学院_____ 专业_____ 班级_____ 学号_____ 姓名_____

密封线　学生须将答案写在此线以下　本专

鲁东大学 2022—2023 学年第 1 学期

2020 级　软件工程 专业本科卷 A 课程名称 计算机系统基础

课程号（22213332）　考试形式（闭卷笔试）　时间（120 分钟）

题　目	一	二	三	四	五	六	七	总 分	统分人	复核人
得　分										

得分	评卷人

一、信息的表示和处理（本题共 3 小题，满分 20 分）

1、（6 分）整数编码。用每行第一列中描述的数字填写下表中的空格（可以用未展开的简单算术表达式给出答案，比如 $15^{23}+42$)。对于此问题，假设字长为 6 位。

描述	数值
T_{min}	
(unsigned) ((int) -7)	
(((unsigned) 0x21) << 1) & 0x3F)	
(int) (20 + 12)	
12 && 4	
(! 0x15) > 16	

2、（4 分）整数运算。在下面的代码中，我们省略了常数 M 和 N 的定义：

```
#define M /* Mystery number 1 */
#define N /* Mystery number 2 */
int arith(int x, int y) {
    int result = 0;
    result = x/M + y*N;  /* M and N are mystery numbers. */
    return result;
}
```

我们以某个 M 和 N 的值编译这段代码。编译器对乘法和除法做了适度优化。下面是将产生出的机器代码翻译回 C 语言的结果：

```
/* Translation of assembly code for arith */
int optarith(int x,  int y) {
    int t = x;
    x >>= 8;
    x -= t;
    if (y < 0) y += 2;
    y <<= 6; /* Arithmetic shift */
    return x+y;
}
```

M 和 N 的值为多少？

解答：M =　　N =

3、（10 分）表达式求值。假定变量 x、f 和 d 的类型分别是 int、float 和 double。除了 f 和 d 都不能等于+、-或者 NaN，它们的值是任意的。对于下面每个 C 表达式，判断它对所有参数值都为真（也就是求值为 1），还是不为真（也就是求值为 0）？

（1）x == (int)(float) x	为真	不为真
（2）x == (int)(double) x	为真	不为真
（3）f == (float)(double) f	为真	不为真
（4）d == (double)(float) d	为真	不为真
（5）f == -(-f);	为真	不为真
（6）2/3 == 2/3.0	为真	不为真
（7）d < 0.0 ⇒ ((d*2) < 0.0)	为真	不为真
（8）d > f ⇒ -f > -d	为真	不为真
（9）d * d >= 0.0	为真	不为真
（10）(d+f)-d == f	为真	不为真

得分	评卷人

二、程序的机器级表示（本题共 3 小题，满分 30 分）

1、（8 分）函数。对于如下 C 代码：

```
long loop_while2(long a, long b)
{
    long result = _____;
    while (_____){
        result = _____;
        b = _____;
    }
    return result;
}
```

以命令行选项-O1 运行 GCC，产生如下代码：

```
    a in %rdi, b in %rsi
1   loop_while 2:
2      testq   %rsi, %rsi
3      jle     .L8
4      movq    %rsi, %rax
```

```
5  .L7:
6      imulq     %rdi, %rax
7      subq      %rdi, %rsi
8      testq     %rsi, %rsi
9      jg        .L7
10     rep; ret
11 .L8:
12     movq      %rsi, %rax
13     ret
```

根据等价的汇编代码行为填写 C 代码中缺失的部分。

2、（12 分）结构。考虑下面的结构声明：

```
struct prob {
    int *p;
    struct {
        int x;
        int y;
    } s;
    struct prob *next;
};
```

下面的过程(省略了某些表达式)对这个结构进行操作：

```
void sp_init(struct prob *sp) {
    sp->s.x  = _________;
    sp->p    = _________;
    sp->next = _________;
}
```

A. 下列字段的偏移量是多少(以字节为单位)?

```
    s.y:  _________
    next: _________
```

B. 这个结构总共需要多少字节?

C. 编译器为 sp_init 的主体产生的汇编代码如下：

```
    void sp_init(struct prob *sp)
    sp in %rdi
1   sp_init:
2     movl 12(%rdi), %eax
3     movl %eax, 8(%rdi)
4     leaq 8(%rdi), %rax
5     movq %rax, (%rdi)
6     movq %rdi, 16(%rdi)
7     ret
```

根据这些信息，填写 sp_init 代码中缺失的表达式。

3、（10 分）switch 语句。对于一个通用结构的 C 函数 switcher：

```
void switcher(long a, long b, long c, long *dest)
{
    long val;
    switch(a) {
    case _____________:   /* Case A */
        c = _____________;
        /* Fall through */
    case _____________:   /* Case B */
        val = _____________;
        break;
    case _____________:   /* Case C */
    case _____________:   /* Case D */
        val =  _____________;
        break;
    case _____________:   /* Case E */
        val =  _____________;
        break;
    default:
        val =  _____________;
    }
    *dest = val;
}
```

