f09 <+96>: callq 0x400b00 <\_\_stack\_chk\_fail@plt>

0x0000000000400f0e <+101>: add $0x28,%rsp

---Type <return> to continue, or q <return> to quit---

0x0000000000400f12 <+105>: pop %rbx

0x0000000000400f13 <+106>: pop %rbp

0x0000000000400f14 <+107>: retq

End of assembler dump.

(gdb) ni

0x0000000000400ee1 in phase\_2 ()

(gdb) ni

0x0000000000400ee4 in phase\_2 ()

(gdb) disas

Dump of assembler code for function phase\_2:

0x0000000000400ea9 <+0>: push %rbp

0x0000000000400eaa <+1>: push %rbx

0x0000000000400eab <+2>: sub $0x28,%rsp

0x0000000000400eaf <+6>: mov %fs:0x28,%rax

0x0000000000400eb8 <+15>: mov %rax,0x18(%rsp)

0x0000000000400ebd <+20>: xor %eax,%eax

0x0000000000400ebf <+22>: mov %rsp,%rsi

0x0000000000400ec2 <+25>: callq 0x40145f <read\_six\_numbers>

0x0000000000400ec7 <+30>: cmpl $0x0,(%rsp)

0x0000000000400ecb <+34>: jne 0x400ed4 <phase\_2+43>

0x0000000000400ecd <+36>: cmpl $0x1,0x4(%rsp)

0x0000000000400ed2 <+41>: je 0x400ed9 <phase\_2+48>

0x0000000000400ed4 <+43>: callq 0x40143d <explode\_bomb>

0x0000000000400ed9 <+48>: mov %rsp,%rbx

0x0000000000400edc <+51>: lea 0x10(%rsp),%rbp

0x0000000000400ee1 <+56>: mov 0x4(%rbx),%eax

=> 0x0000000000400ee4 <+59>: add (%rbx),%eax

0x0000000000400ee6 <+61>: cmp %eax,0x8(%rbx)

0x0000000000400ee9 <+64>: je 0x400ef0 <phase\_2+71>

0x0000000000400eeb <+66>: callq 0x40143d <explode\_bomb>

0x0000000000400ef0 <+71>: add $0x4,%rbx

0x0000000000400ef4 <+75>: cmp %rbp,%rbx

0x0000000000400ef7 <+78>: jne 0x400ee1 <phase\_2+56>

0x0000000000400ef9 <+80>: mov 0x18(%rsp),%rax

0x0000000000400efe <+85>: xor %fs:0x28,%rax

0x0000000000400f07 <+94>: je 0x400f0e <phase\_2+101>

0x0000000000400f09 <+96>: callq 0x400b00 <\_\_stack\_chk\_fail@plt>

0x0000000000400f0e <+101>: add $0x28,%rsp

---Type <return> to continue, or q <return> to quit---

0x0000000000400f12 <+105>: pop %rbx

0x0000000000400f13 <+106>: pop %rbp

0x0000000000400f14 <+107>: retq

End of assembler dump.

(gdb) ni

0x0000000000400ee6 in phase\_2 ()

(gdb) disas

Dump of assembler code for function phase\_2:

0x0000000000400ea9 <+0>: push %rbp

0x0000000000400eaa <+1>: push %rbx

0x0000000000400eab <+2>: sub $0x28,%rsp

0x0000000000400eaf <+6>: mov %fs:0x28,%rax

0x0000000000400eb8 <+15>: mov %rax,0x18(%rsp)

0x0000000000400ebd <+20>: xor %eax,%eax

0x0000000000400ebf <+22>: mov %rsp,%rsi

0x0000000000400ec2 <+25>: callq 0x40145f <read\_six\_numbers>

0x0000000000400ec7 <+30>: cmpl $0x0,(%rsp)

0x0000000000400ecb <+34>: jne 0x400ed4 <phase\_2+43>

0x0000000000400ecd <+36>: cmpl $0x1,0x4(%rsp)

0x0000000000400ed2 <+41>: je 0x400ed9 <phase\_2+48>

0x0000000000400ed4 <+43>: callq 0x40143d <explode\_bomb>

0x0000000000400ed9 <+48>: mov %rsp,%rbx

0x0000000000400edc <+51>: lea 0x10(%rsp),%rbp

0x0000000000400ee1 <+56>: mov 0x4(%rbx),%eax

0x0000000000400ee4 <+59>: add (%rbx),%eax

=> 0x0000000000400ee6 <+61>: cmp %eax,0x8(%rbx)

0x0000000000400ee9 <+64>: je 0x400ef0 <phase\_2+71>

0x0000000000400eeb <+66>: callq 0x40143d <explode\_bomb>

0x0000000000400ef0 <+71>: add $0x4,%rbx

0x0000000000400ef4 <+75>: cmp %rbp,%rbx

0x0000000000400ef7 <+78>: jne 0x400ee1 <phase\_2+56>

0x0000000000400ef9 <+80>: mov 0x18(%rsp),%rax

0x0000000000400efe <+85>: xor %fs:0x28,%rax

0x0000000000400f07 <+94>: je 0x400f0e <phase\_2+101>

0x0000000000400f09 <+96>: callq 0x400b00 <\_\_stack\_chk\_fail@plt>

0x0000000000400f0e <+101>: add $0x28,%rsp

---Type <return> to continue, or q <return> to quit---

0x0000000000400f12 <+105>: pop %rbx

0x0000000000400f13 <+106>: pop %rbp

0x0000000000400f14 <+107>: retq

End of assembler dump.

(gdb) i r

rax 0x3 3 **// The fifth input is 3**

rbx 0x7ffffffee308 140737488282376

rcx 0x0 0

rdx 0x7ffffffee314 140737488282388

rsi 0x0 0

rdi 0x7ffffffedc70 140737488280688

rbp 0x7ffffffee310 0x7ffffffee310

rsp 0x7ffffffee300 0x7ffffffee300

r8 0x0 0

r9 0x0 0

r10 0x7fffff19ec40 140737473276992

r11 0x4025d4 4203988

r12 0x400c60 4197472

r13 0x7ffffffee420 140737488282656

r14 0x0 0

r15 0x0 0

rip 0x400ee6 0x400ee6 <phase\_2+61>

eflags 0x206 [ PF IF ]

cs 0x33 51

ss 0x2b 43

ds 0x0 0

es 0x0 0

fs 0x0 0

gs 0x0 0

(gdb) r

The program being debugged has been started already.

Start it from the beginning? (y or n) y

Starting program: /mnt/d/one drive/OneDrive/Desktop/12190094/Assignment 1/bomb001/bomb

Welcome to my fiendish little bomb. You have 6 phases with

which to blow yourself up. Have a nice day!

The moon unit will be divided into two divisions.

Phase 1 defused. How about the next one?

0 1 1 2 3 6 **// test input for integer 6**

Breakpoint 1, 0x0000000000400ea9 in phase\_2 ()

(gdb) disas

Dump of assembler code for function phase\_2:

=> 0x0000000000400ea9 <+0>: push %rbp

0x0000000000400eaa <+1>: push %rbx

0x0000000000400eab <+2>: sub $0x28,%rsp

0x0000000000400eaf <+6>: mov %fs:0x28,%rax

0x0000000000400eb8 <+15>: mov %rax,0x18(%rsp)

0x0000000000400ebd <+20>: xor %eax,%eax

0x0000000000400ebf <+22>: mov %rsp,%rsi

0x0000000000400ec2 <+25>: callq 0x40145f <read\_six\_numbers>

0x0000000000400ec7 <+30>: cmpl $0x0,(%rsp)

0x0000000000400ecb <+34>: jne 0x400ed4 <phase\_2+43>

0x0000000000400ecd <+36>: cmpl $0x1,0x4(%rsp)

0x0000000000400ed2 <+41>: je 0x400ed9 <phase\_2+48>

0x0000000000400ed4 <+43>: callq 0x40143d <explode\_bomb>

0x0000000000400ed9 <+48>: mov %rsp,%rbx

0x0000000000400edc <+51>: lea 0x10(%rsp),%rbp

0x0000000000400ee1 <+56>: mov 0x4(%rbx),%eax

0x0000000000400ee4 <+59>: add (%rbx),%eax

0x0000000000400ee6 <+61>: cmp %eax,0x8(%rbx)

0x0000000000400ee9 <+64>: je 0x400ef0 <phase\_2+71>

0x0000000000400eeb <+66>: callq 0x40143d <explode\_bomb>

0x0000000000400ef0 <+71>: add $0x4,%rbx

0x0000000000400ef4 <+75>: cmp %rbp,%rbx

0x0000000000400ef7 <+78>: jne 0x400ee1 <phase\_2+56>

0x0000000000400ef9 <+80>: mov 0x18(%rsp),%rax

0x0000000000400efe <+85>: xor %fs:0x28,%rax

0x0000000000400f07 <+94>: je 0x400f0e <phase\_2+101>

0x0000000000400f09 <+96>: callq 0x400b00 <\_\_stack\_chk\_fail@plt>

0x0000000000400f0e <+101>: add $0x28,%rsp

---Type <return> to continue, or q <return> to quit---

0x0000000000400f12 <+105>: pop %rbx

0x0000000000400f13 <+106>: pop %rbp

0x0000000000400f14 <+107>: retq

End of assembler dump.

