(gdb) break phase\_4

Breakpoint 1 at 0x40100e

(gdb) run

Starting program: /mnt/c/Users/Hp/OneDrive/Documents/3rd Year/Computer System I ITS304/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!

2 409

Halfway there!

hi //Try input

Breakpoint 1, 0x000000000040100e in phase\_4 ()

(gdb) disas

Dump of assembler code for function phase\_4:

=> 0x000000000040100e <+0>: sub $0x18,%rsp //makes stack frame

0x0000000000401012 <+4>: mov %fs:0x28,%rax

0x000000000040101b <+13>: mov %rax,0x8(%rsp)

0x0000000000401020 <+18>: xor %eax,%eax // eax = eax ^ eax

0x0000000000401022 <+20>: lea 0x4(%rsp),%rcx // rcx = rsp - 0x4

0x0000000000401027 <+25>: mov %rsp,%rdx // rdx = rsp

0x000000000040102a <+28>: mov $0x4025cf,%esi //format of answer: %d %d

0x000000000040102f <+33>: callq 0x400bb0 <\_\_isoc99\_sscanf@plt>

0x0000000000401034 <+38>: cmp $0x2,%eax

0x0000000000401037 <+41>: jne 0x40103f <phase\_4+49>

0x0000000000401039 <+43>: cmpl $0xe,(%rsp)

0x000000000040103d <+47>: jbe 0x401044 <phase\_4+54>

0x000000000040103f <+49>: callq 0x40143d <explode\_bomb>

0x0000000000401044 <+54>: mov $0xe,%edx

0x0000000000401049 <+59>: mov $0x0,%esi

0x000000000040104e <+64>: mov (%rsp),%edi

0x0000000000401051 <+67>: callq 0x400fdb <func4>

0x0000000000401056 <+72>: cmp $0x23,%eax

0x0000000000401059 <+75>: jne 0x401062 <phase\_4+84>

0x000000000040105b <+77>: cmpl $0x23,0x4(%rsp)

0x0000000000401060 <+82>: je 0x401067 <phase\_4+89>

0x0000000000401062 <+84>: callq 0x40143d <explode\_bomb>

0x0000000000401067 <+89>: mov 0x8(%rsp),%rax

0x000000000040106c <+94>: xor %fs:0x28,%rax

0x0000000000401075 <+103>: je 0x40107c <phase\_4+110>

0x0000000000401077 <+105>: callq 0x400b00 <\_\_stack\_chk\_fail@plt>

0x000000000040107c <+110>: add $0x18,%rsp

0x0000000000401080 <+114>: retq

End of assembler dump.

(gdb) x/s 0x4025cf

0x4025cf: "%d %d" //format of answer

(gdb) run

The program being debugged has been started already.

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

Starting program: /mnt/c/Users/Hp/OneDrive/Documents/3rd Year/Computer System I ITS304/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!

2 409

Halfway there!

2 3 //input

Breakpoint 1, 0x000000000040100e in phase\_4 ()

(gdb) disas

Dump of assembler code for function phase\_4:

=> 0x000000000040100e <+0>: sub $0x18,%rsp

0x0000000000401012 <+4>: mov %fs:0x28,%rax

0x000000000040101b <+13>: mov %rax,0x8(%rsp)

0x0000000000401020 <+18>: xor %eax,%eax

0x0000000000401022 <+20>: lea 0x4(%rsp),%rcx

0x0000000000401027 <+25>: mov %rsp,%rdx

0x000000000040102a <+28>: mov $0x4025cf,%esi

0x000000000040102f <+33>: callq 0x400bb0 <\_\_isoc99\_sscanf@plt>

0x0000000000401034 <+38>: cmp $0x2,%eax

0x0000000000401037 <+41>: jne 0x40103f <phase\_4+49>

0x0000000000401039 <+43>: cmpl $0xe,(%rsp)

0x000000000040103d <+47>: jbe 0x401044 <phase\_4+54>

0x000000000040103f <+49>: callq 0x40143d <explode\_bomb>

0x0000000000401044 <+54>: mov $0xe,%edx

0x0000000000401049 <+59>: mov $0x0,%esi

0x000000000040104e <+64>: mov (%rsp),%edi

0x0000000000401051 <+67>: callq 0x400fdb <func4>

0x0000000000401056 <+72>: cmp $0x23,%eax

0x0000000000401059 <+75>: jne 0x401062 <phase\_4+84>

0x000000000040105b <+77>: cmpl $0x23,0x4(%rsp)

0x0000000000401060 <+82>: je 0x401067 <phase\_4+89>

0x0000000000401062 <+84>: callq 0x40143d <explode\_bomb>

0x0000000000401067 <+89>: mov 0x8(%rsp),%rax

0x000000000040106c <+94>: xor %fs:0x28,%rax

0x0000000000401075 <+103>: je 0x40107c <phase\_4+110>

0x0000000000401077 <+105>: callq 0x400b00 <\_\_stack\_chk\_fail@plt>

0x000000000040107c <+110>: add $0x18,%rsp

0x0000000000401080 <+114>: retq

End of assembler dump.

(gdb) u\*0x0000000000401034

0x0000000000401034 in phase\_4 ()

(gdb) disas

Dump of assembler code for function phase\_4:

0x000000000040100e <+0>: sub $0x18,%rsp

0x0000000000401012 <+4>: mov %fs:0x28,%rax

0x000000000040101b <+13>: mov %rax,0x8(%rsp)

0x0000000000401020 <+18>: xor %eax,%eax

0x0000000000401022 <+20>: lea 0x4(%rsp),%rcx

0x0000000000401027 <+25>: mov %rsp,%rdx

0x000000000040102a <+28>: mov $0x4025cf,%esi

0x000000000040102f <+33>: callq 0x400bb0 <\_\_isoc99\_sscanf@plt>

=> 0x0000000000401034 <+38>: cmp $0x2,%eax // if NOT equal to 2 jump to explode bomb

0x0000000000401037 <+41>: jne 0x40103f <phase\_4+49>

0x0000000000401039 <+43>: cmpl $0xe,(%rsp)

0x000000000040103d <+47>: jbe 0x401044 <phase\_4+54>

0x000000000040103f <+49>: callq 0x40143d <explode\_bomb>

0x0000000000401044 <+54>: mov $0xe,%edx

0x0000000000401049 <+59>: mov $0x0,%esi

0x000000000040104e <+64>: mov (%rsp),%edi

0x0000000000401051 <+67>: callq 0x400fdb <func4>

0x0000000000401056 <+72>: cmp $0x23,%eax

0x0000000000401059 <+75>: jne 0x401062 <phase\_4+84>

0x000000000040105b <+77>: cmpl $0x23,0x4(%rsp)

0x0000000000401060 <+82>: je 0x401067 <phase\_4+89>

0x0000000000401062 <+84>: callq 0x40143d <explode\_bomb>

0x0000000000401067 <+89>: mov 0x8(%rsp),%rax

0x000000000040106c <+94>: xor %fs:0x28,%rax

0x0000000000401075 <+103>: je 0x40107c <phase\_4+110>

0x0000000000401077 <+105>: callq 0x400b00 <\_\_stack\_chk\_fail@plt>

0x000000000040107c <+110>: add $0x18,%rsp

0x0000000000401080 <+114>: retq

End of assembler dump.

