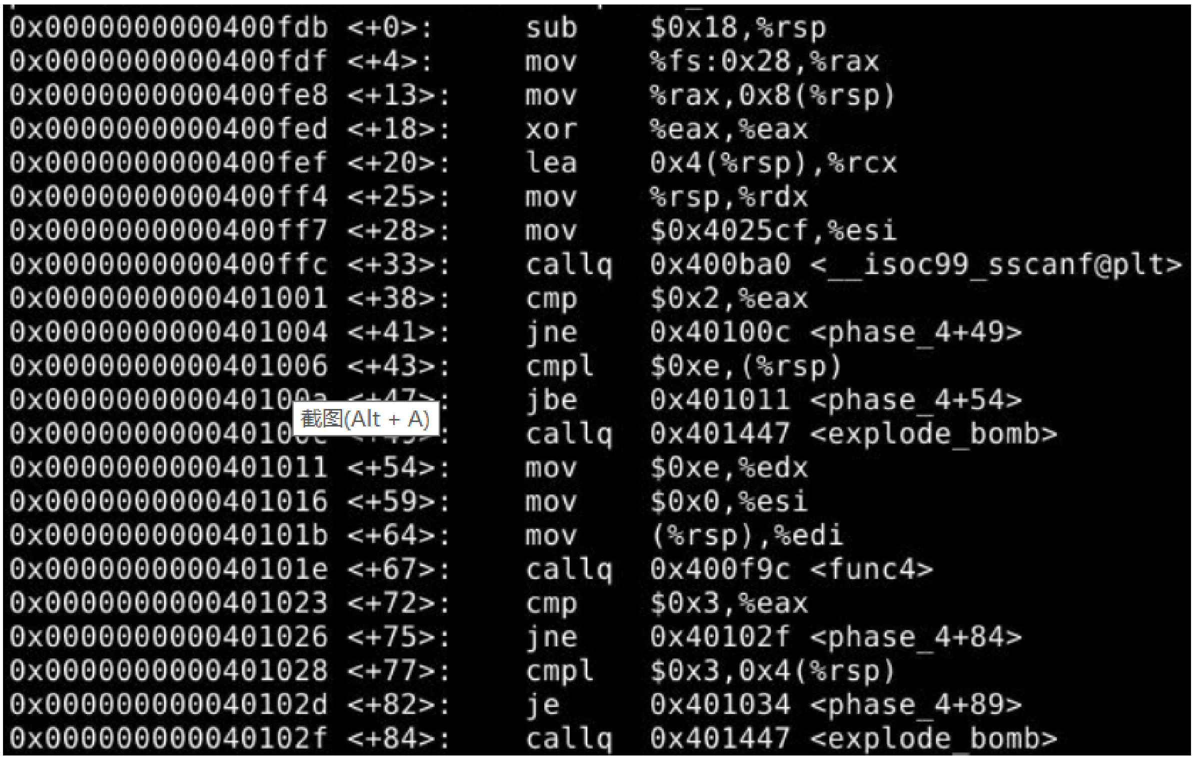
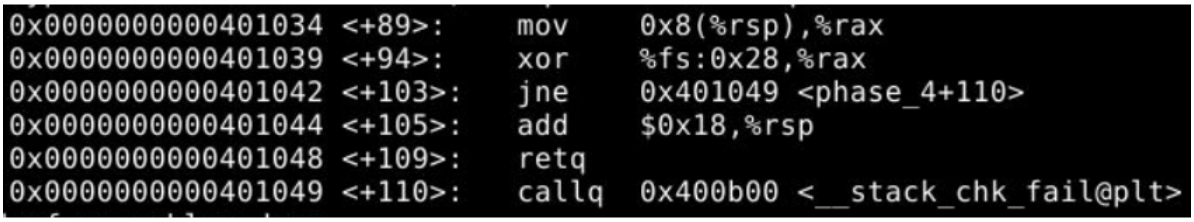
反汇编结果：



```
0x0000000000400fdb <+0>:     sub    $0x18,%rsp
0x0000000000400fdf <+4>:     mov    %fs:0x28,%rax
0x0000000000400fe8 <+13>:    mov    %rax,0x8(%rsp)
0x0000000000400fed <+18>:    xor    %eax,%eax
0x0000000000400fef <+20>:    lea    0x4(%rsp),%rcx
0x0000000000400ff4 <+25>:    mov    %rsp,%rdx
0x0000000000400ff7 <+28>:    mov    $0x4025cf,%esi
0x0000000000400ffc <+33>:    callq  0x400ba0 <__isoc99_sscanf@plt>
0x0000000000401001 <+38>:    cmp    $0x2,%eax
0x0000000000401004 <+41>:    jne    0x40100c <phase_4+49>
0x0000000000401006 <+43>:    cmpl   $0xe,(%rsp)
0x000000000040100a <+47>:    jbe    0x401011 <phase_4+54>
0x000000000040100c <+49>:    callq  0x401447 <explode_bomb>
0x0000000000401011 <+54>:    mov    $0xe,%edx
0x0000000000401016 <+59>:    mov    $0x0,%esi
0x000000000040101b <+64>:    mov    (%rsp),%edi
0x000000000040101e <+67>:    callq  0x400f9c <func4>
0x0000000000401023 <+72>:    cmp    $0x3,%eax
0x0000000000401026 <+75>:    jne    0x40102f <phase_4+84>
0x0000000000401028 <+77>:    cmpl   $0x3,0x4(%rsp)
0x000000000040102d <+82>:    je     0x401034 <phase_4+89>
0x000000000040102f <+84>:    callq  0x401447 <explode_bomb>
```



```
0x0000000000401034 <+89>:    mov    0x8(%rsp),%rax
0x0000000000401039 <+94>:    xor    %fs:0x28,%rax
0x0000000000401042 <+103>:   jne    0x401049 <phase_4+110>
0x0000000000401044 <+105>:   add    $0x18,%rsp
0x0000000000401048 <+109>:   retq
0x0000000000401049 <+110>:   callq  0x400b00 <__stack_chk_fail@plt>
```