(gdb) until \*0x0000000000400ee6

0x0000000000400ee6 in phase\_2 ()

(gdb) disas

Dump of assembler code for function phase\_2:

0x0000000000400ea9 <+0>: push %rbp

0x0000000000400eaa <+1>: push %rbx

0x0000000000400eab <+2>: sub $0x28,%rsp

0x0000000000400eaf <+6>: mov %fs:0x28,%rax

0x0000000000400eb8 <+15>: mov %rax,0x18(%rsp)

0x0000000000400ebd <+20>: xor %eax,%eax

0x0000000000400ebf <+22>: mov %rsp,%rsi

0x0000000000400ec2 <+25>: callq 0x40145f <read\_six\_numbers>

0x0000000000400ec7 <+30>: cmpl $0x0,(%rsp)

0x0000000000400ecb <+34>: jne 0x400ed4 <phase\_2+43>

0x0000000000400ecd <+36>: cmpl $0x1,0x4(%rsp)

0x0000000000400ed2 <+41>: je 0x400ed9 <phase\_2+48>

0x0000000000400ed4 <+43>: callq 0x40143d <explode\_bomb>

0x0000000000400ed9 <+48>: mov %rsp,%rbx

0x0000000000400edc <+51>: lea 0x10(%rsp),%rbp

0x0000000000400ee1 <+56>: mov 0x4(%rbx),%eax

0x0000000000400ee4 <+59>: add (%rbx),%eax

=> 0x0000000000400ee6 <+61>: cmp %eax,0x8(%rbx)

0x0000000000400ee9 <+64>: je 0x400ef0 <phase\_2+71>

0x0000000000400eeb <+66>: callq 0x40143d <explode\_bomb>

0x0000000000400ef0 <+71>: add $0x4,%rbx

0x0000000000400ef4 <+75>: cmp %rbp,%rbx

0x0000000000400ef7 <+78>: jne 0x400ee1 <phase\_2+56>

0x0000000000400ef9 <+80>: mov 0x18(%rsp),%rax

0x0000000000400efe <+85>: xor %fs:0x28,%rax

0x0000000000400f07 <+94>: je 0x400f0e <phase\_2+101>

0x0000000000400f09 <+96>: callq 0x400b00 <\_\_stack\_chk\_fail@plt>

0x0000000000400f0e <+101>: add $0x28,%rsp

---Type <return> to continue, or q <return> to quit---

0x0000000000400f12 <+105>: pop %rbx

0x0000000000400f13 <+106>: pop %rbp

0x0000000000400f14 <+107>: retq

End of assembler dump.

(gdb) i r

rax 0x1 1

rbx 0x7ffffffee300 140737488282368

rcx 0x0 0

rdx 0x7ffffffee314 140737488282388

rsi 0x0 0

rdi 0x7ffffffedc70 140737488280688

rbp 0x7ffffffee310 0x7ffffffee310

rsp 0x7ffffffee300 0x7ffffffee300

r8 0x0 0

r9 0x0 0

r10 0x7fffff19ec40 140737473276992

r11 0x4025d4 4203988

r12 0x400c60 4197472

r13 0x7ffffffee420 140737488282656

r14 0x0 0

r15 0x0 0

rip 0x400ee6 0x400ee6 <phase\_2+61>

eflags 0x202 [ IF ]

cs 0x33 51

ss 0x2b 43

ds 0x0 0

es 0x0 0

fs 0x0 0

gs 0x0 0

(gdb) ni

0x0000000000400ee9 in phase\_2 ()

(gdb) disas

Dump of assembler code for function phase\_2:

0x0000000000400ea9 <+0>: push %rbp

0x0000000000400eaa <+1>: push %rbx

0x0000000000400eab <+2>: sub $0x28,%rsp

0x0000000000400eaf <+6>: mov %fs:0x28,%rax

0x0000000000400eb8 <+15>: mov %rax,0x18(%rsp)

0x0000000000400ebd <+20>: xor %eax,%eax

0x0000000000400ebf <+22>: mov %rsp,%rsi

0x0000000000400ec2 <+25>: callq 0x40145f <read\_six\_numbers>

0x0000000000400ec7 <+30>: cmpl $0x0,(%rsp)

0x0000000000400ecb <+34>: jne 0x400ed4 <phase\_2+43>

0x0000000000400ecd <+36>: cmpl $0x1,0x4(%rsp)

0x0000000000400ed2 <+41>: je 0x400ed9 <phase\_2+48>

0x0000000000400ed4 <+43>: callq 0x40143d <explode\_bomb>

0x0000000000400ed9 <+48>: mov %rsp,%rbx

0x0000000000400edc <+51>: lea 0x10(%rsp),%rbp

0x0000000000400ee1 <+56>: mov 0x4(%rbx),%eax

0x0000000000400ee4 <+59>: add (%rbx),%eax

0x0000000000400ee6 <+61>: cmp %eax,0x8(%rbx)

=> 0x0000000000400ee9 <+64>: je 0x400ef0 <phase\_2+71>

0x0000000000400eeb <+66>: callq 0x40143d <explode\_bomb>

0x0000000000400ef0 <+71>: add $0x4,%rbx

0x0000000000400ef4 <+75>: cmp %rbp,%rbx

0x0000000000400ef7 <+78>: jne 0x400ee1 <phase\_2+56>

0x0000000000400ef9 <+80>: mov 0x18(%rsp),%rax

0x0000000000400efe <+85>: xor %fs:0x28,%rax

0x0000000000400f07 <+94>: je 0x400f0e <phase\_2+101>

0x0000000000400f09 <+96>: callq 0x400b00 <\_\_stack\_chk\_fail@plt>

0x0000000000400f0e <+101>: add $0x28,%rsp

---Type <return> to continue, or q <return> to quit---

0x0000000000400f12 <+105>: pop %rbx

0x0000000000400f13 <+106>: pop %rbp

0x0000000000400f14 <+107>: retq

End of assembler dump.

(gdb) ni

0x0000000000400ef0 in phase\_2 ()

(gdb) disas

Dump of assembler code for function phase\_2:

0x0000000000400ea9 <+0>: push %rbp

0x0000000000400eaa <+1>: push %rbx

0x0000000000400eab <+2>: sub $0x28,%rsp

0x0000000000400eaf <+6>: mov %fs:0x28,%rax

0x0000000000400eb8 <+15>: mov %rax,0x18(%rsp)

0x0000000000400ebd <+20>: xor %eax,%eax

0x0000000000400ebf <+22>: mov %rsp,%rsi

0x0000000000400ec2 <+25>: callq 0x40145f <read\_six\_numbers>

0x0000000000400ec7 <+30>: cmpl $0x0,(%rsp)

0x0000000000400ecb <+34>: jne 0x400ed4 <phase\_2+43>

0x0000000000400ecd <+36>: cmpl $0x1,0x4(%rsp)

0x0000000000400ed2 <+41>: je 0x400ed9 <phase\_2+48>

0x0000000000400ed4 <+43>: callq 0x40143d <explode\_bomb>

0x0000000000400ed9 <+48>: mov %rsp,%rbx

0x0000000000400edc <+51>: lea 0x10(%rsp),%rbp

0x0000000000400ee1 <+56>: mov 0x4(%rbx),%eax

0x0000000000400ee4 <+59>: add (%rbx),%eax

0x0000000000400ee6 <+61>: cmp %eax,0x8(%rbx)

0x0000000000400ee9 <+64>: je 0x400ef0 <phase\_2+71>

0x0000000000400eeb <+66>: callq 0x40143d <explode\_bomb>

=> 0x0000000000400ef0 <+71>: add $0x4,%rbx

0x0000000000400ef4 <+75>: cmp %rbp,%rbx

0x0000000000400ef7 <+78>: jne 0x400ee1 <phase\_2+56>

0x0000000000400ef9 <+80>: mov 0x18(%rsp),%rax

0x0000000000400efe <+85>: xor %fs:0x28,%rax

0x0000000000400f07 <+94>: je 0x400f0e <phase\_2+101>

0x0000000000400f09 <+96>: callq 0x400b00 <\_\_stack\_chk\_fail@plt>

0x0000000000400f0e <+101>: add $0x28,%rsp

---Type <return> to continue, or q <return> to quit---

0x0000000000400f12 <+105>: pop %rbx

0x0000000000400f13 <+106>: pop %rbp

0x0000000000400f14 <+107>: retq

End of assembler dump.