(gdb) i r

rax 0x2 2

rbx 0x4021f0 4202992

rcx 0x0 0

rdx 0x7ffffffee294 140737488282260

rsi 0x0 0

rdi 0x7ffffffedc40 140737488280640

rbp 0x0 0x0

rsp 0x7ffffffee290 0x7ffffffee290

r8 0xffffffff 4294967295

r9 0x0 0

r10 0x7fffff74eac0 140737479240384

r11 0x0 0

r12 0x400c60 4197472

r13 0x7ffffffee3a0 140737488282528

r14 0x0 0

r15 0x0 0

rip 0x401034 0x401034 <phase\_4+38>

eflags 0x202 [ 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\_4:

0x000000000040100e <+0>: sub $0x18,%rsp

0x0000000000401012 <+4>: mov %fs:0x28,%rax

0x000000000040101b <+13>: mov %rax,0x8(%rsp)

0x0000000000401020 <+18>: xor %eax,%eax

0x0000000000401022 <+20>: lea 0x4(%rsp),%rcx

0x0000000000401027 <+25>: mov %rsp,%rdx

0x000000000040102a <+28>: mov $0x4025cf,%esi

0x000000000040102f <+33>: callq 0x400bb0 <\_\_isoc99\_sscanf@plt>

=> 0x0000000000401034 <+38>: cmp $0x2,%eax

0x0000000000401037 <+41>: jne 0x40103f <phase\_4+49>

0x0000000000401039 <+43>: cmpl $0xe,(%rsp) //0xe =14

0x000000000040103d <+47>: jbe 0x401044 <phase\_4+54> //if first input(Destination) < 14(Source), skip bomb

0x000000000040103f <+49>: callq 0x40143d <explode\_bomb>

0x0000000000401044 <+54>: mov $0xe,%edx //%edx = 14

0x0000000000401049 <+59>: mov $0x0,%esi //%esi = 0

0x000000000040104e <+64>: mov (%rsp),%edi //%edi = first input

0x0000000000401051 <+67>: callq 0x400fdb <func4>

0x0000000000401056 <+72>: cmp $0x23,%eax

0x0000000000401059 <+75>: jne 0x401062 <phase\_4+84>

0x000000000040105b <+77>: cmpl $0x23,0x4(%rsp)

0x0000000000401060 <+82>: je 0x401067 <phase\_4+89>

0x0000000000401062 <+84>: callq 0x40143d <explode\_bomb>

0x0000000000401067 <+89>: mov 0x8(%rsp),%rax

0x000000000040106c <+94>: xor %fs:0x28,%rax

0x0000000000401075 <+103>: je 0x40107c <phase\_4+110>

0x0000000000401077 <+105>: callq 0x400b00 <\_\_stack\_chk\_fail@plt>

0x000000000040107c <+110>: add $0x18,%rsp

0x0000000000401080 <+114>: retq

End of assembler dump.

(gdb) ni

0x0000000000401037 in phase\_4 ()

(gdb) ni

0x0000000000401039 in phase\_4 ()

(gdb) disas

Dump of assembler code for function phase\_4:

0x000000000040100e <+0>: sub $0x18,%rsp

0x0000000000401012 <+4>: mov %fs:0x28,%rax

0x000000000040101b <+13>: mov %rax,0x8(%rsp)

0x0000000000401020 <+18>: xor %eax,%eax

0x0000000000401022 <+20>: lea 0x4(%rsp),%rcx

0x0000000000401027 <+25>: mov %rsp,%rdx

0x000000000040102a <+28>: mov $0x4025cf,%esi

0x000000000040102f <+33>: callq 0x400bb0 <\_\_isoc99\_sscanf@plt>

0x0000000000401034 <+38>: cmp $0x2,%eax

0x0000000000401037 <+41>: jne 0x40103f <phase\_4+49>

=> 0x0000000000401039 <+43>: cmpl $0xe,(%rsp)

0x000000000040103d <+47>: jbe 0x401044 <phase\_4+54>

0x000000000040103f <+49>: callq 0x40143d <explode\_bomb>

0x0000000000401044 <+54>: mov $0xe,%edx

0x0000000000401049 <+59>: mov $0x0,%esi

0x000000000040104e <+64>: mov (%rsp),%edi

0x0000000000401051 <+67>: callq 0x400fdb <func4>

0x0000000000401056 <+72>: cmp $0x23,%eax

0x0000000000401059 <+75>: jne 0x401062 <phase\_4+84>

0x000000000040105b <+77>: cmpl $0x23,0x4(%rsp)

0x0000000000401060 <+82>: je 0x401067 <phase\_4+89>

0x0000000000401062 <+84>: callq 0x40143d <explode\_bomb>

0x0000000000401067 <+89>: mov 0x8(%rsp),%rax

0x000000000040106c <+94>: xor %fs:0x28,%rax

0x0000000000401075 <+103>: je 0x40107c <phase\_4+110>

0x0000000000401077 <+105>: callq 0x400b00 <\_\_stack\_chk\_fail@plt>

0x000000000040107c <+110>: add $0x18,%rsp

0x0000000000401080 <+114>: retq

End of assembler dump.

