基本信息：
姓名：_____ 学号：20000_____
——
答卷说明：
a.本卷共8页，卷面分110分.
b.约定：如无特别说明，假设代码全部在Intel x86-64上运行.

一. (20 分) 单项选择题

1. 若想把立即数 x 的 16 进制表示不经修改地放入未清空的通用目的寄存器%rax 中, 下列方法可行的是 (　　)(注: 约定下面的指令中源操作数必须是立即数, 但未必是 $x).
   A. 若 x∈[0x0..0xffff], 则可以使用 movw 指令.
   B. 若 x∈[0x0..0x7fffffff], 则可以使用 movl 指令.
   C. 若 x∈[0x80000000..0xffffffff], 则可以使用 movq 指令.
   D. 若 x∈[0x100000000..0xffffffff7fffffff], 则可以使用 movq 指令.

2. 假设 gcc 开了编译优化，则下列四个函数中第 (　　) 个将被编译为条件传送.

```
long f1(long a, long b) {
    return (++a > --b) ? a : b;
}
```

```
long f2(long *a, long *b) {
    return (*a > *b) ? --(*a) : (*b)--;
}
```

```
long f3(long *a, long *b) {
    return a ? *a : (b ? *b : 0);
}
```

```
long f4(long a, long b) {
    return (a > b) ? a++ : ++b;
}
```

   A. 1
   B. 2
   C. 3
   D. 4

3. 在下面的代码中，A 和B 是用#define 定义的常数：

```
typedef struct {int x[A][B]; long y;} str1;
typedef struct {char array[B]; int t; short s[A]; long u;} str2;
```

```
void setVal(str1 *p, str2 *q) {
    long v1 = q->t; long v2 = q->u;
    p->y = v1+v2;
}
```

GCC 为setVal 产生下面的代码:

```
setVal:
movslq 12(%rsi), %rax
addq 32(%rsi), %rax
movq %rax, 184(%rdi)
ret
```

则A=_____，B=_____.

A. 5,9

B. 9,5

C. 5,12

D. 4,11

4. 考虑下面的联合声明:

```
union ele {
    struct {
        long *p;    long y;
    } e1;
    struct {
    long x;
    union ele *next;
    } e2;
};
```

已知下面的函数 proc(省略了一些表达式) 对一个链表进行操作，生成了对应的汇编代码.

```
//proc函数
void proc (union ele *up) {
    up->____= *(____) - ____;
    }
//汇编
//void proc (union ele *up)
//up in %rdi
proc:
movq 8(%rdi), %rax
movq (%rax), %rdx
movq (%rdx), %rdx
```

```
    subq 8(%rax), %rdx
    movq %rdx, (%rdi)
    ret
```

根据上述信息, 下面说法中错误的是:

A. 假设某结构体只含有 char,short,int,double 这样的基本数据类型. 则按照从大到小的顺序排列变量可以使结构体所占空间最小.

B. e2.next 字段的偏移量为 8.

C. 题中所给结构总共需要 16 字节.

D. proc 函数补全后该行应为 up->e2.x = *(up->e2.next->e1.y) - up->e2.next->e1.p.

二. (90 分) 非选择题

1. (10 分) 在 x86-64、LINUX 操作系统下，考虑如下的 C 定义：

```
typedef union {
    char c[7];
    short h;
} union_e;
typedef struct {
    char d[3];
    union_e u;
    int i;
} struct_e;
struct_e s;
```

回答如下问题：

(1) `s.u.c` 的首地址相对于 s 的首地址的偏移量是_____字节。

(2) `sizeof(union_e)` = _____字节。

(3) s.i 的首地址相对于 s 的首地址的偏移量是_____字节。

(4) 若将`i` 的类型改成`short`、将`h` 的类型改成`int`，那么`sizeof(union_e) =` _____字节，`sizeof(struct_e) =` _____字节。

