

1. 在 x86-64、Linux 操作系统下有如下 C 定义：

```
struct A {
    char CC1[6];
    int II1;
    long LL1;
    char CC2[10];
    long LL2;
    int II2;
};
```

(1) `sizeof(A)` = _____

(2) 将 A 重排后，令结构体尽可能小，那么得到的新的结构体大小为_____字节。

2. X86-64 的程序员可见状态

(1) 通用寄存器：____,____,____,____,%rbp,%rsp,____,____,%r8~%r16

(2) 程序计数器(PC)： ____

(3) 条件码： ____, ____, SF, OF

(4) 其他寄存器（浮点、段等等，不重要）

(5) 内存

3. 寄存器的作用

被调用者保存 callee saved	调用者保存 caller saved
参数（第一个到第六个）	其他

4. 在 C 语言中，定义 `int x = 2`；判断表达式 “`x & 2 == 2`” 的值： _____

5. 书上习题 3.32

接下来的题目分别为：20 年期中第三题、21 年期中第二、三题

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。

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) ____ (1) _____
    bar(params, ____ (2) ____); (2) _____
    params->n--;
    foo(params);
}
```

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 0x00005555555518e 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
0x7fffffffef308	0xffffef340
0x7fffffffef304	0x00000000
0x7fffffffef300	0x00000000
0x7fffffffef2fc	0x00005555
0x7fffffffef2f8	(12) _____
0x7fffffffef2f4	0x00007fff
0x7fffffffef2f0	0xffffef310
0x7fffffffef2ec	0x00007fff
0x7fffffffef2e8	0xffffef340
0x7fffffffef2e4	0x00000004
0x7fffffffef2e0	0xffffef350
0x7fffffffef2dc	0x00005555
0x7fffffffef2d8	(13) _____
0x7fffffffef2d4	0x00007fff
0x7fffffffef2d0	(14) _____

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

得分

第二题 请结合教材第二章“信息的表示和处理”的相关知识,回答下列问题(10分)

1. 假设某 x86-64 机器在地址 0x100 和 0x101 处存储的数据用二进制表示分别为 $[1010\ 1100]_2$ 和 $[1111\ 1011]_2$. 又假设 x 是一个 short 类型的变量, 其地址为 0x100. 则 x 的十六进制补码表示为 (1) 0x_____, 这是一个 (2) _____ (填“正”或“负”) 数, 其绝对值为 (3) _____ (用十进制数字表示).

2. 设整型变量采用 32 位补码表示, 判断正误 (填“√”或“×”):

i. 设 x, y, z 是整型变量, 且 $x < y < z < 0$, 则 $(-y) > (-z) > 0$. ((4) _____)

ii. 表达式 “ $(-5) + \text{sizeof}(\text{int}) < 0$ ” 为真. ((5) _____)

3. 考虑一种新的遵从 IEEE 规范的浮点格式系统, 该浮点数系统包含 1 位符号, 4 位阶码, 7 位尾数.

i. 该浮点数系统所能表示的最大规格化数 (十进制) 为 (6) _____.

ii. 将 $-\frac{3}{256}$ 用该浮点数系统表示, 写成十六进制为 (7) 0x_____. 按照向偶数

舍入的原则, 把 $\frac{3}{256}$ 的二进制表示舍入到最近的一百二十八分之一, 将得到

(8) _____ (填最简真分数).

4. 有下面一段程序:

```
int i, j, k;
for(i = 0; i <= 2147483647 - 2; i ++){
    j = i + 1;
    k = j + 1;
    float *x = (float *)(&i);
    float *y = (float *)(&j);
    float *z = (float *)(&k);
    if(*y - *x != *z - *y){
        printf("0x%08x\n", i);    //输出 8 位 16 进制数
        break;
    }
}
```

其中 float 类型表示 IEEE-754 规定的浮点数, 包括 1 位符号, 8 位阶码, 23 位尾数. 请问该程序是否会有输出? (9) _____ (填“是”或“否”). 若有输出, 请给出输出内容, 若没有输出, 请说明理由 (10) _____.

得分

第三题 (15 分) 请阅读并分析下面的 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。

C 代码如下:

