Dump of assembler code for function phase\_2:

0x0000000000400e49 <+0>: push %rbp

0x0000000000400e4a <+1>: push %rbx

0x0000000000400e4b <+2>: sub $0x28,%rsp

0x0000000000400e4f <+6>: mov %rsp,%rsi

0x0000000000400e52 <+9>: callq 0x4014bc <read\_six\_numbers>

0x0000000000400e57 <+14>: cmpl $0x0,(%rsp)

The first number is 0, otherwise it gonna explode

0x0000000000400e5b <+18>: jne 0x400e64 <phase\_2+27>

0x0000000000400e5d <+20>: cmpl $0x1,0x4(%rsp)

The second number is 1, otherwise it gonna explode

0x0000000000400e62 <+25>: je 0x400e69 <phase\_2+32>

0x0000000000400e64 <+27>: callq 0x401486 <explode\_bomb>

0x0000000000400e69 <+32>: mov %rsp,%rbx

0x0000000000400e6c <+35>: lea 0x10(%rsp),%rbp

0x0000000000400e71 <+40>: mov 0x4(%rbx),%eax

0x0000000000400e74 <+43>: add (%rbx),%eax

0x0000000000400e76 <+45>: cmp %eax,0x8(%rbx)

0x0000000000400e79 <+48>: je 0x400e80 <phase\_2+55>

0x0000000000400e7b <+50>: callq 0x401486 <explode\_bomb>

0x0000000000400e80 <+55>: add $0x4,%rbx

0x0000000000400e84 <+59>: cmp %rbp,%rbx

0x0000000000400e87 <+62>: jne 0x400e71 <phase\_2+40>

0x0000000000400e89 <+64>: add $0x28,%rsp

0x0000000000400e8d <+68>: pop %rbx

0x0000000000400e8e <+69>: pop %rbp

0x0000000000400e8f <+70>: retq

End of assembler dump.

Fibonacci sequence