(gdb) x/d $rsp

0x7ffffffee290: 2

(gdb) disas

Dump of assembler code for function phase\_4:

0x000000000040100e <+0>: sub $0x18,%rsp

0x0000000000401012 <+4>: mov %fs:0x28,%rax

0x000000000040101b <+13>: mov %rax,0x8(%rsp)

0x0000000000401020 <+18>: xor %eax,%eax

0x0000000000401022 <+20>: lea 0x4(%rsp),%rcx

0x0000000000401027 <+25>: mov %rsp,%rdx

0x000000000040102a <+28>: mov $0x4025cf,%esi

0x000000000040102f <+33>: callq 0x400bb0 <\_\_isoc99\_sscanf@plt>

0x0000000000401034 <+38>: cmp $0x2,%eax

0x0000000000401037 <+41>: jne 0x40103f <phase\_4+49>

=> 0x0000000000401039 <+43>: cmpl $0xe,(%rsp)

0x000000000040103d <+47>: jbe 0x401044 <phase\_4+54>

0x000000000040103f <+49>: callq 0x40143d <explode\_bomb>

0x0000000000401044 <+54>: mov $0xe,%edx

0x0000000000401049 <+59>: mov $0x0,%esi

0x000000000040104e <+64>: mov (%rsp),%edi

0x0000000000401051 <+67>: callq 0x400fdb <func4> //calls <func4>

0x0000000000401056 <+72>: cmp $0x23,%eax

0x0000000000401059 <+75>: jne 0x401062 <phase\_4+84>

0x000000000040105b <+77>: cmpl $0x23,0x4(%rsp)

0x0000000000401060 <+82>: je 0x401067 <phase\_4+89>

0x0000000000401062 <+84>: callq 0x40143d <explode\_bomb>

0x0000000000401067 <+89>: mov 0x8(%rsp),%rax

0x000000000040106c <+94>: xor %fs:0x28,%rax

0x0000000000401075 <+103>: je 0x40107c <phase\_4+110>

0x0000000000401077 <+105>: callq 0x400b00 <\_\_stack\_chk\_fail@plt>

0x000000000040107c <+110>: add $0x18,%rsp

0x0000000000401080 <+114>: retq

End of assembler dump.

(gdb) u\*0x0000000000401051

0x0000000000401051 in phase\_4 ()

(gdb) si //getting inside function

0x0000000000400fdb in func4 ()

(gdb) disas

Dump of assembler code for function func4:

=> 0x0000000000400fdb <+0>: push %rbx

0x0000000000400fdc <+1>: mov %edx,%eax

0x0000000000400fde <+3>: sub %esi,%eax

0x0000000000400fe0 <+5>: mov %eax,%ebx

0x0000000000400fe2 <+7>: shr $0x1f,%ebx

0x0000000000400fe5 <+10>: add %ebx,%eax

0x0000000000400fe7 <+12>: sar %eax

0x0000000000400fe9 <+14>: lea (%rax,%rsi,1),%ebx

0x0000000000400fec <+17>: cmp %edi,%ebx

0x0000000000400fee <+19>: jle 0x400ffc <func4+33>

0x0000000000400ff0 <+21>: lea -0x1(%rbx),%edx

0x0000000000400ff3 <+24>: callq 0x400fdb <func4>

0x0000000000400ff8 <+29>: add %ebx,%eax

0x0000000000400ffa <+31>: jmp 0x40100c <func4+49>

0x0000000000400ffc <+33>: mov %ebx,%eax

0x0000000000400ffe <+35>: cmp %edi,%ebx

0x0000000000401000 <+37>: jge 0x40100c <func4+49>

0x0000000000401002 <+39>: lea 0x1(%rbx),%esi

0x0000000000401005 <+42>: callq 0x400fdb <func4>

0x000000000040100a <+47>: add %ebx,%eax

0x000000000040100c <+49>: pop %rbx

0x000000000040100d <+50>: retq

End of assembler dump.

(gdb) u\*0x0000000000400fec

0x0000000000400fec in func4 ()

(gdb) disas

Dump of assembler code for function func4:

0x0000000000400fdb <+0>: push %rbx

0x0000000000400fdc <+1>: mov %edx,%eax

0x0000000000400fde <+3>: sub %esi,%eax

0x0000000000400fe0 <+5>: mov %eax,%ebx

0x0000000000400fe2 <+7>: shr $0x1f,%ebx

0x0000000000400fe5 <+10>: add %ebx,%eax

0x0000000000400fe7 <+12>: sar %eax

0x0000000000400fe9 <+14>: lea (%rax,%rsi,1),%ebx

=> 0x0000000000400fec <+17>: cmp %edi,%ebx //if %ebx <= %edi jump instruction

0x0000000000400fee <+19>: jle 0x400ffc <func4+33>

0x0000000000400ff0 <+21>: lea -0x1(%rbx),%edx

0x0000000000400ff3 <+24>: callq 0x400fdb <func4>

0x0000000000400ff8 <+29>: add %ebx,%eax

0x0000000000400ffa <+31>: jmp 0x40100c <func4+49>

0x0000000000400ffc <+33>: mov %ebx,%eax

0x0000000000400ffe <+35>: cmp %edi,%ebx

0x0000000000401000 <+37>: jge 0x40100c <func4+49>

0x0000000000401002 <+39>: lea 0x1(%rbx),%esi

0x0000000000401005 <+42>: callq 0x400fdb <func4>

0x000000000040100a <+47>: add %ebx,%eax

0x000000000040100c <+49>: pop %rbx

0x000000000040100d <+50>: retq

End of assembler dump.

(gdb) i r

rax 0x7 7

rbx 0x7 7

rcx 0x0 0

rdx 0xe 14

rsi 0x0 0

rdi 0x2 2

rbp 0x0 0x0

rsp 0x7ffffffee280 0x7ffffffee280

r8 0xffffffff 4294967295

r9 0x0 0

r10 0x7fffff74eac0 140737479240384

r11 0x0 0

r12 0x400c60 4197472

r13 0x7ffffffee3a0 140737488282528

r14 0x0 0

r15 0x0 0

rip 0x400fec 0x400fec <func4+17>

eflags 0x202 [ IF ]

cs 0x33 51

ss 0x2b 43

ds 0x0 0

es 0x0 0

fs 0x0 0

gs 0x0 0

(gdb) run

The program being debugged has been started already.

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

Starting program: /mnt/c/Users/Hp/OneDrive/Documents/3rd Year/Computer System I ITS304/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!

2 409

Halfway there!

8 3

Breakpoint 1, 0x000000000040100e in phase\_4 ()

(gdb) disas

Dump of assembler code for function phase\_4:

=> 0x000000000040100e <+0>: sub $0x18,%rsp

0x0000000000401012 <+4>: mov %fs:0x28,%rax

0x000000000040101b <+13>: mov %rax,0x8(%rsp)

0x0000000000401020 <+18>: xor %eax,%eax

0x0000000000401022 <+20>: lea 0x4(%rsp),%rcx

0x0000000000401027 <+25>: mov %rsp,%rdx

0x000000000040102a <+28>: mov $0x4025cf,%esi

0x000000000040102f <+33>: callq 0x400bb0 <\_\_isoc99\_sscanf@plt>

0x0000000000401034 <+38>: cmp $0x2,%eax

0x0000000000401037 <+41>: jne 0x40103f <phase\_4+49>

0x0000000000401039 <+43>: cmpl $0xe,(%rsp)

