

北京大学信息科学技术学院考试试卷

考试科目： 计算机系统导论 姓名： _____ 学号： _____

考试时间： 2013 年 11 月 12 日 任课教师： _____

题号	一	二	三	四	五	六	七	八	总分
分数									
阅卷人									

北京大学考场纪律

1、考生进入考场后，按照监考老师安排隔位就座，将学生证放在桌面上。无学生证者不能参加考试；迟到超过 15 分钟不得入场。在考试开始 30 分钟后方可交卷出场。

2、除必要的文具和主考教师允许的工具书、参考书、计算器以外，其它所有物品（包括空白纸张、手机、或有存储、编程、查询功能的电子用品等）不得带入座位，已经带入考场的必须放在监考人员指定的位置。

3、考试使用的试题、答卷、草稿纸由监考人员统一发放，考试结束时收回，一律不准带出考场。若有试题印制问题请向监考教师提出，不得向其他考生询问。提前答完试卷，应举手示意请监考人员收卷后方可离开；交卷后不得在考场内逗留或在附近高声交谈。未交卷擅自离开考场，不得重新进入考场答卷。考试结束时间到，考生立即停止答卷，在座位上等待监考人员收卷清点后，方可离场。

4、考生要严格遵守考场规则，在规定时间内独立完成答卷。不准交头接耳，不准偷看、夹带、抄袭或者有意让他人抄袭答题内容，不准接传答案或者试卷等。凡有违纪作弊者，一经发现，当场取消其考试资格，并根据《北京大学本科考试工作与学术规范条例》及相关规定严肃处理。

5、考生须确认自己填写的个人信息真实、准确，并承担信息填写错误带来的一切责任与后果。

学校倡议所有考生以北京大学学生的荣誉与诚信答卷，共同维护北京大学的学术声誉。

以下为试题和答题纸，共 14 页。

得分

第一题 选择题（每小题 2 分，共 34 分）

（注：每小题有一个或多个正确答案）

1、变量 x 的值为 0x01234567，地址 $\&x$ 为 0x100；则该变量的值在 x86 和 Sun 机器内存中的存储排列顺序正确的是（ ）

选项	机器类型	地址			
		0x100	0x101	0x102	0x103
A	x86	67	45	23	01
	Sun	01	23	45	67
B	x86	76	54	32	10
	Sun	01	23	45	67
C	x86	01	23	45	67
	Sun	67	45	23	01
D	x86	01	23	45	67
	Sun	01	23	45	67

2、假设下列 int 和 unsigned 数均为 32 位，

`int x = 0x80000000;`

`unsigned y = 0x00000001;`

`int z = 0x80000001;`

以下表达式正确的是（ ）

A. $(-x) < 0$

B. $(-1) > y$

C. $(z < 3) == (z * 8)$

D. $y * 24 == z < 5 - z < 3$

3、对 $x = 1\frac{1}{8}$ 和 $y = 1\frac{3}{8}$ 进行小数点后两位取整（rounding to nearest even），结果是正确的是（ ）

A. $1\frac{1}{4}, 1\frac{1}{4}$ B. $1, 1\frac{1}{4}$ C. $1\frac{1}{4}, 1\frac{1}{2}$ D. $1, 1\frac{1}{2}$

4、在完成本课程的 Bomb Lab 的时候，通常先执行 gdb bomb 启动调试，然后执行 ____ explode_bomb 命令以防引爆炸弹，之后在进行其他必要的设置后，最后执行 ____ 命令以便开始执行程序。上述两个空格对应的命令是（ ）

A. st, ru B. br, go C. br, ru D. st, go

5、已知函数 `int x(int n) { return n*____; }` 对应的汇编代码如下：

```
lea (%rdi, %rdi, 4), %rdi
lea (%rdi, %rdi, 1), %eax
retq
```

请问横线上的数字应该是 ()

- A. 4 B. 5 C. 2 D. 10

6、32 位 x86 计算机、Windows 操作系统下定义的一个 `structure S` 包含三个部分：`double a, int b, char c`，请问 `S` 在内存空间中最多和最少分别能占据多少个字节（32 位 Windows 系统按 1、4、8 的原则对齐 `char`、`int`、`double`）？答：()

- A. 16, 13
B. 16, 16
C. 24, 13
D. 24, 16

7、x86 体系结构的内存寻址方式有多种格式，请问下列哪些指令是正确的：()

- A. `movl $34, (%eax)`
B. `movl (%eax), %eax`
C. `movl $23, 10(%edx, %eax)`
D. `movl (%eax), 8(%ebx)`

8、x86 体系结构中，下面哪些选项是错误的？答：()

- A. `leal` 指令只能够用来计算内存地址
B. x86_64 机器可以使用栈来给函数传递参数
C. 在一个函数内，改变任一寄存器的值之前必须先将其原始数据保存在栈内
D. 判断两个寄存器中值大小关系，只需 `SF`（符号）和 `ZF`（零）两个条件码

9、下面对 `RISC` 和 `CISC` 的描述中，错误的是：()