(gdb) ni

0x0000000000400ef4 in phase\_2 ()

(gdb) disas

Dump of assembler code for function phase\_2:

0x0000000000400ea9 <+0>: push %rbp

0x0000000000400eaa <+1>: push %rbx

0x0000000000400eab <+2>: sub $0x28,%rsp

0x0000000000400eaf <+6>: mov %fs:0x28,%rax

0x0000000000400eb8 <+15>: mov %rax,0x18(%rsp)

0x0000000000400ebd <+20>: xor %eax,%eax

0x0000000000400ebf <+22>: mov %rsp,%rsi

0x0000000000400ec2 <+25>: callq 0x40145f <read\_six\_numbers>

0x0000000000400ec7 <+30>: cmpl $0x0,(%rsp)

0x0000000000400ecb <+34>: jne 0x400ed4 <phase\_2+43>

0x0000000000400ecd <+36>: cmpl $0x1,0x4(%rsp)

0x0000000000400ed2 <+41>: je 0x400ed9 <phase\_2+48>

0x0000000000400ed4 <+43>: callq 0x40143d <explode\_bomb>

0x0000000000400ed9 <+48>: mov %rsp,%rbx

0x0000000000400edc <+51>: lea 0x10(%rsp),%rbp

0x0000000000400ee1 <+56>: mov 0x4(%rbx),%eax

0x0000000000400ee4 <+59>: add (%rbx),%eax

0x0000000000400ee6 <+61>: cmp %eax,0x8(%rbx)

0x0000000000400ee9 <+64>: je 0x400ef0 <phase\_2+71>

0x0000000000400eeb <+66>: callq 0x40143d <explode\_bomb>

0x0000000000400ef0 <+71>: add $0x4,%rbx

=> 0x0000000000400ef4 <+75>: cmp %rbp,%rbx

0x0000000000400ef7 <+78>: jne 0x400ee1 <phase\_2+56>

0x0000000000400ef9 <+80>: mov 0x18(%rsp),%rax

0x0000000000400efe <+85>: xor %fs:0x28,%rax

0x0000000000400f07 <+94>: je 0x400f0e <phase\_2+101>

0x0000000000400f09 <+96>: callq 0x400b00 <\_\_stack\_chk\_fail@plt>

0x0000000000400f0e <+101>: add $0x28,%rsp

---Type <return> to continue, or q <return> to quit---

0x0000000000400f12 <+105>: pop %rbx

0x0000000000400f13 <+106>: pop %rbp

0x0000000000400f14 <+107>: retq

End of assembler dump.

(gdb) i r

rax 0x1 1

rbx 0x7ffffffee304 140737488282372

rcx 0x0 0

rdx 0x7ffffffee314 140737488282388

rsi 0x0 0

rdi 0x7ffffffedc70 140737488280688

rbp 0x7ffffffee310 0x7ffffffee310

rsp 0x7ffffffee300 0x7ffffffee300

r8 0x0 0

r9 0x0 0

r10 0x7fffff19ec40 140737473276992

r11 0x4025d4 4203988

r12 0x400c60 4197472

r13 0x7ffffffee420 140737488282656

r14 0x0 0

r15 0x0 0

rip 0x400ef4 0x400ef4 <phase\_2+75>

eflags 0x202 [ IF ]

cs 0x33 51

ss 0x2b 43

ds 0x0 0

es 0x0 0

fs 0x0 0

gs 0x0 0

(gdb) ni

0x0000000000400ef7 in phase\_2 ()

(gdb) disas

Dump of assembler code for function phase\_2:

0x0000000000400ea9 <+0>: push %rbp

0x0000000000400eaa <+1>: push %rbx

0x0000000000400eab <+2>: sub $0x28,%rsp

0x0000000000400eaf <+6>: mov %fs:0x28,%rax

0x0000000000400eb8 <+15>: mov %rax,0x18(%rsp)

0x0000000000400ebd <+20>: xor %eax,%eax

0x0000000000400ebf <+22>: mov %rsp,%rsi

0x0000000000400ec2 <+25>: callq 0x40145f <read\_six\_numbers>

0x0000000000400ec7 <+30>: cmpl $0x0,(%rsp)

0x0000000000400ecb <+34>: jne 0x400ed4 <phase\_2+43>

0x0000000000400ecd <+36>: cmpl $0x1,0x4(%rsp)

0x0000000000400ed2 <+41>: je 0x400ed9 <phase\_2+48>

0x0000000000400ed4 <+43>: callq 0x40143d <explode\_bomb>

0x0000000000400ed9 <+48>: mov %rsp,%rbx

0x0000000000400edc <+51>: lea 0x10(%rsp),%rbp

0x0000000000400ee1 <+56>: mov 0x4(%rbx),%eax

0x0000000000400ee4 <+59>: add (%rbx),%eax

0x0000000000400ee6 <+61>: cmp %eax,0x8(%rbx)

0x0000000000400ee9 <+64>: je 0x400ef0 <phase\_2+71>

0x0000000000400eeb <+66>: callq 0x40143d <explode\_bomb>

0x0000000000400ef0 <+71>: add $0x4,%rbx

0x0000000000400ef4 <+75>: cmp %rbp,%rbx

=> 0x0000000000400ef7 <+78>: jne 0x400ee1 <phase\_2+56>

0x0000000000400ef9 <+80>: mov 0x18(%rsp),%rax

0x0000000000400efe <+85>: xor %fs:0x28,%rax

0x0000000000400f07 <+94>: je 0x400f0e <phase\_2+101>

0x0000000000400f09 <+96>: callq 0x400b00 <\_\_stack\_chk\_fail@plt>

0x0000000000400f0e <+101>: add $0x28,%rsp

---Type <return> to continue, or q <return> to quit---

0x0000000000400f12 <+105>: pop %rbx

0x0000000000400f13 <+106>: pop %rbp

0x0000000000400f14 <+107>: retq

End of assembler dump.

(gdb) ni

0x0000000000400ee1 in phase\_2 ()

(gdb) disas

Dump of assembler code for function phase\_2:

0x0000000000400ea9 <+0>: push %rbp

0x0000000000400eaa <+1>: push %rbx

0x0000000000400eab <+2>: sub $0x28,%rsp

0x0000000000400eaf <+6>: mov %fs:0x28,%rax

0x0000000000400eb8 <+15>: mov %rax,0x18(%rsp)

0x0000000000400ebd <+20>: xor %eax,%eax

0x0000000000400ebf <+22>: mov %rsp,%rsi

0x0000000000400ec2 <+25>: callq 0x40145f <read\_six\_numbers>

0x0000000000400ec7 <+30>: cmpl $0x0,(%rsp)

0x0000000000400ecb <+34>: jne 0x400ed4 <phase\_2+43>

0x0000000000400ecd <+36>: cmpl $0x1,0x4(%rsp)

0x0000000000400ed2 <+41>: je 0x400ed9 <phase\_2+48>

0x0000000000400ed4 <+43>: callq 0x40143d <explode\_bomb>

0x0000000000400ed9 <+48>: mov %rsp,%rbx

0x0000000000400edc <+51>: lea 0x10(%rsp),%rbp

=> 0x0000000000400ee1 <+56>: mov 0x4(%rbx),%eax

0x0000000000400ee4 <+59>: add (%rbx),%eax

0x0000000000400ee6 <+61>: cmp %eax,0x8(%rbx)

0x0000000000400ee9 <+64>: je 0x400ef0 <phase\_2+71>

0x0000000000400eeb <+66>: callq 0x40143d <explode\_bomb>

0x0000000000400ef0 <+71>: add $0x4,%rbx

0x0000000000400ef4 <+75>: cmp %rbp,%rbx

0x0000000000400ef7 <+78>: jne 0x400ee1 <phase\_2+56>

0x0000000000400ef9 <+80>: mov 0x18(%rsp),%rax

0x0000000000400efe <+85>: xor %fs:0x28,%rax

0x0000000000400f07 <+94>: je 0x400f0e <phase\_2+101>

0x0000000000400f09 <+96>: callq 0x400b00 <\_\_stack\_chk\_fail@plt>

0x0000000000400f0e <+101>: add $0x28,%rsp

---Type <return> to continue, or q <return> to quit---

0x0000000000400f12 <+105>: pop %rbx

0x0000000000400f13 <+106>: pop %rbp

0x0000000000400f14 <+107>: retq

End of assembler dump.