0x000000000040103d <+47>: jbe 0x401044 <phase\_4+54>

0x000000000040103f <+49>: callq 0x40143d <explode\_bomb>

0x0000000000401044 <+54>: mov $0xe,%edx

0x0000000000401049 <+59>: mov $0x0,%esi

0x000000000040104e <+64>: mov (%rsp),%edi

0x0000000000401051 <+67>: callq 0x400fdb <func4>

0x0000000000401056 <+72>: cmp $0x23,%eax

0x0000000000401059 <+75>: jne 0x401062 <phase\_4+84>

0x000000000040105b <+77>: cmpl $0x23,0x4(%rsp)

0x0000000000401060 <+82>: je 0x401067 <phase\_4+89>

0x0000000000401062 <+84>: callq 0x40143d <explode\_bomb>

0x0000000000401067 <+89>: mov 0x8(%rsp),%rax

0x000000000040106c <+94>: xor %fs:0x28,%rax

0x0000000000401075 <+103>: je 0x40107c <phase\_4+110>

0x0000000000401077 <+105>: callq 0x400b00 <\_\_stack\_chk\_fail@plt>

0x000000000040107c <+110>: add $0x18,%rsp

0x0000000000401080 <+114>: retq

End of assembler dump.

(gdb) u\* 0x0000000000401051

0x0000000000401051 in phase\_4 ()

(gdb) disas

Dump of assembler code for function phase\_4:

0x000000000040100e <+0>: sub $0x18,%rsp

0x0000000000401012 <+4>: mov %fs:0x28,%rax

0x000000000040101b <+13>: mov %rax,0x8(%rsp)

0x0000000000401020 <+18>: xor %eax,%eax

0x0000000000401022 <+20>: lea 0x4(%rsp),%rcx

0x0000000000401027 <+25>: mov %rsp,%rdx

0x000000000040102a <+28>: mov $0x4025cf,%esi

0x000000000040102f <+33>: callq 0x400bb0 <\_\_isoc99\_sscanf@plt>

0x0000000000401034 <+38>: cmp $0x2,%eax

0x0000000000401037 <+41>: jne 0x40103f <phase\_4+49>

0x0000000000401039 <+43>: cmpl $0xe,(%rsp)

0x000000000040103d <+47>: jbe 0x401044 <phase\_4+54>

0x000000000040103f <+49>: callq 0x40143d <explode\_bomb>

0x0000000000401044 <+54>: mov $0xe,%edx

0x0000000000401049 <+59>: mov $0x0,%esi

0x000000000040104e <+64>: mov (%rsp),%edi

=> 0x0000000000401051 <+67>: callq 0x400fdb <func4>

0x0000000000401056 <+72>: cmp $0x23,%eax

0x0000000000401059 <+75>: jne 0x401062 <phase\_4+84>

0x000000000040105b <+77>: cmpl $0x23,0x4(%rsp)

0x0000000000401060 <+82>: je 0x401067 <phase\_4+89>

0x0000000000401062 <+84>: callq 0x40143d <explode\_bomb>

0x0000000000401067 <+89>: mov 0x8(%rsp),%rax

0x000000000040106c <+94>: xor %fs:0x28,%rax

0x0000000000401075 <+103>: je 0x40107c <phase\_4+110>

0x0000000000401077 <+105>: callq 0x400b00 <\_\_stack\_chk\_fail@plt>

0x000000000040107c <+110>: add $0x18,%rsp

0x0000000000401080 <+114>: retq

End of assembler dump.

(gdb) si

0x0000000000400fdb in func4 ()

(gdb) disas

Dump of assembler code for function func4:

=> 0x0000000000400fdb <+0>: push %rbx

0x0000000000400fdc <+1>: mov %edx,%eax

0x0000000000400fde <+3>: sub %esi,%eax

0x0000000000400fe0 <+5>: mov %eax,%ebx

0x0000000000400fe2 <+7>: shr $0x1f,%ebx

0x0000000000400fe5 <+10>: add %ebx,%eax

0x0000000000400fe7 <+12>: sar %eax

0x0000000000400fe9 <+14>: lea (%rax,%rsi,1),%ebx

0x0000000000400fec <+17>: cmp %edi,%ebx

0x0000000000400fee <+19>: jle 0x400ffc <func4+33>

0x0000000000400ff0 <+21>: lea -0x1(%rbx),%edx

0x0000000000400ff3 <+24>: callq 0x400fdb <func4>

0x0000000000400ff8 <+29>: add %ebx,%eax

0x0000000000400ffa <+31>: jmp 0x40100c <func4+49>

0x0000000000400ffc <+33>: mov %ebx,%eax

0x0000000000400ffe <+35>: cmp %edi,%ebx

0x0000000000401000 <+37>: jge 0x40100c <func4+49>

0x0000000000401002 <+39>: lea 0x1(%rbx),%esi

0x0000000000401005 <+42>: callq 0x400fdb <func4>

0x000000000040100a <+47>: add %ebx,%eax

0x000000000040100c <+49>: pop %rbx

0x000000000040100d <+50>: retq

End of assembler dump.

(gdb) u\*0x0000000000400fec

0x0000000000400fec in func4 ()

(gdb) disas

Dump of assembler code for function func4:

0x0000000000400fdb <+0>: push %rbx

0x0000000000400fdc <+1>: mov %edx,%eax

0x0000000000400fde <+3>: sub %esi,%eax

0x0000000000400fe0 <+5>: mov %eax,%ebx

0x0000000000400fe2 <+7>: shr $0x1f,%ebx

0x0000000000400fe5 <+10>: add %ebx,%eax

0x0000000000400fe7 <+12>: sar %eax

0x0000000000400fe9 <+14>: lea (%rax,%rsi,1),%ebx

=> 0x0000000000400fec <+17>: cmp %edi,%ebx

0x0000000000400fee <+19>: jle 0x400ffc <func4+33>

0x0000000000400ff0 <+21>: lea -0x1(%rbx),%edx

0x0000000000400ff3 <+24>: callq 0x400fdb <func4>

0x0000000000400ff8 <+29>: add %ebx,%eax

0x0000000000400ffa <+31>: jmp 0x40100c <func4+49>

0x0000000000400ffc <+33>: mov %ebx,%eax

0x0000000000400ffe <+35>: cmp %edi,%ebx

0x0000000000401000 <+37>: jge 0x40100c <func4+49>

0x0000000000401002 <+39>: lea 0x1(%rbx),%esi

0x0000000000401005 <+42>: callq 0x400fdb <func4>

0x000000000040100a <+47>: add %ebx,%eax

0x000000000040100c <+49>: pop %rbx

0x000000000040100d <+50>: retq

End of assembler dump.