```
long f(long n, long m)
{
    if (n == 0 || (1) _____)
        return m;
    if ((2) _____)
    {
        long ret = (3) _____;
        return ret;
    }
    else
    {
        long ret = f(n - 1, m >> 1);
        return ret;
    }
}
```

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):

```
0x00005555555555149 <f>:
    5149:  f3 0f 1e fa          endbr64
    514d:  55                   push    %rbp
    514e:  48 89 e5             mov     (4) _____, %rbp
    5151:  48 83 ec 20          sub     $0x20, %rsp
    5155:  48 89 7d e8          mov     %rdi, -0x18(%rbp)
    5159:  48 89 75 e0          mov     %rsi, -0x20(%rbp)
    515d:  48 83 7d e8 00        cmpq    $0x0, -0x18(%rbp)
    5162:  74 (5) _____    je      (6) _____
```

5164:	48 83 7d e0 01	cmpq	\$0x1, -0x20(%rbp)
5169:	75 06	jne	5171 <f+0x28>
516b:	48 8b 45 e0	mov	(7) _____(%rbp), %rax
516f:	eb 5f	jmp	51d0 <f+0x87>
5171:	48 8b 45 e0	mov	-0x20(%rbp), %rax
5175:	83 e0 01	and	\$0x1, %eax
5178:	48 85 c0	test	%rax, %rax
517b:	74 ??	je	(8) _____
517d:	48 8b 55 e0	mov	-0x20(%rbp), %rdx
5181:	48 89 d0	mov	%rdx, %rax
5184:	48 01 c0	add	%rax, %rax
5187:	48 01 d0	add	%rdx, %rax
518a:	48 8d 50 01	lea	0x1(%rax), %rdx
518e:	48 8b 45 e8	mov	-0x18(%rbp), %rax
5192:	48 83 e8 01	sub	\$0x1, %rax
5196:	48 89 d6	mov	(9) _____, %rsi
5199:	48 89 c7	mov	%rax, %rdi
519c:	e8 a8 ff ff ff	callq	5149 <f>
51a1:	48 89 45 f8	mov	%rax, -0x8(%rbp)
51a5:	48 8b 45 f8	mov	-0x8(%rbp), %rax
51a9:	eb 25	jmp	51d0 <f+0x87>
51ab:	48 8b 45 e0	mov	-0x20(%rbp), %rax
51af:	48 d1 f8	(10) _____	%rax
51b2:	48 89 c2	mov	%rax, %rdx
51b5:	48 8b 45 e8	mov	-0x18(%rbp), %rax
51b9:	48 83 e8 01	sub	\$0x1, %rax
51bd:	48 89 d6	mov	(9) _____, %rsi
51c0:	48 89 c7	mov	%rax, %rdi
51c3:	e8 81 ff ff ff	callq	5149 <f>
51c8:	48 89 45 f0	mov	%rax, -0x10(%rbp)
51cc:	48 8b 45 f0	mov	-0x10(%rbp), %rax
51d0:	c9	leaveq	
51d1:	c3	retq	

2. 已知在调用函数 $f(7, 6)$ 时，我们在 gdb 中使用 `b f` 指令在函数 f 处加上了断点，下面是程序某一次运行到断点时从栈顶开始的栈的内容，请在空格中填入相应的值。（U 表示不要求填写）

0x7fffffffef558	0x00005555555551c8
0x7fffffffef550	(11) _____
0x7fffffffef548	U
0x7fffffffef540	U
0x7fffffffef538	U
0x7fffffffef530	(12) _____
0x7fffffffef528	(13) _____
0x7fffffffef520	0x00007fffffffef550
0x7fffffffef518	U
0x7fffffffef510	U
0x7fffffffef508	(14) _____
0x7fffffffef500	0x0000000000000010
0x7fffffffef4f8	0x00005555555551c8

3. 运行函数 $f(7, 6)$ 后得到的值是多少？ (15) _____