(gdb) ni

0x0000000000400ee4 in phase\_2 ()

(gdb) ni

0x0000000000400ee6 in phase\_2 ()

(gdb) disas

Dump of assembler code for function phase\_2:

0x0000000000400ea9 <+0>: push %rbp

0x0000000000400eaa <+1>: push %rbx

0x0000000000400eab <+2>: sub $0x28,%rsp

0x0000000000400eaf <+6>: mov %fs:0x28,%rax

0x0000000000400eb8 <+15>: mov %rax,0x18(%rsp)

0x0000000000400ebd <+20>: xor %eax,%eax

0x0000000000400ebf <+22>: mov %rsp,%rsi

0x0000000000400ec2 <+25>: callq 0x40145f <read\_six\_numbers>

0x0000000000400ec7 <+30>: cmpl $0x0,(%rsp)

0x0000000000400ecb <+34>: jne 0x400ed4 <phase\_2+43>

0x0000000000400ecd <+36>: cmpl $0x1,0x4(%rsp)

0x0000000000400ed2 <+41>: je 0x400ed9 <phase\_2+48>

0x0000000000400ed4 <+43>: callq 0x40143d <explode\_bomb>

0x0000000000400ed9 <+48>: mov %rsp,%rbx

0x0000000000400edc <+51>: lea 0x10(%rsp),%rbp

0x0000000000400ee1 <+56>: mov 0x4(%rbx),%eax

0x0000000000400ee4 <+59>: add (%rbx),%eax

=> 0x0000000000400ee6 <+61>: cmp %eax,0x8(%rbx)

0x0000000000400ee9 <+64>: je 0x400ef0 <phase\_2+71>

0x0000000000400eeb <+66>: callq 0x40143d <explode\_bomb>

0x0000000000400ef0 <+71>: add $0x4,%rbx

0x0000000000400ef4 <+75>: cmp %rbp,%rbx

0x0000000000400ef7 <+78>: jne 0x400ee1 <phase\_2+56>

0x0000000000400ef9 <+80>: mov 0x18(%rsp),%rax

0x0000000000400efe <+85>: xor %fs:0x28,%rax

0x0000000000400f07 <+94>: je 0x400f0e <phase\_2+101>

0x0000000000400f09 <+96>: callq 0x400b00 <\_\_stack\_chk\_fail@plt>

0x0000000000400f0e <+101>: add $0x28,%rsp

---Type <return> to continue, or q <return> to quit---

0x0000000000400f12 <+105>: pop %rbx

0x0000000000400f13 <+106>: pop %rbp

0x0000000000400f14 <+107>: retq

End of assembler dump.

(gdb) i r

rax 0x2 2

rbx 0x7ffffffee304 140737488282372

rcx 0x0 0

rdx 0x7ffffffee314 140737488282388

rsi 0x0 0

rdi 0x7ffffffedc70 140737488280688

rbp 0x7ffffffee310 0x7ffffffee310

rsp 0x7ffffffee300 0x7ffffffee300

r8 0x0 0

r9 0x0 0

r10 0x7fffff19ec40 140737473276992

r11 0x4025d4 4203988

r12 0x400c60 4197472

r13 0x7ffffffee420 140737488282656

r14 0x0 0

r15 0x0 0

rip 0x400ee6 0x400ee6 <phase\_2+61>

eflags 0x202 [ IF ]

cs 0x33 51

ss 0x2b 43

ds 0x0 0

es 0x0 0

fs 0x0 0

gs 0x0 0

(gdb) ni

0x0000000000400ee9 in phase\_2 ()

(gdb) disas

Dump of assembler code for function phase\_2:

0x0000000000400ea9 <+0>: push %rbp

0x0000000000400eaa <+1>: push %rbx

0x0000000000400eab <+2>: sub $0x28,%rsp

0x0000000000400eaf <+6>: mov %fs:0x28,%rax

0x0000000000400eb8 <+15>: mov %rax,0x18(%rsp)

0x0000000000400ebd <+20>: xor %eax,%eax

0x0000000000400ebf <+22>: mov %rsp,%rsi

0x0000000000400ec2 <+25>: callq 0x40145f <read\_six\_numbers>

0x0000000000400ec7 <+30>: cmpl $0x0,(%rsp)

0x0000000000400ecb <+34>: jne 0x400ed4 <phase\_2+43>

0x0000000000400ecd <+36>: cmpl $0x1,0x4(%rsp)

0x0000000000400ed2 <+41>: je 0x400ed9 <phase\_2+48>

0x0000000000400ed4 <+43>: callq 0x40143d <explode\_bomb>

0x0000000000400ed9 <+48>: mov %rsp,%rbx

0x0000000000400edc <+51>: lea 0x10(%rsp),%rbp

0x0000000000400ee1 <+56>: mov 0x4(%rbx),%eax

0x0000000000400ee4 <+59>: add (%rbx),%eax

0x0000000000400ee6 <+61>: cmp %eax,0x8(%rbx)

=> 0x0000000000400ee9 <+64>: je 0x400ef0 <phase\_2+71>

0x0000000000400eeb <+66>: callq 0x40143d <explode\_bomb>

0x0000000000400ef0 <+71>: add $0x4,%rbx

0x0000000000400ef4 <+75>: cmp %rbp,%rbx

0x0000000000400ef7 <+78>: jne 0x400ee1 <phase\_2+56>

0x0000000000400ef9 <+80>: mov 0x18(%rsp),%rax

0x0000000000400efe <+85>: xor %fs:0x28,%rax

0x0000000000400f07 <+94>: je 0x400f0e <phase\_2+101>

0x0000000000400f09 <+96>: callq 0x400b00 <\_\_stack\_chk\_fail@plt>

0x0000000000400f0e <+101>: add $0x28,%rsp

---Type <return> to continue, or q <return> to quit---

0x0000000000400f12 <+105>: pop %rbx

0x0000000000400f13 <+106>: pop %rbp

0x0000000000400f14 <+107>: retq

End of assembler dump.

(gdb) ni

0x0000000000400ef0 in phase\_2 ()

(gdb) disas

Dump of assembler code for function phase\_2:

0x0000000000400ea9 <+0>: push %rbp

0x0000000000400eaa <+1>: push %rbx

0x0000000000400eab <+2>: sub $0x28,%rsp

0x0000000000400eaf <+6>: mov %fs:0x28,%rax

0x0000000000400eb8 <+15>: mov %rax,0x18(%rsp)

0x0000000000400ebd <+20>: xor %eax,%eax

0x0000000000400ebf <+22>: mov %rsp,%rsi

0x0000000000400ec2 <+25>: callq 0x40145f <read\_six\_numbers>

0x0000000000400ec7 <+30>: cmpl $0x0,(%rsp)

0x0000000000400ecb <+34>: jne 0x400ed4 <phase\_2+43>

0x0000000000400ecd <+36>: cmpl $0x1,0x4(%rsp)

0x0000000000400ed2 <+41>: je 0x400ed9 <phase\_2+48>

0x0000000000400ed4 <+43>: callq 0x40143d <explode\_bomb>

0x0000000000400ed9 <+48>: mov %rsp,%rbx

0x0000000000400edc <+51>: lea 0x10(%rsp),%rbp

0x0000000000400ee1 <+56>: mov 0x4(%rbx),%eax

0x0000000000400ee4 <+59>: add (%rbx),%eax

0x0000000000400ee6 <+61>: cmp %eax,0x8(%rbx)

0x0000000000400ee9 <+64>: je 0x400ef0 <phase\_2+71>

0x0000000000400eeb <+66>: callq 0x40143d <explode\_bomb>

=> 0x0000000000400ef0 <+71>: add $0x4,%rbx

0x0000000000400ef4 <+75>: cmp %rbp,%rbx

0x0000000000400ef7 <+78>: jne 0x400ee1 <phase\_2+56>

0x0000000000400ef9 <+80>: mov 0x18(%rsp),%rax

0x0000000000400efe <+85>: xor %fs:0x28,%rax

0x0000000000400f07 <+94>: je 0x400f0e <phase\_2+101>

0x0000000000400f09 <+96>: callq 0x400b00 <\_\_stack\_chk\_fail@plt>

0x0000000000400f0e <+101>: add $0x28,%rsp

---Type <return> to continue, or q <return> to quit---

0x0000000000400f12 <+105>: pop %rbx

0x0000000000400f13 <+106>: pop %rbp

0x0000000000400f14 <+107>: retq

End of assembler dump.