(gdb) i r

rax 0x7 7

rbx 0x7 7

rcx 0x0 0

rdx 0xe 14

rsi 0x0 0

rdi 0x8 8

rbp 0x0 0x0

rsp 0x7ffffffee280 0x7ffffffee280

r8 0xffffffff 4294967295

r9 0x0 0

r10 0x7fffff74eac0 140737479240384

r11 0x0 0

r12 0x400c60 4197472

r13 0x7ffffffee3a0 140737488282528

r14 0x0 0

r15 0x0 0

rip 0x400fec 0x400fec <func4+17>

eflags 0x202 [ 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 func4:

0x0000000000400fdb <+0>: push %rbx

0x0000000000400fdc <+1>: mov %edx,%eax

0x0000000000400fde <+3>: sub %esi,%eax

0x0000000000400fe0 <+5>: mov %eax,%ebx

0x0000000000400fe2 <+7>: shr $0x1f,%ebx

0x0000000000400fe5 <+10>: add %ebx,%eax

0x0000000000400fe7 <+12>: sar %eax

0x0000000000400fe9 <+14>: lea (%rax,%rsi,1),%ebx

=> 0x0000000000400fec <+17>: cmp %edi,%ebx

0x0000000000400fee <+19>: jle 0x400ffc <func4+33> //%ebx <= %edi, so jump instruction

0x0000000000400ff0 <+21>: lea -0x1(%rbx),%edx

0x0000000000400ff3 <+24>: callq 0x400fdb <func4>

0x0000000000400ff8 <+29>: add %ebx,%eax

0x0000000000400ffa <+31>: jmp 0x40100c <func4+49>

0x0000000000400ffc <+33>: mov %ebx,%eax

0x0000000000400ffe <+35>: cmp %edi,%ebx

0x0000000000401000 <+37>: jge 0x40100c <func4+49>

0x0000000000401002 <+39>: lea 0x1(%rbx),%esi

0x0000000000401005 <+42>: callq 0x400fdb <func4>

0x000000000040100a <+47>: add %ebx,%eax

0x000000000040100c <+49>: pop %rbx

0x000000000040100d <+50>: retq

End of assembler dump.

(gdb) ni

0x0000000000400fee in func4 ()

(gdb) ni

0x0000000000400ffc in func4 ()

(gdb) disas

Dump of assembler code for function func4:

0x0000000000400fdb <+0>: push %rbx

0x0000000000400fdc <+1>: mov %edx,%eax

0x0000000000400fde <+3>: sub %esi,%eax

0x0000000000400fe0 <+5>: mov %eax,%ebx

0x0000000000400fe2 <+7>: shr $0x1f,%ebx

0x0000000000400fe5 <+10>: add %ebx,%eax

0x0000000000400fe7 <+12>: sar %eax

0x0000000000400fe9 <+14>: lea (%rax,%rsi,1),%ebx

0x0000000000400fec <+17>: cmp %edi,%ebx

0x0000000000400fee <+19>: jle 0x400ffc <func4+33>

0x0000000000400ff0 <+21>: lea -0x1(%rbx),%edx

0x0000000000400ff3 <+24>: callq 0x400fdb <func4>

0x0000000000400ff8 <+29>: add %ebx,%eax

0x0000000000400ffa <+31>: jmp 0x40100c <func4+49>

=> 0x0000000000400ffc <+33>: mov %ebx,%eax //%ebx=%eax

0x0000000000400ffe <+35>: cmp %edi,%ebx

0x0000000000401000 <+37>: jge 0x40100c <func4+49>

0x0000000000401002 <+39>: lea 0x1(%rbx),%esi

0x0000000000401005 <+42>: callq 0x400fdb <func4>

0x000000000040100a <+47>: add %ebx,%eax

0x000000000040100c <+49>: pop %rbx

0x000000000040100d <+50>: retq

End of assembler dump.

(gdb) ni

0x0000000000400ffe in func4 ()

(gdb) disas

Dump of assembler code for function func4:

0x0000000000400fdb <+0>: push %rbx

0x0000000000400fdc <+1>: mov %edx,%eax

0x0000000000400fde <+3>: sub %esi,%eax

0x0000000000400fe0 <+5>: mov %eax,%ebx

0x0000000000400fe2 <+7>: shr $0x1f,%ebx

0x0000000000400fe5 <+10>: add %ebx,%eax

0x0000000000400fe7 <+12>: sar %eax

0x0000000000400fe9 <+14>: lea (%rax,%rsi,1),%ebx

0x0000000000400fec <+17>: cmp %edi,%ebx

0x0000000000400fee <+19>: jle 0x400ffc <func4+33>

0x0000000000400ff0 <+21>: lea -0x1(%rbx),%edx

0x0000000000400ff3 <+24>: callq 0x400fdb <func4>

0x0000000000400ff8 <+29>: add %ebx,%eax

0x0000000000400ffa <+31>: jmp 0x40100c <func4+49>

0x0000000000400ffc <+33>: mov %ebx,%eax

=> 0x0000000000400ffe <+35>: cmp %edi,%ebx // %ebx >= %edi, jump instruction

0x0000000000401000 <+37>: jge 0x40100c <func4+49>

0x0000000000401002 <+39>: lea 0x1(%rbx),%esi

0x0000000000401005 <+42>: callq 0x400fdb <func4>

0x000000000040100a <+47>: add %ebx,%eax

0x000000000040100c <+49>: pop %rbx

0x000000000040100d <+50>: retq

End of assembler dump.

(gdb) i r

rax 0x7 7

rbx 0x7 7

rcx 0x0 0

rdx 0xe 14

rsi 0x0 0

rdi 0x8 8

rbp 0x0 0x0

rsp 0x7ffffffee280 0x7ffffffee280

r8 0xffffffff 4294967295

r9 0x0 0

r10 0x7fffff74eac0 140737479240384

r11 0x0 0

r12 0x400c60 4197472

r13 0x7ffffffee3a0 140737488282528

r14 0x0 0

r15 0x0 0

rip 0x400ffe 0x400ffe <func4+35>

eflags 0x297 [ CF PF AF SF IF ]

cs 0x33 51

ss 0x2b 43

ds 0x0 0

es 0x0 0

fs 0x0 0

gs 0x0 0

(gdb) ni

0x0000000000401000 in func4 ()

(gdb) ni

0x0000000000401002 in func4 ()

(gdb) ni

0x0000000000401005 in func4 ()

(gdb) ni

0x000000000040100a in func4 ()

(gdb) disas

Dump of assembler code for function func4:

