反汇编：

```
0x000000000040104e <+0>:     sub    $0x18,%rsp
0x0000000000401052 <+4>:     mov    %fs:0x28,%rax
0x000000000040105b <+13>:    mov    %rax,0x8(%rsp)
0x0000000000401060 <+18>:    xor    %eax,%eax
0x0000000000401062 <+20>:    lea    0x4(%rsp),%rcx
0x0000000000401067 <+25>:    mov    %rsp,%rdx
0x000000000040106a <+28>:    mov    $0x4025cf,%esi
0x000000000040106f <+33>:    callq  0x400ba0 <__isoc99_sscanf@plt>
0x0000000000401074 <+38>:    cmp    $0x1,%eax
0x0000000000401077 <+41>:    jle    0x4010d0 <phase_5+130>
0x0000000000401079 <+43>:    mov    (%rsp),%eax
0x000000000040107c <+46>:    and    $0xf,%eax
0x000000000040107f <+49>:    mov    %eax,(%rsp)
0x0000000000401082 <+52>:    cmp    $0xf,%eax
0x0000000000401085 <+55>:    je     0x4010b6 <phase_5+104>
0x0000000000401087 <+57>:    mov    $0x0,%ecx
0x000000000040108c <+62>:    mov    $0x0,%edx
0x0000000000401091 <+67>:    add    $0x1,%edx
0x0000000000401094 <+70>:    cltq
0x0000000000401096 <+72>:    mov    0x402480(,%rax,4),%eax
0x000000000040109d <+79>:    add    %eax,%ecx
0x000000000040109f <+81>:    cmp    $0xf,%eax
```

```
0x00000000004010a2 <+84>:    jne    0x401091 <phase_5+67>
0x00000000004010a4 <+86>:    movl   $0xf,(%rsp)
0x00000000004010ab <+93>:    cmp    $0x3,%edx
0x00000000004010ae <+96>:    jne    0x4010b6 <phase_5+104>
0x00000000004010b0 <+98>:    cmp    %ecx,0x4(%rsp)
0x00000000004010b4 <+102>:   je     0x4010bb <phase_5+109>
0x00000000004010b6 <+104>:   callq  0x401447 <explode_bomb>
0x00000000004010bb <+109>:   mov    0x8(%rsp),%rax
0x00000000004010c0 <+114>:   xor    %fs:0x28,%rax
0x00000000004010c9 <+123>:   jne    0x4010d7 <phase_5+137>
0x00000000004010cb <+125>:   add    $0x18,%rsp
0x00000000004010cf <+129>:   retq
0x00000000004010d0 <+130>:   callq  0x401447 <explode_bomb>
0x00000000004010d5 <+135>:   jmp    0x401079 <phase_5+43>
0x00000000004010d7 <+137>:   callq  0x400b00 <__stack_chk_fail@plt>
```

```
phase_5:
rsp -= 0x18
eax = 0
rcx = rsp + 0x4
rdx = rsp
esi = 0x4025cf  // 还是%d %d输入两个数
sccanf
if(eax <= 0x1){
    call <explode_bomb>
}
eax = *rsp  // 第一个数
eax &= 0xf  // 取二进制的后四位
*(rsp) = eax
if(eax == 0xf){ // 如果第一个数的二进制后四位是1111 bomb
```

```
    call <explode_bomb>
}
ecx = 0x0
edx = 0x0
do{
    edx += 0x1
    cltq    // 将eax符号扩展到rax
    eax = *(0x402480 + 4 * rax)
    ecx += eax
}while(eax != 0xf);
*rsp = 0xf
if(edx != 0x3){
    call <explode_bomb>
}
if(*(rsp + 0x4) == ecx){
    ...
    retq
}
```

x/16 0x402480的结果：

```
0x402480 <array.3415>:  10      2       14      7
0x402490 <array.3415+16>:       8       12      15      11
0x4024a0 <array.3415+32>:       0       4       1       13
0x4024b0 <array.3415+48>:       3       9       6       5
```

分析程序的大概意思就是输入了x y，然后根据x索引找到地址里的数，再将这个数作为地址索引找到对应存放的值，直到3此这样寻找后找到0xf，即15.

答案就是2 35，2->6->15，其中35 = 14 + 6 + 15