(gdb) ni

0x0000000000400ef4 in phase\_2 ()

(gdb) disas

Dump of assembler code for function phase\_2:

0x0000000000400ea9 <+0>: push %rbp

0x0000000000400eaa <+1>: push %rbx

0x0000000000400eab <+2>: sub $0x28,%rsp

0x0000000000400eaf <+6>: mov %fs:0x28,%rax

0x0000000000400eb8 <+15>: mov %rax,0x18(%rsp)

0x0000000000400ebd <+20>: xor %eax,%eax

0x0000000000400ebf <+22>: mov %rsp,%rsi

0x0000000000400ec2 <+25>: callq 0x40145f <read\_six\_numbers>

0x0000000000400ec7 <+30>: cmpl $0x0,(%rsp)

0x0000000000400ecb <+34>: jne 0x400ed4 <phase\_2+43>

0x0000000000400ecd <+36>: cmpl $0x1,0x4(%rsp)

0x0000000000400ed2 <+41>: je 0x400ed9 <phase\_2+48>

0x0000000000400ed4 <+43>: callq 0x40143d <explode\_bomb>

0x0000000000400ed9 <+48>: mov %rsp,%rbx

0x0000000000400edc <+51>: lea 0x10(%rsp),%rbp

0x0000000000400ee1 <+56>: mov 0x4(%rbx),%eax

0x0000000000400ee4 <+59>: add (%rbx),%eax

0x0000000000400ee6 <+61>: cmp %eax,0x8(%rbx)

0x0000000000400ee9 <+64>: je 0x400ef0 <phase\_2+71>

0x0000000000400eeb <+66>: callq 0x40143d <explode\_bomb>

0x0000000000400ef0 <+71>: add $0x4,%rbx

=> 0x0000000000400ef4 <+75>: cmp %rbp,%rbx

0x0000000000400ef7 <+78>: jne 0x400ee1 <phase\_2+56>

0x0000000000400ef9 <+80>: mov 0x18(%rsp),%rax

0x0000000000400efe <+85>: xor %fs:0x28,%rax

0x0000000000400f07 <+94>: je 0x400f0e <phase\_2+101>

0x0000000000400f09 <+96>: callq 0x400b00 <\_\_stack\_chk\_fail@plt>

0x0000000000400f0e <+101>: add $0x28,%rsp

---Type <return> to continue, or q <return> to quit---

0x0000000000400f12 <+105>: pop %rbx

0x0000000000400f13 <+106>: pop %rbp

0x0000000000400f14 <+107>: retq

End of assembler dump.

(gdb) i r

rax 0x2 2

rbx 0x7ffffffee308 140737488282376

rcx 0x0 0

rdx 0x7ffffffee314 140737488282388

rsi 0x0 0

rdi 0x7ffffffedc70 140737488280688

rbp 0x7ffffffee310 0x7ffffffee310

rsp 0x7ffffffee300 0x7ffffffee300

r8 0x0 0

r9 0x0 0

r10 0x7fffff19ec40 140737473276992

r11 0x4025d4 4203988

r12 0x400c60 4197472

r13 0x7ffffffee420 140737488282656

r14 0x0 0

r15 0x0 0

rip 0x400ef4 0x400ef4 <phase\_2+75>

eflags 0x202 [ IF ]

cs 0x33 51

ss 0x2b 43

ds 0x0 0

es 0x0 0

fs 0x0 0

gs 0x0 0

(gdb) ni

0x0000000000400ef7 in phase\_2 ()

(gdb) disas

Dump of assembler code for function phase\_2:

0x0000000000400ea9 <+0>: push %rbp

0x0000000000400eaa <+1>: push %rbx

0x0000000000400eab <+2>: sub $0x28,%rsp

0x0000000000400eaf <+6>: mov %fs:0x28,%rax

0x0000000000400eb8 <+15>: mov %rax,0x18(%rsp)

0x0000000000400ebd <+20>: xor %eax,%eax

0x0000000000400ebf <+22>: mov %rsp,%rsi

0x0000000000400ec2 <+25>: callq 0x40145f <read\_six\_numbers>

0x0000000000400ec7 <+30>: cmpl $0x0,(%rsp)

0x0000000000400ecb <+34>: jne 0x400ed4 <phase\_2+43>

0x0000000000400ecd <+36>: cmpl $0x1,0x4(%rsp)

0x0000000000400ed2 <+41>: je 0x400ed9 <phase\_2+48>

0x0000000000400ed4 <+43>: callq 0x40143d <explode\_bomb>

0x0000000000400ed9 <+48>: mov %rsp,%rbx

0x0000000000400edc <+51>: lea 0x10(%rsp),%rbp

0x0000000000400ee1 <+56>: mov 0x4(%rbx),%eax

0x0000000000400ee4 <+59>: add (%rbx),%eax

0x0000000000400ee6 <+61>: cmp %eax,0x8(%rbx)

0x0000000000400ee9 <+64>: je 0x400ef0 <phase\_2+71>

0x0000000000400eeb <+66>: callq 0x40143d <explode\_bomb>

0x0000000000400ef0 <+71>: add $0x4,%rbx

0x0000000000400ef4 <+75>: cmp %rbp,%rbx

=> 0x0000000000400ef7 <+78>: jne 0x400ee1 <phase\_2+56>

0x0000000000400ef9 <+80>: mov 0x18(%rsp),%rax

0x0000000000400efe <+85>: xor %fs:0x28,%rax

0x0000000000400f07 <+94>: je 0x400f0e <phase\_2+101>

0x0000000000400f09 <+96>: callq 0x400b00 <\_\_stack\_chk\_fail@plt>

0x0000000000400f0e <+101>: add $0x28,%rsp

---Type <return> to continue, or q <return> to quit---

0x0000000000400f12 <+105>: pop %rbx

0x0000000000400f13 <+106>: pop %rbp

0x0000000000400f14 <+107>: retq

End of assembler dump.

(gdb) ni

0x0000000000400ee1 in phase\_2 ()

(gdb) disas

Dump of assembler code for function phase\_2:

0x0000000000400ea9 <+0>: push %rbp

0x0000000000400eaa <+1>: push %rbx

0x0000000000400eab <+2>: sub $0x28,%rsp

0x0000000000400eaf <+6>: mov %fs:0x28,%rax

0x0000000000400eb8 <+15>: mov %rax,0x18(%rsp)

0x0000000000400ebd <+20>: xor %eax,%eax

0x0000000000400ebf <+22>: mov %rsp,%rsi

0x0000000000400ec2 <+25>: callq 0x40145f <read\_six\_numbers>

0x0000000000400ec7 <+30>: cmpl $0x0,(%rsp)

0x0000000000400ecb <+34>: jne 0x400ed4 <phase\_2+43>

0x0000000000400ecd <+36>: cmpl $0x1,0x4(%rsp)

0x0000000000400ed2 <+41>: je 0x400ed9 <phase\_2+48>

0x0000000000400ed4 <+43>: callq 0x40143d <explode\_bomb>

0x0000000000400ed9 <+48>: mov %rsp,%rbx

0x0000000000400edc <+51>: lea 0x10(%rsp),%rbp

=> 0x0000000000400ee1 <+56>: mov 0x4(%rbx),%eax

0x0000000000400ee4 <+59>: add (%rbx),%eax

0x0000000000400ee6 <+61>: cmp %eax,0x8(%rbx)

0x0000000000400ee9 <+64>: je 0x400ef0 <phase\_2+71>

0x0000000000400eeb <+66>: callq 0x40143d <explode\_bomb>

0x0000000000400ef0 <+71>: add $0x4,%rbx

0x0000000000400ef4 <+75>: cmp %rbp,%rbx

0x0000000000400ef7 <+78>: jne 0x400ee1 <phase\_2+56>

0x0000000000400ef9 <+80>: mov 0x18(%rsp),%rax

0x0000000000400efe <+85>: xor %fs:0x28,%rax

0x0000000000400f07 <+94>: je 0x400f0e <phase\_2+101>

0x0000000000400f09 <+96>: callq 0x400b00 <\_\_stack\_chk\_fail@plt>

0x0000000000400f0e <+101>: add $0x28,%rsp

---Type <return> to continue, or q <return> to quit---

0x0000000000400f12 <+105>: pop %rbx

0x0000000000400f13 <+106>: pop %rbp

0x0000000000400f14 <+107>: retq

End of assembler dump.

