```
int bar(parameters *params, int x) {
    params->product *= x;
}
```

void

%rdi：指针. %esi：x

```
0x0000555555555189 <bar>:
5189: f3 0f 1e fa    endbr64
518d: 55             push   %rbp
518e: 48 89 e5       mov    %rsp,%rbp
```

保存%rbp

保存rbp

作乘法

```
5191: 48 89 7d f8    mov    %rdi,-0x8(%rbp)    (3) %rdi
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   -0xc(%rbp),%eax    (4) -0xc
51a3: 89 c2          mov    %eax,%edx
51a5: 48 8b 45 f8    mov    -0x8(%rbp),%rax
```

%rax是指针.
⇒把%rdi移动到-0x8(%rbp)的位置.

%eax：params→product.
⇒要乘以x：%esi

```
51a9: 89 50 04       mov    %edx,0x4(%rax)
```

紫色部分的作用？

返回

```
51ac: 90             nop
51ad: 5d             pop    %rbp        (5) %rbp
51ae: c3             retq
```

返回

```
typedef struct _parameters {
    int n;
    int product;
} parameters;
```

设一个指向parameters的指针存在%rdi中

则 movl (%rdi), %eax 代表什么?

movl 4(%rdi), %eax 代表什么?

```
void foo (parameters *params) {
    if (params->n <= 1)
        ___ (1) ___
    bar(params, ___ (2) ___);
    params->n--;
    foo(params);
}
```

有一分支直接返回

(1) return;

(2) params->n

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
```

分配栈帧

(6) sub

```
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>
```

条件: <= 1

(7) jle

not jump

jump

```
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>
```

第一个参数: params

第二个参数: params->n

调用 bar

```
51de: 48 8b 45 f8    mov    -0x8(%rbp),%rax
51e2: 8b 00          mov    (%rax),%eax
51e4: 8d 50 ff       lea    -0x1(_(8)_),%edx
51e7: 48 8b 45 f8    mov    -0x8(%rbp),%rax
51eb: 89 10          mov    _(9)_,(%rax)
51ed: 48 8b 45 f8    mov    _(10)_,%rax
51f1: 48 89 c7       mov    %rax,%rdi
51f4: e8 b6 ff ff ff    callq  _(11)_
51f9: eb 01          jmp    51fc <foo+0x4d>
```

%rax: params

%eax: params->n

(8) %eax %edx: params->n - 1

%rax: params

(9) %edx 写回

(10) -0x8(%rbp) 准备参数

(11) <foo>

```
51fb: 90             nop
51fc: c9             leaveq
51fd: c3             retq
```

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

n!

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。

地址	值
0x7fffffffe308	0xffffe340
0x7fffffffe304	0x00000000
0x7fffffffe300	0x00000000
0x7fffffffe2fc	0x00005555
0x7fffffffe2f8	(12) 0x555555 1f9
0x7fffffffe2f4	0x00007fff
0x7fffffffe2f0	0xffffe310
0x7fffffffe2ec	0x00007fff
0x7fffffffe2e8	0xffffe340
0x7fffffffe2e4	0x00000004
0x7fffffffe2e0	0xffffe350
0x7fffffffe2dc	0x00005555
0x7fffffffe2d8	(13) 0x555555 1de
0x7fffffffe2d4	0x00007fff
0x7fffffffe2d0	(14) 0x###e2f0