Dump of assembler code for function phase\_3:

0x0000000000400e90 <+0>: sub $0x18,%rsp

0x0000000000400e94 <+4>: lea 0x8(%rsp),%r8

0x0000000000400e99 <+9>: lea 0x7(%rsp),%rcx

0x0000000000400e9e <+14>: lea 0xc(%rsp),%rdx

0x0000000000400ea3 <+19>: mov $0x402406,%esi

0x0000000000400ea8 <+24>: mov $0x0,%eax

0x0000000000400ead <+29>: callq 0x400b50 <\_\_isoc99\_sscanf@plt>

0x0000000000400eb2 <+34>: cmp $0x2,%eax

0x0000000000400eb5 <+37>: jg 0x400ebc <phase\_3+44>

0x0000000000400eb7 <+39>: callq 0x401486 <explode\_bomb>

0x0000000000400ebc <+44>: cmpl $0x7,0xc(%rsp)

0x0000000000400ec1 <+49>: ja 0x400fc3 <phase\_3+307>

Jump above 所以我知道了第一个数字要小于7

0x0000000000400ec7 <+55>: mov 0xc(%rsp),%eax 你输入的第一个数传输给eax

0x0000000000400ecb <+59>: jmpq \*0x402420(,%rax,8)

在这里跳转

rax 如果是 0 就跳到 p/x \*0x402420

rax 如果是 1 就跳到 p/x \*0x402428

0x0000000000400ed2 <+66>: mov $0x67,%eax. eax 等于 0x67

0x0000000000400ed7 <+71>: cmpl $0x19c,0x8(%rsp)。第三个数是 0x19c

0x0000000000400edf <+79>: je 0x400fcd <phase\_3+317>

0x0000000000400ee5 <+85>: callq 0x401486 <explode\_bomb>

0x0000000000400eea <+90>: mov $0x67,%eax

0x0000000000400eef <+95>: jmpq 0x400fcd <phase\_3+317>

0x0000000000400ef4 <+100>: mov $0x6c,%eax

0x0000000000400ef9 <+105>: cmpl $0x3ab,0x8(%rsp)

0x0000000000400f01 <+113>: je 0x400fcd <phase\_3+317>

0x0000000000400f07 <+119>: callq 0x401486 <explode\_bomb>

0x0000000000400f0c <+124>: mov $0x6c,%eax

0x0000000000400f11 <+129>: jmpq 0x400fcd <phase\_3+317>

0x0000000000400f16 <+134>: mov $0x6a,%eax

0x0000000000400f1b <+139>: cmpl $0xec,0x8(%rsp)

0x0000000000400f23 <+147>: je 0x400fcd <phase\_3+317>

0x0000000000400f29 <+153>: callq 0x401486 <explode\_bomb>

0x0000000000400f2e <+158>: mov $0x6a,%eax

0x0000000000400f33 <+163>: jmpq 0x400fcd <phase\_3+317>

0x0000000000400f38 <+168>: mov $0x7a,%eax

0x0000000000400f3d <+173>: cmpl $0x274,0x8(%rsp)

0x0000000000400f45 <+181>: je 0x400fcd <phase\_3+317>

0x0000000000400f4b <+187>: callq 0x401486 <explode\_bomb>

0x0000000000400f50 <+192>: mov $0x7a,%eax

0x0000000000400f55 <+197>: jmp 0x400fcd <phase\_3+317>

0x0000000000400f57 <+199>: mov $0x61,%eax

0x0000000000400f5c <+204>: cmpl $0x19a,0x8(%rsp)

0x0000000000400f64 <+212>: je 0x400fcd <phase\_3+317>

0x0000000000400f66 <+214>: callq 0x401486 <explode\_bomb>

0x0000000000400f6b <+219>: mov $0x61,%eax

0x0000000000400f70 <+224>: jmp 0x400fcd <phase\_3+317>

0x0000000000400f72 <+226>: mov $0x67,%eax

0x0000000000400f77 <+231>: cmpl $0x2e1,0x8(%rsp)

0x0000000000400f7f <+239>: je 0x400fcd <phase\_3+317>

0x0000000000400f81 <+241>: callq 0x401486 <explode\_bomb>

0x0000000000400f86 <+246>: mov $0x67,%eax

0x0000000000400f8b <+251>: jmp 0x400fcd <phase\_3+317>

0x0000000000400f8d <+253>: mov $0x6f,%eax

0x0000000000400f92 <+258>: cmpl $0x20c,0x8(%rsp)

0x0000000000400f9a <+266>: je 0x400fcd <phase\_3+317>

0x0000000000400f9c <+268>: callq 0x401486 <explode\_bomb>

0x0000000000400fa1 <+273>: mov $0x6f,%eax

0x0000000000400fa6 <+278>: jmp 0x400fcd <phase\_3+317>

0x0000000000400fa8 <+280>: mov $0x72,%eax

0x0000000000400fad <+285>: cmpl $0x3cf,0x8(%rsp)

0x0000000000400fb5 <+293>: je 0x400fcd <phase\_3+317>

0x0000000000400fb7 <+295>: callq 0x401486 <explode\_bomb>

0x0000000000400fbc <+300>: mov $0x72,%eax

0x0000000000400fc1 <+305>: jmp 0x400fcd <phase\_3+317>

0x0000000000400fc3 <+307>: callq 0x401486 <explode\_bomb>

0x0000000000400fc8 <+312>: mov $0x63,%eax

0x0000000000400fcd <+317>: cmp 0x7(%rsp),%al. eax 这里就是 al 因为是一个char 所以通过 man ascii 转换成 对应的字符 ‘g’ 所以第二个数是 g

0x0000000000400fd1 <+321>: je 0x400fd8 <phase\_3+328>

0x0000000000400fd3 <+323>: callq 0x401486 <explode\_bomb>

0x0000000000400fd8 <+328>: add $0x18,%rsp

0x0000000000400fdc <+332>: retq

End of assembler dump

x/s 402406

:”%d %c %d”

所以应该是两个整数