0x0000000000400fdb <+0>: push %rbx

0x0000000000400fdc <+1>: mov %edx,%eax

0x0000000000400fde <+3>: sub %esi,%eax

0x0000000000400fe0 <+5>: mov %eax,%ebx

0x0000000000400fe2 <+7>: shr $0x1f,%ebx

0x0000000000400fe5 <+10>: add %ebx,%eax

0x0000000000400fe7 <+12>: sar %eax

0x0000000000400fe9 <+14>: lea (%rax,%rsi,1),%ebx

0x0000000000400fec <+17>: cmp %edi,%ebx

0x0000000000400fee <+19>: jle 0x400ffc <func4+33>

0x0000000000400ff0 <+21>: lea -0x1(%rbx),%edx

0x0000000000400ff3 <+24>: callq 0x400fdb <func4>

0x0000000000400ff8 <+29>: add %ebx,%eax

0x0000000000400ffa <+31>: jmp 0x40100c <func4+49>

0x0000000000400ffc <+33>: mov %ebx,%eax

0x0000000000400ffe <+35>: cmp %edi,%ebx

0x0000000000401000 <+37>: jge 0x40100c <func4+49>

0x0000000000401002 <+39>: lea 0x1(%rbx),%esi

0x0000000000401005 <+42>: callq 0x400fdb <func4>

=> 0x000000000040100a <+47>: add %ebx,%eax

0x000000000040100c <+49>: pop %rbx

0x000000000040100d <+50>: retq

End of assembler dump.

(gdb) ni

0x000000000040100c in func4 ()

(gdb) ni

0x000000000040100d in func4 ()

(gdb) disas

Dump of assembler code for function func4:

0x0000000000400fdb <+0>: push %rbx

0x0000000000400fdc <+1>: mov %edx,%eax

0x0000000000400fde <+3>: sub %esi,%eax

0x0000000000400fe0 <+5>: mov %eax,%ebx

0x0000000000400fe2 <+7>: shr $0x1f,%ebx

0x0000000000400fe5 <+10>: add %ebx,%eax

0x0000000000400fe7 <+12>: sar %eax

0x0000000000400fe9 <+14>: lea (%rax,%rsi,1),%ebx

0x0000000000400fec <+17>: cmp %edi,%ebx

0x0000000000400fee <+19>: jle 0x400ffc <func4+33>

0x0000000000400ff0 <+21>: lea -0x1(%rbx),%edx

0x0000000000400ff3 <+24>: callq 0x400fdb <func4>

0x0000000000400ff8 <+29>: add %ebx,%eax

0x0000000000400ffa <+31>: jmp 0x40100c <func4+49>

0x0000000000400ffc <+33>: mov %ebx,%eax

0x0000000000400ffe <+35>: cmp %edi,%ebx

0x0000000000401000 <+37>: jge 0x40100c <func4+49>

0x0000000000401002 <+39>: lea 0x1(%rbx),%esi

0x0000000000401005 <+42>: callq 0x400fdb <func4>

0x000000000040100a <+47>: add %ebx,%eax

0x000000000040100c <+49>: pop %rbx

=> 0x000000000040100d <+50>: retq

End of assembler dump.

(gdb) ni

0x0000000000401056 in phase\_4 ()

(gdb) disas

Dump of assembler code for function phase\_4:

0x000000000040100e <+0>: sub $0x18,%rsp

0x0000000000401012 <+4>: mov %fs:0x28,%rax

0x000000000040101b <+13>: mov %rax,0x8(%rsp)

0x0000000000401020 <+18>: xor %eax,%eax

0x0000000000401022 <+20>: lea 0x4(%rsp),%rcx

0x0000000000401027 <+25>: mov %rsp,%rdx

0x000000000040102a <+28>: mov $0x4025cf,%esi

0x000000000040102f <+33>: callq 0x400bb0 <\_\_isoc99\_sscanf@plt>

0x0000000000401034 <+38>: cmp $0x2,%eax

0x0000000000401037 <+41>: jne 0x40103f <phase\_4+49>

0x0000000000401039 <+43>: cmpl $0xe,(%rsp)

0x000000000040103d <+47>: jbe 0x401044 <phase\_4+54>

0x000000000040103f <+49>: callq 0x40143d <explode\_bomb>

0x0000000000401044 <+54>: mov $0xe,%edx

0x0000000000401049 <+59>: mov $0x0,%esi

0x000000000040104e <+64>: mov (%rsp),%edi

0x0000000000401051 <+67>: callq 0x400fdb <func4>

=> 0x0000000000401056 <+72>: cmp $0x23,%eax

0x0000000000401059 <+75>: jne 0x401062 <phase\_4+84>

0x000000000040105b <+77>: cmpl $0x23,0x4(%rsp)

0x0000000000401060 <+82>: je 0x401067 <phase\_4+89>

0x0000000000401062 <+84>: callq 0x40143d <explode\_bomb>

0x0000000000401067 <+89>: mov 0x8(%rsp),%rax

0x000000000040106c <+94>: xor %fs:0x28,%rax

0x0000000000401075 <+103>: je 0x40107c <phase\_4+110>

0x0000000000401077 <+105>: callq 0x400b00 <\_\_stack\_chk\_fail@plt>

0x000000000040107c <+110>: add $0x18,%rsp

0x0000000000401080 <+114>: retq

End of assembler dump.

(gdb) i r

rax 0x23 35

rbx 0x4021f0 4202992

rcx 0x0 0

rdx 0x8 8

rsi 0x8 8

rdi 0x8 8

rbp 0x0 0x0

rsp 0x7ffffffee290 0x7ffffffee290

r8 0xffffffff 4294967295

r9 0x0 0

r10 0x7fffff74eac0 140737479240384

r11 0x0 0

r12 0x400c60 4197472

r13 0x7ffffffee3a0 140737488282528

r14 0x0 0

r15 0x0 0

rip 0x401056 0x401056 <phase\_4+72>

eflags 0x212 [ AF 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\_4:

0x000000000040100e <+0>: sub $0x18,%rsp

0x0000000000401012 <+4>: mov %fs:0x28,%rax

0x000000000040101b <+13>: mov %rax,0x8(%rsp)

0x0000000000401020 <+18>: xor %eax,%eax

0x0000000000401022 <+20>: lea 0x4(%rsp),%rcx

0x0000000000401027 <+25>: mov %rsp,%rdx

0x000000000040102a <+28>: mov $0x4025cf,%esi

0x000000000040102f <+33>: callq 0x400bb0 <\_\_isoc99\_sscanf@plt>

0x0000000000401034 <+38>: cmp $0x2,%eax

0x0000000000401037 <+41>: jne 0x40103f <phase\_4+49>

0x0000000000401039 <+43>: cmpl $0xe,(%rsp)