(gdb) ni

0x0000000000400ee4 in phase\_2 ()

(gdb) disas

Dump of assembler code for function phase\_2:

0x0000000000400ea9 <+0>: push %rbp

0x0000000000400eaa <+1>: push %rbx

0x0000000000400eab <+2>: sub $0x28,%rsp

0x0000000000400eaf <+6>: mov %fs:0x28,%rax

0x0000000000400eb8 <+15>: mov %rax,0x18(%rsp)

0x0000000000400ebd <+20>: xor %eax,%eax

0x0000000000400ebf <+22>: mov %rsp,%rsi

0x0000000000400ec2 <+25>: callq 0x40145f <read\_six\_numbers>

0x0000000000400ec7 <+30>: cmpl $0x0,(%rsp)

0x0000000000400ecb <+34>: jne 0x400ed4 <phase\_2+43>

0x0000000000400ecd <+36>: cmpl $0x1,0x4(%rsp)

0x0000000000400ed2 <+41>: je 0x400ed9 <phase\_2+48>

0x0000000000400ed4 <+43>: callq 0x40143d <explode\_bomb>

0x0000000000400ed9 <+48>: mov %rsp,%rbx

0x0000000000400edc <+51>: lea 0x10(%rsp),%rbp

0x0000000000400ee1 <+56>: mov 0x4(%rbx),%eax

=> 0x0000000000400ee4 <+59>: add (%rbx),%eax

0x0000000000400ee6 <+61>: cmp %eax,0x8(%rbx)

0x0000000000400ee9 <+64>: je 0x400ef0 <phase\_2+71>

0x0000000000400eeb <+66>: callq 0x40143d <explode\_bomb>

0x0000000000400ef0 <+71>: add $0x4,%rbx

0x0000000000400ef4 <+75>: cmp %rbp,%rbx

0x0000000000400ef7 <+78>: jne 0x400ee1 <phase\_2+56>

0x0000000000400ef9 <+80>: mov 0x18(%rsp),%rax

0x0000000000400efe <+85>: xor %fs:0x28,%rax

0x0000000000400f07 <+94>: je 0x400f0e <phase\_2+101>

0x0000000000400f09 <+96>: callq 0x400b00 <\_\_stack\_chk\_fail@plt>

0x0000000000400f0e <+101>: add $0x28,%rsp

---Type <return> to continue, or q <return> to quit---

0x0000000000400f12 <+105>: pop %rbx

0x0000000000400f13 <+106>: pop %rbp

0x0000000000400f14 <+107>: retq

End of assembler dump.

(gdb) ni

0x0000000000400ee6 in phase\_2 ()

(gdb) disas

Dump of assembler code for function phase\_2:

0x0000000000400ea9 <+0>: push %rbp

0x0000000000400eaa <+1>: push %rbx

0x0000000000400eab <+2>: sub $0x28,%rsp

0x0000000000400eaf <+6>: mov %fs:0x28,%rax

0x0000000000400eb8 <+15>: mov %rax,0x18(%rsp)

0x0000000000400ebd <+20>: xor %eax,%eax

0x0000000000400ebf <+22>: mov %rsp,%rsi

0x0000000000400ec2 <+25>: callq 0x40145f <read\_six\_numbers>

0x0000000000400ec7 <+30>: cmpl $0x0,(%rsp)

0x0000000000400ecb <+34>: jne 0x400ed4 <phase\_2+43>

0x0000000000400ecd <+36>: cmpl $0x1,0x4(%rsp)

0x0000000000400ed2 <+41>: je 0x400ed9 <phase\_2+48>

0x0000000000400ed4 <+43>: callq 0x40143d <explode\_bomb>

0x0000000000400ed9 <+48>: mov %rsp,%rbx

0x0000000000400edc <+51>: lea 0x10(%rsp),%rbp

0x0000000000400ee1 <+56>: mov 0x4(%rbx),%eax

0x0000000000400ee4 <+59>: add (%rbx),%eax

=> 0x0000000000400ee6 <+61>: cmp %eax,0x8(%rbx)

0x0000000000400ee9 <+64>: je 0x400ef0 <phase\_2+71>

0x0000000000400eeb <+66>: callq 0x40143d <explode\_bomb>

0x0000000000400ef0 <+71>: add $0x4,%rbx

0x0000000000400ef4 <+75>: cmp %rbp,%rbx

0x0000000000400ef7 <+78>: jne 0x400ee1 <phase\_2+56>

0x0000000000400ef9 <+80>: mov 0x18(%rsp),%rax

0x0000000000400efe <+85>: xor %fs:0x28,%rax

0x0000000000400f07 <+94>: je 0x400f0e <phase\_2+101>

0x0000000000400f09 <+96>: callq 0x400b00 <\_\_stack\_chk\_fail@plt>

0x0000000000400f0e <+101>: add $0x28,%rsp

---Type <return> to continue, or q <return> to quit---

0x0000000000400f12 <+105>: pop %rbx

0x0000000000400f13 <+106>: pop %rbp

0x0000000000400f14 <+107>: retq

End of assembler dump.

(gdb) i r

rax 0x3 3

rbx 0x7ffffffee308 140737488282376

rcx 0x0 0

rdx 0x7ffffffee314 140737488282388

rsi 0x0 0

rdi 0x7ffffffedc70 140737488280688

rbp 0x7ffffffee310 0x7ffffffee310

rsp 0x7ffffffee300 0x7ffffffee300

r8 0x0 0

r9 0x0 0

r10 0x7fffff19ec40 140737473276992

r11 0x4025d4 4203988

r12 0x400c60 4197472

r13 0x7ffffffee420 140737488282656

r14 0x0 0

r15 0x0 0

rip 0x400ee6 0x400ee6 <phase\_2+61>

eflags 0x206 [ PF IF ]

cs 0x33 51

ss 0x2b 43

ds 0x0 0

es 0x0 0

fs 0x0 0

gs 0x0 0

(gdb) disas

Dump of assembler code for function phase\_2:

0x0000000000400ea9 <+0>: push %rbp

0x0000000000400eaa <+1>: push %rbx

0x0000000000400eab <+2>: sub $0x28,%rsp

0x0000000000400eaf <+6>: mov %fs:0x28,%rax

0x0000000000400eb8 <+15>: mov %rax,0x18(%rsp)

0x0000000000400ebd <+20>: xor %eax,%eax

0x0000000000400ebf <+22>: mov %rsp,%rsi

0x0000000000400ec2 <+25>: callq 0x40145f <read\_six\_numbers>

0x0000000000400ec7 <+30>: cmpl $0x0,(%rsp)

0x0000000000400ecb <+34>: jne 0x400ed4 <phase\_2+43>

0x0000000000400ecd <+36>: cmpl $0x1,0x4(%rsp)

0x0000000000400ed2 <+41>: je 0x400ed9 <phase\_2+48>

0x0000000000400ed4 <+43>: callq 0x40143d <explode\_bomb>

0x0000000000400ed9 <+48>: mov %rsp,%rbx

0x0000000000400edc <+51>: lea 0x10(%rsp),%rbp

0x0000000000400ee1 <+56>: mov 0x4(%rbx),%eax

0x0000000000400ee4 <+59>: add (%rbx),%eax

=> 0x0000000000400ee6 <+61>: cmp %eax,0x8(%rbx)

0x0000000000400ee9 <+64>: je 0x400ef0 <phase\_2+71>

0x0000000000400eeb <+66>: callq 0x40143d <explode\_bomb>

0x0000000000400ef0 <+71>: add $0x4,%rbx

0x0000000000400ef4 <+75>: cmp %rbp,%rbx

0x0000000000400ef7 <+78>: jne 0x400ee1 <phase\_2+56>

0x0000000000400ef9 <+80>: mov 0x18(%rsp),%rax

0x0000000000400efe <+85>: xor %fs:0x28,%rax

0x0000000000400f07 <+94>: je 0x400f0e <phase\_2+101>

0x0000000000400f09 <+96>: callq 0x400b00 <\_\_stack\_chk\_fail@plt>

0x0000000000400f0e <+101>: add $0x28,%rsp

---Type <return> to continue, or q <return> to quit---

0x0000000000400f12 <+105>: pop %rbx

0x0000000000400f13 <+106>: pop %rbp

0x0000000000400f14 <+107>: retq

End of assembler dump.

(gdb) i r

rax 0x3 3

rbx 0x7ffffffee308 140737488282376

rcx 0x0 0

rdx 0x7ffffffee314 140737488282388

rsi 0x0 0

rdi 0x7ffffffedc70 140737488280688

rbp 0x7ffffffee310 0x7ffffffee310

rsp 0x7ffffffee300 0x7ffffffee300

r8 0x0 0

r9 0x0 0

r10 0x7fffff19ec40 140737473276992

r11 0x4025d4 4203988

r12 0x400c60 4197472

r13 0x7ffffffee420 140737488282656

r14 0x0 0

r15 0x0 0

rip 0x400ee6 0x400ee6 <phase\_2+61>

eflags 0x206 [ PF IF ]

cs 0x33 51

ss 0x2b 43

ds 0x0 0

es 0x0 0

fs 0x0 0

gs 0x0 0

(gdb) disas

Dump of assembler code for function phase\_2:

0x0000000000400ea9 <+0>: push %rbp

0x0000000000400eaa <+1>: push %rbx

0x0000000000400eab <+2>: sub $0x28,%rsp

0x0000000000400eaf <+6>: mov %fs:0x28,%rax

0x0000000000400eb8 <+15>: mov %rax,0x18(%rsp)