GCC 产生如下图所示的汇编代码和跳转表。

填写 C 代码中缺失的部分。除了情况标号 C 和 D 的顺序之外，将不同情况填入这个模板的方式是唯一的。

得分	评卷人

三、优化程序性能（本题共 1 小题，满分 8 分）

考虑下面的函数，它将一个数组的内容复制到另一个数组：

```
1     void copy_array(long *src, long *dest, long n)
2     {
3           long i;
4           for (i=0; i<n; i++)
5                 dest[i] = src[i];
6     }
```

假设 a 是一个长度为 1000 的数组，被初始化为每个元素 a[*i*]等于 *i*。

回答以下问题：

```
    void switcher(long a, long b, long c, long *dest)
    a in %rdi, b in %rsi, c in %rdx, dest in %rcx
1   switcher:
2     cmpq    $7, %rdi
3     ja      .L2
4     jmp     *.L4(,%rdi,8)
5     .section        .rodata
6   .L7:
7     xorq    $15, %rsi
8     movq    %rsi, %rdx
9   .L3:
10    leaq    112(%rdx), %rdi
11    jmp     .L6
12  .L5:
13    leaq    (%rdx,%rsi), %rdi
14    salq    $2, %rdi
15    jmp     .L6
16  .L2:
17    movq    %rsi, %rdi
18  .L6:
19    movq    %rdi, (%rcx)
20    ret
```

三、3、(a)代码

```
1   .L4:
2     .quad   .L3
3     .quad   .L2
4     .quad   .L5
5     .quad   .L2
6     .quad   .L6
7     .quad   .L7
8     .quad   .L2
9     .quad   .L5
```

三、3、(b)跳转表

A.调用 copy_array(a+1,a,999) 的效果是什么?

B.调用 copy_array(a,a+1,999) 的效果是什么?

C.我们的性能测试表明问题 A 调用的 CPE 为 1.2(循环展开因子为 4 时，该值下降到 1.0)，而问题 B 调用的 CPE 为 5.0。你认为是什么因素造成了这样的性能差异?

D.你预计调用 copy_array(a,a,999) 的性能会是怎样的?

得分	评卷人

四、存储器层次结构（本题共 2 小题，满分 18 分）

1、(6 分）栈。下图给出了两个函数 top 和 leaf 的反汇编代码，以及 main 函数中调用 top 处的代码。每条指令都以标号标出：L1~L2 (leaf 中)，T1~T4 (main 中) 和 M1~M2 (main 中)。


```
    Disassembly of leaf(long y)
    y in %rdi
1   0000000000400540 <leaf>:
2     400540:  48 8d 47 02          lea    0x2(%rdi),%rax  L1: y+2
3     400544:  c3                   retq                   L2: Return

4   0000000000400545 <top>:
    Disassembly of top(long x)
    x in %rdi
5     400545:  48 83 ef 05          sub    $0x5,%rdi       T1: x-5
6     400549:  e8 f2 ff ff ff       callq  400540 <leaf>   T2: Call leaf(x-5)
7     40054e:  48 01 c0             add    %rax,%rax       T3: Double result
8     400551:  c3                   retq                   T4: Return

    . . .
     Call to top from function main
9     40055b:  e8 e5 ff ff ff       callq  400545 <top>    M1: Call top(100)
10    400560:  48 89 c2             mov    %rax,%rdx       M2: Resume
```

在 main 调用 top（100）之前的程序状态如下：

指令			状态值（指令执行前）				描述
标号	PC	指令	%rdi	%rax	%rsp	*%rsp	
M1	0x40055b	callq	100	—	0x7fffffffe820	—	调用top(100)

请你写出：top 的指令 T3 执行前，寄存器%rax、%rsp 的内容，以及位于栈顶的值。

2、（12 分）cache 命中。考虑下面的转置函数：

```
typedef int array[2][2];
void transpose(array dst, array src) {
    int i, j;
    for (j = 0; j < 2; j++) {
        for (i = 0; i < 2; i++) {
            dst[i][j] = src[j][i];
        }
    }
}
```

假设在一台具有以下属性的计算机上运行：

- sizeof(int) == 4。
- src 数组从地址 0 开始，dst 数组从地址 16(十进制) 开始。
- 只有一个 L1 数据高速缓存，它是直接映射、直写和写分配的，块大小为 8 个字节。
- 对 src 和 dst 数组的访问分别是对缓存读和写访问的唯一来源。