x86-64传参规则：当参数个数<=6时，使用rdi，rsi，rdx，rcx，r8，r9

```
<phase_4>:
rsp -= 0x18
*(rsp + 0x8) = rax
eax = 0
rcx = (rsp + 0x4)
rdx = rsp
esi = 0x4025cf       (%d %d)
if(eax - 0x2 != 0){           // 说明输入的参数必须是2个
    call <explode_bomb>
}
if(*rsp - 0xe <= 0){          // 说明输入的第一个数 < 0xe = 14
    edx = 0xe
    esi = 0x0
    edi = *(rsp)
    call <func4>              // 相当于func4(*(rsp)（输入的第一个数）, 0x0, 0xe)
    if(eax - 0x3 != 0){       // 当eax==3时满足条件
        call <explode_bomb>
    }
    if(*(rsp + 0x4) - 0x3 == 0){    // 说明输入的第二个参数等于3
        retq
    }
}

<func4>: rdi = x, rsi = 0x0, rdx = 0xe
rsp -= 0x8
eax = edx - esi
ecx = esi + (((eax >> 0x1f) + eax) >> 1)
if(ecx > edi){
    edx = rcx - 0x1
    call <func4>
    eax *= 2;
    rsp += 0x8
    retq
}
else {
    eax = 0x0
    if(ecx < edi){
        esi = (rcx + 0x1)
        call <func4>
        eax = rax * 2 + 0x1
        rsp += 0x8
        retq
    }
    else {
        rsp += 0x8
        retq
    }
}
```

将func4函数写成C语言：

```
#include <bits/stdc++.h>

using namespace std;

int func4(int rdi, int rsi, int rdx){
    int rax = rdx - rsi;
    int rcx = rsi + (((rax >> 31) + rax) >> 1);
    if(rcx > rdi){
        rdx = rcx - 0x1;
        rax = func4(rdi, rsi, rdx);
        rax *= 2;
        return rax;
    }
    else {
        rax = 0x0;
        if(rcx < rdi){
            rsi = rcx + 0x1;
            rax = func4(rdi, rsi, rdx);
            rax = rax * 2 + 1;
            return rax;
        }
        else {
            return rax;
        }
    }
}

int main(){
    for(int i = 0; i <= 14; i ++){
        int ans = func4(i, 0x0, 0xe);
        cout << i << ": " << ans << endl;
    }
    return 0;
}
```

发现当x=12或13时，eax = 3。故答案为12 3或13 3