0x0000000000400ebd <+20>: xor %eax,%eax

0x0000000000400ebf <+22>: mov %rsp,%rsi

0x0000000000400ec2 <+25>: callq 0x40145f <read\_six\_numbers>

0x0000000000400ec7 <+30>: cmpl $0x0,(%rsp)

0x0000000000400ecb <+34>: jne 0x400ed4 <phase\_2+43>

0x0000000000400ecd <+36>: cmpl $0x1,0x4(%rsp)

0x0000000000400ed2 <+41>: je 0x400ed9 <phase\_2+48>

0x0000000000400ed4 <+43>: callq 0x40143d <explode\_bomb>

0x0000000000400ed9 <+48>: mov %rsp,%rbx

0x0000000000400edc <+51>: lea 0x10(%rsp),%rbp

0x0000000000400ee1 <+56>: mov 0x4(%rbx),%eax

0x0000000000400ee4 <+59>: add (%rbx),%eax

=> 0x0000000000400ee6 <+61>: cmp %eax,0x8(%rbx)

0x0000000000400ee9 <+64>: je 0x400ef0 <phase\_2+71>

0x0000000000400eeb <+66>: callq 0x40143d <explode\_bomb>

0x0000000000400ef0 <+71>: add $0x4,%rbx

0x0000000000400ef4 <+75>: cmp %rbp,%rbx

0x0000000000400ef7 <+78>: jne 0x400ee1 <phase\_2+56>

0x0000000000400ef9 <+80>: mov 0x18(%rsp),%rax

0x0000000000400efe <+85>: xor %fs:0x28,%rax

0x0000000000400f07 <+94>: je 0x400f0e <phase\_2+101>

0x0000000000400f09 <+96>: callq 0x400b00 <\_\_stack\_chk\_fail@plt>

0x0000000000400f0e <+101>: add $0x28,%rsp

---Type <return> to continue, or q <return> to quit---

0x0000000000400f12 <+105>: pop %rbx

0x0000000000400f13 <+106>: pop %rbp

0x0000000000400f14 <+107>: retq

End of assembler dump.

(gdb) ni

0x0000000000400ee9 in phase\_2 ()

(gdb) disas

Dump of assembler code for function phase\_2:

0x0000000000400ea9 <+0>: push %rbp

0x0000000000400eaa <+1>: push %rbx

0x0000000000400eab <+2>: sub $0x28,%rsp

0x0000000000400eaf <+6>: mov %fs:0x28,%rax

0x0000000000400eb8 <+15>: mov %rax,0x18(%rsp)

0x0000000000400ebd <+20>: xor %eax,%eax

0x0000000000400ebf <+22>: mov %rsp,%rsi

0x0000000000400ec2 <+25>: callq 0x40145f <read\_six\_numbers>

0x0000000000400ec7 <+30>: cmpl $0x0,(%rsp)

0x0000000000400ecb <+34>: jne 0x400ed4 <phase\_2+43>

0x0000000000400ecd <+36>: cmpl $0x1,0x4(%rsp)

0x0000000000400ed2 <+41>: je 0x400ed9 <phase\_2+48>

0x0000000000400ed4 <+43>: callq 0x40143d <explode\_bomb>

0x0000000000400ed9 <+48>: mov %rsp,%rbx

0x0000000000400edc <+51>: lea 0x10(%rsp),%rbp

0x0000000000400ee1 <+56>: mov 0x4(%rbx),%eax

0x0000000000400ee4 <+59>: add (%rbx),%eax

0x0000000000400ee6 <+61>: cmp %eax,0x8(%rbx)

=> 0x0000000000400ee9 <+64>: je 0x400ef0 <phase\_2+71>

0x0000000000400eeb <+66>: callq 0x40143d <explode\_bomb>

0x0000000000400ef0 <+71>: add $0x4,%rbx

0x0000000000400ef4 <+75>: cmp %rbp,%rbx

0x0000000000400ef7 <+78>: jne 0x400ee1 <phase\_2+56>

0x0000000000400ef9 <+80>: mov 0x18(%rsp),%rax

0x0000000000400efe <+85>: xor %fs:0x28,%rax

0x0000000000400f07 <+94>: je 0x400f0e <phase\_2+101>

0x0000000000400f09 <+96>: callq 0x400b00 <\_\_stack\_chk\_fail@plt>

0x0000000000400f0e <+101>: add $0x28,%rsp

---Type <return> to continue, or q <return> to quit---

0x0000000000400f12 <+105>: pop %rbx

0x0000000000400f13 <+106>: pop %rbp

0x0000000000400f14 <+107>: retq

End of assembler dump.

(gdb) ni

0x0000000000400ef0 in phase\_2 ()

(gdb) disas

Dump of assembler code for function phase\_2:

0x0000000000400ea9 <+0>: push %rbp

0x0000000000400eaa <+1>: push %rbx

0x0000000000400eab <+2>: sub $0x28,%rsp

0x0000000000400eaf <+6>: mov %fs:0x28,%rax

0x0000000000400eb8 <+15>: mov %rax,0x18(%rsp)

0x0000000000400ebd <+20>: xor %eax,%eax

0x0000000000400ebf <+22>: mov %rsp,%rsi

0x0000000000400ec2 <+25>: callq 0x40145f <read\_six\_numbers>

0x0000000000400ec7 <+30>: cmpl $0x0,(%rsp)

0x0000000000400ecb <+34>: jne 0x400ed4 <phase\_2+43>

0x0000000000400ecd <+36>: cmpl $0x1,0x4(%rsp)

0x0000000000400ed2 <+41>: je 0x400ed9 <phase\_2+48>

0x0000000000400ed4 <+43>: callq 0x40143d <explode\_bomb>

0x0000000000400ed9 <+48>: mov %rsp,%rbx

0x0000000000400edc <+51>: lea 0x10(%rsp),%rbp

0x0000000000400ee1 <+56>: mov 0x4(%rbx),%eax

0x0000000000400ee4 <+59>: add (%rbx),%eax

0x0000000000400ee6 <+61>: cmp %eax,0x8(%rbx)

0x0000000000400ee9 <+64>: je 0x400ef0 <phase\_2+71>

0x0000000000400eeb <+66>: callq 0x40143d <explode\_bomb>

=> 0x0000000000400ef0 <+71>: add $0x4,%rbx

0x0000000000400ef4 <+75>: cmp %rbp,%rbx

0x0000000000400ef7 <+78>: jne 0x400ee1 <phase\_2+56>

0x0000000000400ef9 <+80>: mov 0x18(%rsp),%rax

0x0000000000400efe <+85>: xor %fs:0x28,%rax

0x0000000000400f07 <+94>: je 0x400f0e <phase\_2+101>

0x0000000000400f09 <+96>: callq 0x400b00 <\_\_stack\_chk\_fail@plt>

0x0000000000400f0e <+101>: add $0x28,%rsp

---Type <return> to continue, or q <return> to quit---

0x0000000000400f12 <+105>: pop %rbx

0x0000000000400f13 <+106>: pop %rbp

0x0000000000400f14 <+107>: retq

End of assembler dump.

(gdb) ni

0x0000000000400ef4 in phase\_2 ()

(gdb) disas

Dump of assembler code for function phase\_2:

0x0000000000400ea9 <+0>: push %rbp

0x0000000000400eaa <+1>: push %rbx

0x0000000000400eab <+2>: sub $0x28,%rsp

0x0000000000400eaf <+6>: mov %fs:0x28,%rax

0x0000000000400eb8 <+15>: mov %rax,0x18(%rsp)

0x0000000000400ebd <+20>: xor %eax,%eax

0x0000000000400ebf <+22>: mov %rsp,%rsi

0x0000000000400ec2 <+25>: callq 0x40145f <read\_six\_numbers>

0x0000000000400ec7 <+30>: cmpl $0x0,(%rsp)

0x0000000000400ecb <+34>: jne 0x400ed4 <phase\_2+43>

0x0000000000400ecd <+36>: cmpl $0x1,0x4(%rsp)

0x0000000000400ed2 <+41>: je 0x400ed9 <phase\_2+48>

0x0000000000400ed4 <+43>: callq 0x40143d <explode\_bomb>

0x0000000000400ed9 <+48>: mov %rsp,%rbx

0x0000000000400edc <+51>: lea 0x10(%rsp),%rbp

0x0000000000400ee1 <+56>: mov 0x4(%rbx),%eax

0x0000000000400ee4 <+59>: add (%rbx),%eax

0x0000000000400ee6 <+61>: cmp %eax,0x8(%rbx)

0x0000000000400ee9 <+64>: je 0x400ef0 <phase\_2+71>

0x0000000000400eeb <+66>: callq 0x40143d <explode\_bomb>

0x0000000000400ef0 <+71>: add $0x4,%rbx

=> 0x0000000000400ef4 <+75>: cmp %rbp,%rbx

0x0000000000400ef7 <+78>: jne 0x400ee1 <phase\_2+56>

0x0000000000400ef9 <+80>: mov 0x18(%rsp),%rax

0x0000000000400efe <+85>: xor %fs:0x28,%rax

0x0000000000400f07 <+94>: je 0x400f0e <phase\_2+101>

0x0000000000400f09 <+96>: callq 0x400b00 <\_\_stack\_chk\_fail@plt>

0x0000000000400f0e <+101>: add $0x28,%rsp

---Type <return> to continue, or q <return> to quit---

0x0000000000400f12 <+105>: pop %rbx

0x0000000000400f13 <+106>: pop %rbp

0x0000000000400f14 <+107>: retq

End of assembler dump.