A. 假设高速缓存的总大小为 16 个数据字节(即，块大小乘以组的数量是 16 字节)，并且

高速缓存最初是空的。

对于每个 row 和 col，指出每个对 src[row][col]和 dst[row][col]的访问是命中(H)还是未命中(M)（例如，读取 src[0][0]是未命中，写入 dst[0][0]也是未命中）。

src 数组		
	col 0	col 1
row 0	m	
row 1		

dst 数组		
	col 0	col 1
row 0	m	
row 1		

B. 对于总大小为 32 个数据字节的高速缓存，重复 A 部分。

src 数组		
	col 0	col 1
row 0	m	
row 1		

dst 数组		
	col 0	col 1
row 0	m	
row 1		

得分	评卷人

五、链接（本题共 1 小题，满分 6 分）

考虑可执行目标文件 a.out，它是使用命令

```
unix> gcc -o a.out main.c foo.c
```

编译和链接的，文件 main.c 和 foo.c 由以下代码组成：

```
/* main.c */
#include <stdio.h>
int a = 1;
static int b = 2;
int c = 3;
void foo();

int main()
{
  int c = 4;

  foo();
  printf("a=%d b=%d c=%d\n", a, b, c);
  return 0;
}
```

```
/* foo.c */
int a, b, c;

void foo()
{
    a = 5;
    b = 6;
    c = 7;
}
```

问：a.out 的输出是什么？

得分	评卷人

六、异常控制流（本题共 1 小题，满分 6 分）

考虑以下 C 程序。（由于篇幅原因，我们不检查错误返回码，因此假设所有函数都正常返回。）

```
int main()
{
    int val = 2;

    printf("%d", 0);
    fflush(stdout);

    if (fork() == 0) {
        val++;
        printf("%d", val);
        fflush(stdout);
    }
    else {
        val--;
        printf("%d", val);
        fflush(stdout);
        wait(NULL);
    }
    val++;
    printf("%d", val);
    fflush(stdout);
    exit(0);
}
```

对于以下每个字符串，圈出 (Y) 或 (N) 该字符串是否是程序的可能输出。

A. 01432 Y N

B. 03142 Y N

C. 01234 Y N

得分	评卷人

七、虚拟存储（本题共 1 小题，满分 12 分）地址翻译。

这个问题涉及将虚拟地址转换为物理地址的方式。

- 存储器是字节可寻址的，内存访问是针对 1 字节（不是 4 字节）的字。
- 虚拟地址是 17 位宽。
- 物理地址是 12 位宽。
- 页面大小为 256 字节。

- 每个页表条目包括：
  - 物理页号
  - 有效位
- TLB 是 4 路组相联，总共有 16 个条目。
- cache 是 2 路组相联，cache 行大小是 4 字节，总共有 64 个总条目。

在下表中，**所有数字均以十六进制形式给出**。TLB 的内容和前 32 页的页表，以及缓存如下：

TLB

Index	Tag	PPN	Valid
0	55	6	0
	48	F	1
	00	C	0
	77	9	1
1	01	4	1
	32	A	1
	02	F	0
	73	0	1
2	02	3	1
	0F	B	0
	04	3	0
	26	C	0
3	00	8	1
	7A	2	1
	21	1	0
	17	E	0

页表

VPN	PPN	Valid	VPN	PPN	Valid
000	C	0	010	1	1
001	7	1	011	8	1
002	3	1	012	3	0
003	8	1	013	E	1
004	0	0	014	6	0
005	5	0	015	C	0
006	C	1	016	7	0
007	4	1	017	2	1
008	D	1	018	9	1
009	F	0	019	A	0
00A	3	1	01A	B	0
00B	0	1	01B	3	1
00C	0	0	01C	2	1
00D	F	1	01D	9	0
00E	4	0	01E	5	0
00F	7	1	01F	B	1

2路组相联 Cache

Index	Tag	Valid	Byte 0	Byte 1	Byte 2	Byte 3	Tag	Valid	Byte 0	Byte 1	Byte 2	Byte 3
0	7A	1	09	EE	12	64	00	0	99	04	03	48
1	02	0	60	17	18	19	38	1	00	BC	0B	37
2	55	1	30	EB	C2	0D	0B	0	8F	E2	05	BD
3	07	1	03	04	05	06	5D	1	7A	08	03	22
4	12	0	06	78	07	C5	05	1	40	67	C2	3B
5	71	1	0B	DE	18	4B	6E	0	B0	39	D3	F7
6	91	1	A0	B7	26	2D	F0	0	0C	71	40	10
7	46	0	B1	0A	32	0F	DE	1	12	C0	88	37

对于给定的虚拟地址：01FAD，指出访问的 TLB 条目和物理地址、TLB 是否未命中以及是否发生缺页。如果存在 cache 未命中，请在“返回的 cache 字节值”输入“-”。如果存在缺页，请在“PPN”中输入“-”并将后续部分留空。

A. 回答问题：每个页表条目有多少位？页表大小是多少？（2^i 形式的回答也可接受）

B. 地址翻译：

参数	值
VPN	0x
TLB Index	0x
TLB Tag	0x
TLB 命中? (Y/N)	
缺页? (Y/N)	
PPN	0x

C. 物理内存引用：

参数	值
Block Offset	0x
Cache Index	0x
Cache Tag	0x
Cache 命中? (Y/N)	
返回的 Cache字节值	0x