0x000000000040103d <+47>: jbe 0x401044 <phase\_4+54>

0x000000000040103f <+49>: callq 0x40143d <explode\_bomb>

0x0000000000401044 <+54>: mov $0xe,%edx

0x0000000000401049 <+59>: mov $0x0,%esi

0x000000000040104e <+64>: mov (%rsp),%edi

0x0000000000401051 <+67>: callq 0x400fdb <func4>

=> 0x0000000000401056 <+72>: cmp $0x23,%eax

0x0000000000401059 <+75>: jne 0x401062 <phase\_4+84>

0x000000000040105b <+77>: cmpl $0x23,0x4(%rsp)

0x0000000000401060 <+82>: je 0x401067 <phase\_4+89>

0x0000000000401062 <+84>: callq 0x40143d <explode\_bomb>

0x0000000000401067 <+89>: mov 0x8(%rsp),%rax

0x000000000040106c <+94>: xor %fs:0x28,%rax

0x0000000000401075 <+103>: je 0x40107c <phase\_4+110>

0x0000000000401077 <+105>: callq 0x400b00 <\_\_stack\_chk\_fail@plt>

0x000000000040107c <+110>: add $0x18,%rsp

0x0000000000401080 <+114>: retq

End of assembler dump.

(gdb) ni

0x0000000000401059 in phase\_4 ()

(gdb) ni

0x000000000040105b in phase\_4 ()

(gdb) disas

Dump of assembler code for function phase\_4:

0x000000000040100e <+0>: sub $0x18,%rsp

0x0000000000401012 <+4>: mov %fs:0x28,%rax

0x000000000040101b <+13>: mov %rax,0x8(%rsp)

0x0000000000401020 <+18>: xor %eax,%eax

0x0000000000401022 <+20>: lea 0x4(%rsp),%rcx

0x0000000000401027 <+25>: mov %rsp,%rdx

0x000000000040102a <+28>: mov $0x4025cf,%esi

0x000000000040102f <+33>: callq 0x400bb0 <\_\_isoc99\_sscanf@plt>

0x0000000000401034 <+38>: cmp $0x2,%eax

0x0000000000401037 <+41>: jne 0x40103f <phase\_4+49>

0x0000000000401039 <+43>: cmpl $0xe,(%rsp)

0x000000000040103d <+47>: jbe 0x401044 <phase\_4+54>

0x000000000040103f <+49>: callq 0x40143d <explode\_bomb>

0x0000000000401044 <+54>: mov $0xe,%edx

0x0000000000401049 <+59>: mov $0x0,%esi

0x000000000040104e <+64>: mov (%rsp),%edi

0x0000000000401051 <+67>: callq 0x400fdb <func4>

0x0000000000401056 <+72>: cmp $0x23,%eax

0x0000000000401059 <+75>: jne 0x401062 <phase\_4+84>

=> 0x000000000040105b <+77>: cmpl $0x23,0x4(%rsp)

0x0000000000401060 <+82>: je 0x401067 <phase\_4+89>

0x0000000000401062 <+84>: callq 0x40143d <explode\_bomb> // if 0x4(%rsp) is NOT equal to $0x23 jump to explode bomb

0x0000000000401067 <+89>: mov 0x8(%rsp),%rax

0x000000000040106c <+94>: xor %fs:0x28,%rax

0x0000000000401075 <+103>: je 0x40107c <phase\_4+110>

0x0000000000401077 <+105>: callq 0x400b00 <\_\_stack\_chk\_fail@plt>

0x000000000040107c <+110>: add $0x18,%rsp

0x0000000000401080 <+114>: retq

End of assembler dump.

(gdb) x/d 0x4+$rsp

0x7ffffffee294: 3

(gdb) run

The program being debugged has been started already.

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

Starting program: /mnt/c/Users/Hp/OneDrive/Documents/3rd Year/Computer System I ITS304/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!

2 409

Halfway there!

8 35

Breakpoint 1, 0x000000000040100e in phase\_4 ()

(gdb) disas

Dump of assembler code for function phase\_4:

=> 0x000000000040100e <+0>: sub $0x18,%rsp

0x0000000000401012 <+4>: mov %fs:0x28,%rax

0x000000000040101b <+13>: mov %rax,0x8(%rsp)

0x0000000000401020 <+18>: xor %eax,%eax

0x0000000000401022 <+20>: lea 0x4(%rsp),%rcx

0x0000000000401027 <+25>: mov %rsp,%rdx

0x000000000040102a <+28>: mov $0x4025cf,%esi

0x000000000040102f <+33>: callq 0x400bb0 <\_\_isoc99\_sscanf@plt>

0x0000000000401034 <+38>: cmp $0x2,%eax

0x0000000000401037 <+41>: jne 0x40103f <phase\_4+49>

0x0000000000401039 <+43>: cmpl $0xe,(%rsp)

0x000000000040103d <+47>: jbe 0x401044 <phase\_4+54>

0x000000000040103f <+49>: callq 0x40143d <explode\_bomb>

0x0000000000401044 <+54>: mov $0xe,%edx

0x0000000000401049 <+59>: mov $0x0,%esi

0x000000000040104e <+64>: mov (%rsp),%edi

0x0000000000401051 <+67>: callq 0x400fdb <func4>

0x0000000000401056 <+72>: cmp $0x23,%eax

0x0000000000401059 <+75>: jne 0x401062 <phase\_4+84>

0x000000000040105b <+77>: cmpl $0x23,0x4(%rsp)

0x0000000000401060 <+82>: je 0x401067 <phase\_4+89>

0x0000000000401062 <+84>: callq 0x40143d <explode\_bomb>

0x0000000000401067 <+89>: mov 0x8(%rsp),%rax

0x000000000040106c <+94>: xor %fs:0x28,%rax

0x0000000000401075 <+103>: je 0x40107c <phase\_4+110>

0x0000000000401077 <+105>: callq 0x400b00 <\_\_stack\_chk\_fail@plt>

0x000000000040107c <+110>: add $0x18,%rsp

0x0000000000401080 <+114>: retq

End of assembler dump.