(gdb) i r

rax 0x3 3

rbx 0x7ffffffee30c 140737488282380

rcx 0x0 0

rdx 0x7ffffffee314 140737488282388

rsi 0x0 0

rdi 0x7ffffffedc70 140737488280688

rbp 0x7ffffffee310 0x7ffffffee310

rsp 0x7ffffffee300 0x7ffffffee300

r8 0x0 0

r9 0x0 0

r10 0x7fffff19ec40 140737473276992

r11 0x4025d4 4203988

r12 0x400c60 4197472

r13 0x7ffffffee420 140737488282656

r14 0x0 0

r15 0x0 0

rip 0x400ef4 0x400ef4 <phase\_2+75>

eflags 0x206 [ PF IF ]

cs 0x33 51

ss 0x2b 43

ds 0x0 0

es 0x0 0

fs 0x0 0

gs 0x0 0

(gdb) ni

0x0000000000400ef7 in phase\_2 ()

(gdb) ni

0x0000000000400ee1 in phase\_2 ()

(gdb) disas

Dump of assembler code for function phase\_2:

0x0000000000400ea9 <+0>: push %rbp

0x0000000000400eaa <+1>: push %rbx

0x0000000000400eab <+2>: sub $0x28,%rsp

0x0000000000400eaf <+6>: mov %fs:0x28,%rax

0x0000000000400eb8 <+15>: mov %rax,0x18(%rsp)

0x0000000000400ebd <+20>: xor %eax,%eax

0x0000000000400ebf <+22>: mov %rsp,%rsi

0x0000000000400ec2 <+25>: callq 0x40145f <read\_six\_numbers>

0x0000000000400ec7 <+30>: cmpl $0x0,(%rsp)

0x0000000000400ecb <+34>: jne 0x400ed4 <phase\_2+43>

0x0000000000400ecd <+36>: cmpl $0x1,0x4(%rsp)

0x0000000000400ed2 <+41>: je 0x400ed9 <phase\_2+48>

0x0000000000400ed4 <+43>: callq 0x40143d <explode\_bomb>

0x0000000000400ed9 <+48>: mov %rsp,%rbx

0x0000000000400edc <+51>: lea 0x10(%rsp),%rbp

=> 0x0000000000400ee1 <+56>: mov 0x4(%rbx),%eax

0x0000000000400ee4 <+59>: add (%rbx),%eax

0x0000000000400ee6 <+61>: cmp %eax,0x8(%rbx)

0x0000000000400ee9 <+64>: je 0x400ef0 <phase\_2+71>

0x0000000000400eeb <+66>: callq 0x40143d <explode\_bomb>

0x0000000000400ef0 <+71>: add $0x4,%rbx

0x0000000000400ef4 <+75>: cmp %rbp,%rbx

0x0000000000400ef7 <+78>: jne 0x400ee1 <phase\_2+56>

0x0000000000400ef9 <+80>: mov 0x18(%rsp),%rax

0x0000000000400efe <+85>: xor %fs:0x28,%rax

0x0000000000400f07 <+94>: je 0x400f0e <phase\_2+101>

0x0000000000400f09 <+96>: callq 0x400b00 <\_\_stack\_chk\_fail@plt>

0x0000000000400f0e <+101>: add $0x28,%rsp

---Type <return> to continue, or q <return> to quit---

0x0000000000400f12 <+105>: pop %rbx

0x0000000000400f13 <+106>: pop %rbp

0x0000000000400f14 <+107>: retq

End of assembler dump.

(gdb) ni

0x0000000000400ee4 in phase\_2 ()

(gdb) ni

0x0000000000400ee6 in phase\_2 ()

(gdb) disas

Dump of assembler code for function phase\_2:

0x0000000000400ea9 <+0>: push %rbp

0x0000000000400eaa <+1>: push %rbx

0x0000000000400eab <+2>: sub $0x28,%rsp

0x0000000000400eaf <+6>: mov %fs:0x28,%rax

0x0000000000400eb8 <+15>: mov %rax,0x18(%rsp)

0x0000000000400ebd <+20>: xor %eax,%eax

0x0000000000400ebf <+22>: mov %rsp,%rsi

0x0000000000400ec2 <+25>: callq 0x40145f <read\_six\_numbers>

0x0000000000400ec7 <+30>: cmpl $0x0,(%rsp)

0x0000000000400ecb <+34>: jne 0x400ed4 <phase\_2+43>

0x0000000000400ecd <+36>: cmpl $0x1,0x4(%rsp)

0x0000000000400ed2 <+41>: je 0x400ed9 <phase\_2+48>

0x0000000000400ed4 <+43>: callq 0x40143d <explode\_bomb>

0x0000000000400ed9 <+48>: mov %rsp,%rbx

0x0000000000400edc <+51>: lea 0x10(%rsp),%rbp

0x0000000000400ee1 <+56>: mov 0x4(%rbx),%eax

0x0000000000400ee4 <+59>: add (%rbx),%eax

=> 0x0000000000400ee6 <+61>: cmp %eax,0x8(%rbx)

0x0000000000400ee9 <+64>: je 0x400ef0 <phase\_2+71>

0x0000000000400eeb <+66>: callq 0x40143d <explode\_bomb>

0x0000000000400ef0 <+71>: add $0x4,%rbx

0x0000000000400ef4 <+75>: cmp %rbp,%rbx

0x0000000000400ef7 <+78>: jne 0x400ee1 <phase\_2+56>

0x0000000000400ef9 <+80>: mov 0x18(%rsp),%rax

0x0000000000400efe <+85>: xor %fs:0x28,%rax

0x0000000000400f07 <+94>: je 0x400f0e <phase\_2+101>

0x0000000000400f09 <+96>: callq 0x400b00 <\_\_stack\_chk\_fail@plt>

0x0000000000400f0e <+101>: add $0x28,%rsp

---Type <return> to continue, or q <return> to quit---

0x0000000000400f12 <+105>: pop %rbx

0x0000000000400f13 <+106>: pop %rbp

0x0000000000400f14 <+107>: retq

End of assembler dump.

(gdb) i r

rax 0x5 5 **// the last input integer is 5**

rbx 0x7ffffffee30c 140737488282380

rcx 0x0 0

rdx 0x7ffffffee314 140737488282388

rsi 0x0 0

rdi 0x7ffffffedc70 140737488280688

rbp 0x7ffffffee310 0x7ffffffee310

rsp 0x7ffffffee300 0x7ffffffee300

r8 0x0 0

r9 0x0 0

r10 0x7fffff19ec40 140737473276992

r11 0x4025d4 4203988

r12 0x400c60 4197472

r13 0x7ffffffee420 140737488282656

r14 0x0 0

r15 0x0 0

rip 0x400ee6 0x400ee6 <phase\_2+61>

eflags 0x206 [ PF IF ]

cs 0x33 51

ss 0x2b 43

ds 0x0 0

es 0x0 0

fs 0x0 0

gs 0x0 0

(gdb) r

The program being debugged has been started already.

Start it from the beginning? (y or n) y

Starting program: /mnt/d/one drive/OneDrive/Desktop/12190094/Assignment 1/bomb001/bomb

Welcome to my fiendish little bomb. You have 6 phases with

which to blow yourself up. Have a nice day!

The moon unit will be divided into two divisions.

Phase 1 defused. How about the next one?

0 1 1 2 3 5

Breakpoint 1, 0x0000000000400ea9 in phase\_2 ()

(gdb) i b

Num Type Disp Enb Address What

1 breakpoint keep y 0x0000000000400ea9 <phase\_2>

breakpoint already hit 1 time

2 breakpoint keep y 0x000000000040143d <explode\_bomb>

(gdb) del 1

(gdb) r

The program being debugged has been started already.

Start it from the beginning? (y or n) y

Starting program: /mnt/d/one drive/OneDrive/Desktop/12190094/Assignment 1/bomb001/bomb

Welcome to my fiendish little bomb. You have 6 phases with

which to blow yourself up. Have a nice day!

The moon unit will be divided into two divisions.

Phase 1 defused. How about the next one?

0 1 1 2 3 5

That's number 2. Keep going!