Dump of assembler code for function phase\_4:

=> 0x0000000000400f75 <+0>: sub $0x18,%rsp

0x0000000000400f79 <+4>: lea 0x8(%rsp),%rcx

0x0000000000400f7e <+9>: lea 0xc(%rsp),%rdx

0x0000000000400f83 <+14>: mov $0x40241f,%esi

0x0000000000400f88 <+19>: mov $0x0,%eax

0x0000000000400f8d <+24>: callq 0x400ba0 <\_\_isoc99\_sscanf@plt>

0x0000000000400f92 <+29>: cmp $0x2,%eax

0x0000000000400f95 <+32>: jne 0x400f9e <phase\_4+41>

0x0000000000400f97 <+34>: cmpl $0xe,0xc(%rsp)

0x0000000000400f9c <+39>: jbe 0x400fa3 <phase\_4+46>

0x0000000000400f9e <+41>: callq 0x40135f <explode\_bomb>

0x0000000000400fa3 <+46>: mov $0xe,%edx

0x0000000000400fa8 <+51>: mov $0x0,%esi

0x0000000000400fad <+56>: mov 0xc(%rsp),%edi

0x0000000000400fb1 <+60>: callq 0x400f42 <func4>

0x0000000000400fb6 <+65>: cmp $0xb,%eax

0x0000000000400fb9 <+68>: jne 0x400fc2 <phase\_4+77>

0x0000000000400fbb <+70>: cmpl $0xb,0x8(%rsp)

0x0000000000400fc0 <+75>: je 0x400fc7 <phase\_4+82>

0x0000000000400fc2 <+77>: callq 0x40135f <explode\_bomb>

0x0000000000400fc7 <+82>: add $0x18,%rsp

0x0000000000400fcb <+86>: retq

End of assembler dump.

(gdb) n

Single stepping until exit from function phase\_4,

which has no line number information.

Breakpoint 2, 0x0000000000400f42 in func4 ()

(gdb) disas

Dump of assembler code for function func4: edx 14 esi 0 edi fisrt

=> 0x0000000000400f42 <+0>: push %rbx

0x0000000000400f43 <+1>: mov %edx,%eax eax = edx

0x0000000000400f45 <+3>: sub %esi,%eax

0x0000000000400f47 <+5>: mov %eax,%ebx ebx = eax = edx

0x0000000000400f49 <+7>: shr $0x1f,%ebx ebx >> 31 // 마이너스 판별? 마이너스면 1이 나올 것 펑션4 들어오자마자는 ebx= 14이다 //31번하면 0임

0x0000000000400f4c <+10>: add %ebx,%eax 14+0 = 14 = eax

0x0000000000400f4e <+12>: sar %eax eax>>1 최상위 bit유지 +- 유지한다는말

0x0000000000400f50 <+14>: lea (%rax,%rsi,1),%ebx ebx = rax+ rsi = 14 + 0

0x0000000000400f53 <+17>: cmp %edi,%ebx

0x0000000000400f55 <+19>: jle 0x400f63 <func4+33> edi <= ebx = jump

0x0000000000400f57 <+21>: lea -0x1(%rbx),%edx

0x0000000000400f5a <+24>: callq 0x400f42 <func4>

0x0000000000400f5f <+29>: add %ebx,%eax

0x0000000000400f61 <+31>: jmp 0x400f73 <func4+49>

0x0000000000400f63 <+33>: mov %ebx,%eax

0x0000000000400f65 <+35>: cmp %edi,%ebx

0x0000000000400f67 <+37>: jge 0x400f73 <func4+49>

0x0000000000400f69 <+39>: lea 0x1(%rbx),%esi

0x0000000000400f6c <+42>: callq 0x400f42 <func4>

0x0000000000400f71 <+47>: add %ebx,%eax

0x0000000000400f73 <+49>: pop %rbx

0x0000000000400f74 <+50>: retq