得分

第三题（20 分）

请分析下面的 C 语言程序和对应的 x86-64 汇编代码。

1.其中，有一部分缺失的代码（用标号标出），请在标号对应的横线上填写缺失的内容。注:汇编与机器码中的数字用 16 进制数填写。

（1-11 每空 1 分，共 11 分）

C 语言代码如下：

```
typedef struct _parameters {
    int n;
    int product;
} parameters;
int bar(parameters *params, int x) {
    params->product *= x;
}
void foo (parameters *params) {
    if (params->n <= 1)
        ___ (1) ___
    bar(params, ___ (2) ___);
    params->n--;
    foo(params);
}
```

(1) _____

(2) _____

x86-64 汇编代码如下（为简单起见，函数内指令地址只给出后四位，需要时可补全):（

```
0x0000555555555189 <bar>:
    5189: f3 0f 1e fa    endbr64
    518d: 55             push   %rbp
    518e: 48 89 e5       mov    %rsp,%rbp
    5191: 48 89 7d f8    mov    _(3)_,-0x8(%rbp)  (3)_____
    5195: 89 75 f4       mov    %esi,-0xc(%rbp)
    5198: 48 8b 45 f8    mov    -0x8(%rbp),%rax
    519c: 8b 40 04       mov    0x4(%rax),%eax
    519f: 0f af 45 f4    imul   _(4)_(%rbp),%eax  (4)_____
    51a3: 89 c2          mov    %eax,%edx
    51a5: 48 8b 45 f8    mov    -0x8(%rbp),%rax
```

```
    51a9: 89 50 04      mov    %edx,0x4(%rax)
    51ac: 90            nop
    51ad: 5d            pop    _(5)_           (5)_________
    51ae: c3            retq

00005555555551af <foo>:
    51af: f3 0f 1e fa   endbr64
    51b3: 55            push   %rbp
    51b4: 48 89 e5      mov    %rsp,%rbp
    51b7: 48 83 ec 10   _(6)_  $0x10,%rsp       (6)_________
    51bb: 48 89 7d f8   mov    %rdi,-0x8(%rbp)
    51bf: 48 8b 45 f8    mov    -0x8(%rbp),%rax
    51c3: 8b 00         mov    (%rax),%eax
    51c5: 83 f8 01      cmp    $0x1,%eax
    51c8: 7e 31         _(7)_51fb <foo+0x4c>   (7)_________
    51ca: 48 8b 45 f8    mov    -0x8(%rbp),%rax
    51ce: 8b 10         mov    (%rax),%edx
    51d0: 48 8b 45 f8    mov    -0x8(%rbp),%rax
    51d4: 89 d6         mov    %edx,%esi
    51d6: 48 89 c7      mov    %rax,%rdi
    51d9: e8 ab ff ff ff   callq  0x0000555555555189 <bar>
    51de: 48 8b 45 f8    mov    -0x8(%rbp),%rax
    51e2: 8b 00         mov    (%rax),%eax
    51e4: 8d 50 ff      lea  -0x1(_(8)_),%edx   (8)_________
    51e7: 48 8b 45 f8   mov    -0x8(%rbp),%rax
    51eb: 89 10         mov   _(9)_,(%rax)      (9)__________
    51ed: 48 8b 45 f8   mov  _(10)_,%rax       (10)_________
    51f1: 48 89 c7      mov    %rax,%rdi
    51f4: e8 b6 ff ff ff   callq   _(11)_       (11)_________
    51f9: eb 01         jmp    51fc <foo+0x4d>
    51fb: 90            nop
    51fc: c9            leaveq
    51fd: c3            retq
```

2.在程序执行到 0x000055555555518e 时(该指令还未执行),此时的栈帧如下,请填写空格中对应的值。(每空 2 分,共 6 分)

地址	值
0x7fffffffe308	0xffffe340
0x7fffffffe304	0x00000000
0x7fffffffe300	0x00000000
0x7fffffffe2fc	0x00005555
0x7fffffffe2f8	(12) _____
0x7fffffffe2f4	0x00007fff
0x7fffffffe2f0	0xffffe310
0x7fffffffe2ec	0x00007fff
0x7fffffffe2e8	0xffffe340
0x7fffffffe2e4	0x00000004
0x7fffffffe2e0	0xffffe350
0x7fffffffe2dc	0x00005555
0x7fffffffe2d8	(13) _____
0x7fffffffe2d4	0x00007fff
0x7fffffffe2d0	(14) _____

3.当 params={n,1}时, foo(&params)函数的功能是什么? (3 分)

参考答案:

1、(注:本题十六进制未带 0x,不扣分)

(1) return; (注:未带分号,不扣分)

(2) params->n

(3) %rdi

(4) -0xc

(5) %rbp

(6) sub (注:写成 subq,不扣分)

(7) jle

(8) %rax

(9) %edx

(10) -0x8(%rbp)

(11) 0x00005555555551af < foo > (只写数值或者<foo>,都算正确)

(12) 0x555551f9

(13) 0x555551de

（14）0xffffe2f0

2、计算阶乘

考察内容：

1. 对 51c8 及 51fb 的理解，由 51c8 和 51fb 可以知道这是跳转指令，跳转到 51fb，阅读汇编代码可知这里就结束了。结合源代码这里是 return 退出
2. 对 51ca 和 51ce 的理解，由 51ca 可知此时 rax 保存的是 param 的地址，51ce 直接从地址取值，所以取出的是第一个数 n，即 params->n
3. 51d4 和 51d6 通过两个寄存器传参，分别是%esi 和%rdi，5195 使用了%esi，推出此处使用%rdi
4. 对源程序及 5195 的理解，由 519c 可知 eax 保存的是 param->product，所以这里是执行乘法操作，另一个操作数是参数 x，5195 处已经把 x 保存在-0xc(%rbp)中，所以就从这里读取数据
5. Callee 保存的参数，518d 保存，这里读取
6. 进入函数后首先分配栈帧，通过减%rsp 实现
7. 源程序 if (params->n <= 1)的判断
8. 源程序 params->n--;，上一句已经把 params->n 读到%eax 中，这里进行修改
9. 对源程序 params->n--;的理解，51e4 已经把计算结果保存在%edx 中，这里把%edx 的结果进行保存
10. 与 51e7 相同，与 51d0 的内容也相同，考察调用函数的传参方法
11. 函数递归调用，由 foo 函数的内容可得
12. 函数调用保存返回地址，首先进入 foo 函数，所以返回的是调用 foo 函数后的下一条指令地址
13. 函数调用保存返回地址，在 foo 函数中进入 bar 函数，所以返回的是调用 bar 后的下一条指令
14. Push 之后栈顶的内容，推测出此时的 rbp 是在上一次函数调用，进入 callee 时设置的。上一次函数调用保存的返回地址在 0x7fffffffe2f8，所以上一次 rbp 的值是 0x7fffffffe2f0