评阅教师	得分

三、stack discipline（本大题共 13 分）。

Consider the following C code and assembly code for two mutually recursive functions:

```
int even(unsigned int n)      0x080483e4 <even+0>:    push   %ebp
{                             0x080483e5 <even+1>:    mov    %esp,%ebp
    if(!n)                    0x080483e7 <even+3>:    sub    $0x8,%esp
    {                         0x080483ea <even+6>:    cmpl   $0x0,0x8(%ebp)
        return 1;             0x080483ee <even+10>:   jne    0x80483f9 <even+21>
    }                         0x080483f0 <even+12>:   movl   $0x1,-0x4(%ebp)
                              0x080483f7 <even+19>:   jmp    0x804840a <even+38>
    return odd(n - 1);        0x080483f9 <even+21>:   mov    0x8(%ebp),%eax
}                             0x080483fc <even+24>:   sub    $0x1,%eax
                              0x080483ff <even+27>:   mov    %eax,(%esp)
                              0x08048402 <even+30>:   call   0x804840f <odd>
                              0x08048407 <even+35>:   mov    %eax,-0x4(%ebp)
                              0x0804840a <even+38>:   mov    -0x4(%ebp),%eax
                              0x0804840d <even+41>:   leave
                              0x0804840e <even+42>:   ret

int odd(unsigned int n)       0x0804840f <odd+0>:     push   %ebp
{                             0x08048410 <odd+1>:     mov    %esp,%ebp
    if(!n)                    0x08048412 <odd+3>:     sub    $0x8,%esp
    {                         0x08048415 <odd+6>:     cmpl   $0x0,0x8(%ebp)
        return 0;             0x08048419 <odd+10>:    jne    0x8048424 <odd+21>
    }                         0x0804841b <odd+12>:    movl   $0x0,-0x4(%ebp)
                              0x08048422 <odd+19>:    jmp    0x8048435 <odd+38>
    return even(n - 1);       0x08048424 <odd+21>:    mov    0x8(%ebp),%eax
}                             0x08048427 <odd+24>:    sub    $0x1,%eax
                              0x0804842a <odd+27>:    mov    %eax,(%esp)
                              0x0804842d <odd+30>:    call   0x80483e4 <even>
                              0x08048432 <odd+35>:    mov    %eax,-0x4(%ebp)
                              0x08048435 <odd+38>:    mov    -0x4(%ebp),%eax
                              0x08048438 <odd+41>:    leave
                              0x08048439 <odd+42>:    ret
```

Imagine that a program makes the procedure call **even(3)**. Also imagine that prior to the invocation, the value of ESP is 0xffff1000 - that is, 0xffff1000 is the value of ESP immediately before the execution of the **call** instruction.

1. Note the the call even(3) will result in the following function invocations: even(3) odd(2), even(1), and odd(0). Full in the stack diagram with the values that would be present immediately before the execution of the ret instruction for odd(0). Cross out each blank for which there is insufficient information to complete.

2. What are the values of ESP and EBP immediately before the execution of the ret instruction for odd(0)?

ESP= ____________________________

EBP=_____________________________

```
+----------------------------------+
|                                  |   0xffff1004
+----------------------------------+
|                                  |   0xffff1000
+----------------------------------+
|                                  |   0xffff0ffc
+----------------------------------+
|                                  |   0xffff0ff8
+----------------------------------+
|                                  |   0xffff0ff4
+----------------------------------+
|                                  |   0xffff0ff0
+----------------------------------+
|                                  |   0xffff0fec
+----------------------------------+
|                                  |   0xffff0fe8
+----------------------------------+
|                                  |   0xffff0fe4
+----------------------------------+
|                                  |   0xffff0fe0
+----------------------------------+
|                                  |   0xffff0fdc
+----------------------------------+
|                                  |   0xffff0fd8
+----------------------------------+
|                                  |   0xffff0fd4
+----------------------------------+
|                                  |   0xffff0fd0
+----------------------------------+
|                                  |   0xffff0fcc
+----------------------------------+
|                                  |   0xffff0fc8
+----------------------------------+
|                                  |   0xffff0fc4
+----------------------------------+
|                                  |   0xffff0fc0
+----------------------------------+
```

如果是你出题呢?