(gdb) u\*0x000000000040105b

0x000000000040105b in phase\_4 ()

(gdb) disas

Dump of assembler code for function phase\_4:

0x000000000040100e <+0>: sub $0x18,%rsp

0x0000000000401012 <+4>: mov %fs:0x28,%rax

0x000000000040101b <+13>: mov %rax,0x8(%rsp)

0x0000000000401020 <+18>: xor %eax,%eax

0x0000000000401022 <+20>: lea 0x4(%rsp),%rcx

0x0000000000401027 <+25>: mov %rsp,%rdx

0x000000000040102a <+28>: mov $0x4025cf,%esi

0x000000000040102f <+33>: callq 0x400bb0 <\_\_isoc99\_sscanf@plt>

0x0000000000401034 <+38>: cmp $0x2,%eax

0x0000000000401037 <+41>: jne 0x40103f <phase\_4+49>

0x0000000000401039 <+43>: cmpl $0xe,(%rsp)

0x000000000040103d <+47>: jbe 0x401044 <phase\_4+54>

0x000000000040103f <+49>: callq 0x40143d <explode\_bomb>

0x0000000000401044 <+54>: mov $0xe,%edx

0x0000000000401049 <+59>: mov $0x0,%esi

0x000000000040104e <+64>: mov (%rsp),%edi

0x0000000000401051 <+67>: callq 0x400fdb <func4>

0x0000000000401056 <+72>: cmp $0x23,%eax

0x0000000000401059 <+75>: jne 0x401062 <phase\_4+84>

=> 0x000000000040105b <+77>: cmpl $0x23,0x4(%rsp)

0x0000000000401060 <+82>: je 0x401067 <phase\_4+89>

0x0000000000401062 <+84>: callq 0x40143d <explode\_bomb>

0x0000000000401067 <+89>: mov 0x8(%rsp),%rax

0x000000000040106c <+94>: xor %fs:0x28,%rax

0x0000000000401075 <+103>: je 0x40107c <phase\_4+110>

0x0000000000401077 <+105>: callq 0x400b00 <\_\_stack\_chk\_fail@plt>

0x000000000040107c <+110>: add $0x18,%rsp

0x0000000000401080 <+114>: retq

End of assembler dump.

(gdb) x/d 0x4+$rsp

0x7ffffffee294: 35

(gdb) disas

Dump of assembler code for function phase\_4:

0x000000000040100e <+0>: sub $0x18,%rsp

0x0000000000401012 <+4>: mov %fs:0x28,%rax

0x000000000040101b <+13>: mov %rax,0x8(%rsp)

0x0000000000401020 <+18>: xor %eax,%eax

0x0000000000401022 <+20>: lea 0x4(%rsp),%rcx

0x0000000000401027 <+25>: mov %rsp,%rdx

0x000000000040102a <+28>: mov $0x4025cf,%esi

0x000000000040102f <+33>: callq 0x400bb0 <\_\_isoc99\_sscanf@plt>

0x0000000000401034 <+38>: cmp $0x2,%eax

0x0000000000401037 <+41>: jne 0x40103f <phase\_4+49>

0x0000000000401039 <+43>: cmpl $0xe,(%rsp)

0x000000000040103d <+47>: jbe 0x401044 <phase\_4+54>

0x000000000040103f <+49>: callq 0x40143d <explode\_bomb>

0x0000000000401044 <+54>: mov $0xe,%edx

0x0000000000401049 <+59>: mov $0x0,%esi

0x000000000040104e <+64>: mov (%rsp),%edi

0x0000000000401051 <+67>: callq 0x400fdb <func4>

0x0000000000401056 <+72>: cmp $0x23,%eax

0x0000000000401059 <+75>: jne 0x401062 <phase\_4+84>

=> 0x000000000040105b <+77>: cmpl $0x23,0x4(%rsp)

0x0000000000401060 <+82>: je 0x401067 <phase\_4+89>

0x0000000000401062 <+84>: callq 0x40143d <explode\_bomb>

0x0000000000401067 <+89>: mov 0x8(%rsp),%rax //%rax = 0x8(%rsp)

0x000000000040106c <+94>: xor %fs:0x28,%rax // %rax = %fs ^ 0x28

0x0000000000401075 <+103>: je 0x40107c <phase\_4+110>

0x0000000000401077 <+105>: callq 0x400b00 <\_\_stack\_chk\_fail@plt>

0x000000000040107c <+110>: add $0x18,%rsp // %rsp = $0x18 + %rsp

0x0000000000401080 <+114>: retq

End of assembler dump.

(gdb) ni

0x0000000000401060 in phase\_4 ()

(gdb) ni

0x0000000000401067 in phase\_4 ()

(gdb) disas

Dump of assembler code for function phase\_4:

0x000000000040100e <+0>: sub $0x18,%rsp

0x0000000000401012 <+4>: mov %fs:0x28,%rax

0x000000000040101b <+13>: mov %rax,0x8(%rsp)

0x0000000000401020 <+18>: xor %eax,%eax

0x0000000000401022 <+20>: lea 0x4(%rsp),%rcx

0x0000000000401027 <+25>: mov %rsp,%rdx

0x000000000040102a <+28>: mov $0x4025cf,%esi

0x000000000040102f <+33>: callq 0x400bb0 <\_\_isoc99\_sscanf@plt>

0x0000000000401034 <+38>: cmp $0x2,%eax

0x0000000000401037 <+41>: jne 0x40103f <phase\_4+49>

0x0000000000401039 <+43>: cmpl $0xe,(%rsp)

0x000000000040103d <+47>: jbe 0x401044 <phase\_4+54>

0x000000000040103f <+49>: callq 0x40143d <explode\_bomb>

0x0000000000401044 <+54>: mov $0xe,%edx

0x0000000000401049 <+59>: mov $0x0,%esi

0x000000000040104e <+64>: mov (%rsp),%edi

0x0000000000401051 <+67>: callq 0x400fdb <func4>

0x0000000000401056 <+72>: cmp $0x23,%eax

0x0000000000401059 <+75>: jne 0x401062 <phase\_4+84>

0x000000000040105b <+77>: cmpl $0x23,0x4(%rsp)

0x0000000000401060 <+82>: je 0x401067 <phase\_4+89>

0x0000000000401062 <+84>: callq 0x40143d <explode\_bomb>

=> 0x0000000000401067 <+89>: mov 0x8(%rsp),%rax

0x000000000040106c <+94>: xor %fs:0x28,%rax

0x0000000000401075 <+103>: je 0x40107c <phase\_4+110>

0x0000000000401077 <+105>: callq 0x400b00 <\_\_stack\_chk\_fail@plt>

0x000000000040107c <+110>: add $0x18,%rsp

0x0000000000401080 <+114>: retq

End of assembler dump.

(gdb) ni

0x000000000040106c in phase\_4 ()

(gdb) ni

0x0000000000401075 in phase\_4 ()

(gdb) ni

0x000000000040107c in phase\_4 ()

(gdb) ni

0x0000000000401080 in phase\_4 ()

(gdb) ni

main (argc=<optimized out>, argv=<optimized out>) at bomb.c:96

96 phase\_defused(); //Phase\_4 has been diffused

(gdb) d

Delete all breakpoints? (y or n) y

(gdb) run

The program being debugged has been started already.

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

Starting program: /mnt/c/Users/Hp/OneDrive/Documents/3rd Year/Computer System I ITS304/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!

2 409

Halfway there!

8 35 //Correct input of Phase\_4 so the phase has been diffused

So you got that one. Try this one.