2. (40 分) 下面的 C 程序包含`main()`, `caller()`, `callee()` 三个函数。本题给出了该程序的部分 C 代码和 x86-64 汇编与机器代码。请分析给出的代码，补全空白处的内容，并回答问题。
注：汇编与机器码中的数字用 16 进制数填写

```
00000000004006cd <caller>:
4006cd:55                   push %rbp
4006ce:48 89 e5             mov %rsp, %rbp
4006d1:48 83 ec 50          sub $0x50, %rsp
4006d5:48 89 7d b8          mov %rdi, -0x48(%rbp)
4006d9:64 48 8b 04 25 28 00 mov %fs:0x28, %rax
4006e0:00 00
4006e2:48 89 45 f8          mov %rax, -0x8(%rbp)
4006e6:31 c0                xor %eax, %eax
4006e8:c6 45 d0 00          movb $0x0, -0x30(%rbp)
4006ec:c6 45 e0 00          movb $0x0, _(1)_
4006f0:48 8b 45 b8          mov _(2)_ , %rax
4006f4:48 89 c7             mov %rax, %rdi
4006f7:                     callq 400510 <strlen@plt>
4006fc:89 45 cc             mov _(3)_ , -0x34(%rbp)
4006ff:83 7d cc 0e          cmpl $0xe, -0x34(%rbp)
400703:7f _(4)_             jg 400752 <caller+0x85>
400705:83 7d cc 09          cmpl $0x9, -0x34(%rbp)
400709:                     jg 400720 <caller+0x53>
40070b:48 8b 55 b8          mov -0x48(%rbp), %rdx
40070f:48 8d 45 d0          lea _(5)_ , %rax
400713:48 89 d6             mov %rdx, %rsi
400716:48 89 c7             mov %rax, %rdi
400719:                     callq 400500 <strcpy@plt>
40071e:                     jmp 40073b <caller+0x6e>
400720:48 8b 45 b8          mov -0x48(%rbp), %rax
400724:48 8d 50 0a          lea 0xa(%rax), %rdx
400728:48 8d 45 d0          lea -0x30(%rbp), %rax
40072c:48 83 c0 10          add _(6)_ , %rax
400730:48 89 d6             mov %rdx, %rsi
400733:48 89 c7             mov %rax, %rdi
400736:                     callq 400500 <strcpy@plt>
40073b:ff 75 e8             pushq -0x18(%rbp)
40073e:ff 75 e0             pushq -0x20(%rbp)
400741:ff 75 d8             pushq -0x28(%rbp)
400744:ff 75 d0             pushq -0x30(%rbp)
400747:e8 _(7)_             callq 400666 <callee>
40074c:48 83 c4 20          add $0x20, %rsp
400750:                     jmp 400753 <caller+0x86>
```

```
400752:90                    nop
400753:48 8b 45 f8           mov _(8)_ , %rax
400757:64 48 33 04 25 28 00  xor %fs:0x28, %rax
40075e:00 00
400760:                      je 400767 <caller+0x9a>
400762:                      callq 400520 <__stack_chk_fail@plt>
400767:c9                    leaveq
400768:c3                    retq
```

```c
#include <stdio.h>
#include "string.h"
#define N _(9)_
#define M _(10)_
typedef union {
    char str_u[N];
    long l;
} union_e;
typedef struct {
    char str_s[M];
    union_e u;
    long c;
} struct_e;
void callee(struct_e s) {
    char buf[M+N];
    strcpy(buf, s.str_s);
    strcat(buf, s.u.str_u);
    printf("%s␣\n", buf);
}
```

```c
void caller(char *str) {
    struct_e s;
    s.str_s[0] = '\0';
    s.u.str_u[0] = '\0';
    int len = strlen(str);
    if (len >= M+N)
        _(11)_;
    else if (len < N) {
        strcpy(s.str_s, _(12)_);
    }
    else {
        strcpy(s.u.str_u, _(13)_);
    }
    callee(s);
}
int main(int argc, char *argv[]){
    caller("0123456789abcd");
    return 0;
}
```

(1) _____ (6) _____ (11) _____
(2) _____ (7) _____ (12) _____
(3) _____ (8) _____ (13) _____
(4) _____ (9) _____
(5) _____ (10) _____

`caller` 函数中，变量`s` 所占的内存空间为：(14)_____.
该程序运行后，`printf` 函数是否有输出？输出结果为：(15)_____.

3. (40 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。

```
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);
}
```

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)
    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
    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)_
    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
    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>
    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
    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>
    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。

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？