- A. `CISC` 指令系统中的指令数目较多，有些指令的执行周期很长；而 `RISC` 指令系统中通常指令数目较少，指令的执行周期都较短。
B. `CISC` 指令系统中的指令编码长度不固定；`RISC` 指令系统中的指令编码长度固定，这样使得 `RISC` 机器可以获得了更短的代码长度。
C. `CISC` 指令系统支持多种寻址方式，`RISC` 指令系统支持的寻址方式较少。
D. `CISC` 机器的寄存器数目较少，函数参数必须通过栈进行传递；`RISC` 机器的寄存器数目较多，可以通过寄存器来传递参数，避免了不必要的存储访问。

- 10、下面对流水线技术的描述，正确的是：（ ）
- A. 流水线技术不仅能够提高指令的吞吐率，还能减少单条指令的执行时间。
 - B. 不断加深流水线级数，总能获得性能上的提升。
 - C. 流水级划分应尽量均衡，吞吐率会受到最慢的流水级影响。
 - D. 指令间数据相关可能引发数据冒险，可通过数据转发或暂停流水线来解决。

(11-13)、在教材所描述的流水线处理器（the PIPE processor）上分别运行如下四段Y86程序代码。请分析其中数据冒险的具体情况，并回答后续3个小题。

#Program 1: mrmovl 8(%ebx), %edx rmmovl %edx, 16(%ecx)	#Program 2: mrmovl 8(%ebx), %edx nop rmmovl %edx, 16(%ecx)
#Program 3: mrmovl 8(%ebx), %edx nop nop rmmovl %edx, 16(%ecx)	#Program 4: mrmovl 8(%ebx), %edx nop nop nop rmmovl %edx, 16(%ecx)

- 11、对于每段程序，请指出是否会因为数据冒险导致流水线停顿（Stall）。
- Program 1: (), Program 2: (), Program 3: (), Program 4: ();
- A. Stall B. No-Stall
- 12、对于每段程序，请指出流水线处理器内是否会产生数据转发（Forwarding）。
- Program 1: (), Program 2: (), Program 3: (), Program 4: ();
- A. Forwarding B. No-Forwarding
- 13、对于每段程序，请指出流水线处理器内使用哪个信号进行数据转发，如果不进行数据转发，则用none表示。
- Program 1: (), Program 2: (), Program 3: (), Program 4: ();
- A. m_valM B. W_valM C. none
- 14、下面哪些选项是错误的？答：（ ）
- A. 同一个任务采用时间复杂度为 $O(\log N)$ 算法一定比采用复杂度为 $O(N)$ 算法的执行时间短
 - B. 编译器进行程序优化时，总是可以使用算数结合律来减少计算量
 - C. 增大循环展开（loop unrolling）的级数，有可能降低程序的执行性能（即增加执行时间）
 - D. 分支预测时，“总是预测不跳转”（branch not taken）一定比“总是预测跳转”（branch taken）预测准确率高

15、以下哪些程序优化编译器总是可以自动进行？（假设 `int i, int j, int A[N], int B[N], int m` 都是局部变量，`N` 是一个整数型常量，`int foo(int)` 是一个函数）答：（ ）

	优化前	优化后
A.	<pre>for (j = 0 ; j < N ; j ++) m += i*N*j;</pre>	<pre>int temp = i*N; for (j= 0 ; j < N ; j ++) m += temp * j;</pre>
B.	<pre>for (j = 0 ; j < N ; j ++) B[i] *= A[j];</pre>	<pre>int temp = B[i]; for (j= 0 ; j < N ; j ++) temp *= A[j]; B[i] = temp;</pre>
C.	<pre>for (j = 0 ; j < N ; j ++) m = (m + A[j]) + B[j];</pre>	<pre>for (j = 0 ; j < N ; j ++) m = m + (A[j] + B[j]);</pre>
D.	<pre>for (j = 0 ; j < foo(N) ; j ++) m ++;</pre>	<pre>int temp = foo(N); for (j= 0 ; j < temp ; j ++) m ++;</pre>

16、如果直接映射高速缓存大小是 4KB，并且块（block）大小为 32 字节，请问它每组（set）有多少行（line）？答：（ ）

- A. 128 B. 64 C. 32 D. 1

17、关于局部性（locality）的描述，不正确的是：（ ）

- A. 数组通常具有很好的时间局部性
- B. 数组通常具有很好的空间局部性
- C. 循环通常具有很好的时间局部性
- D. 循环通常具有很好的空间局部性

得分

第二题（8 分）

1) 判断下表中每一行表达式对或错。如果错，请举出反例或说明原因（每行 1 分）

int x, y ;
unsigned u, v ;

	True or False	原因或举出反例
if $x < 0$, then $x * 2 < 0$		
$u \leq -1$		
if $x > y$, then $-x < -y$		
if $u > v$, then $-u > -v$		

2) 请按单精度浮点数表示下表中的数值，首先写出形如 $(-1)^s \times M \times 2^E$ 的表达式，然后给出十六进制的表示。（每格 1 分）

注：符号位 (s): 1-bit; 阶码字段 (exp): 8-bit; 小数字段 (frac): 23-bit; 偏置值 (bias): 127。

Value	$(-1)^s \times M \times 2^E, 1 \leq M < 2$	Hex representation
$-1\frac{1}{2}$		
2^{-149}		

得分

第三题 （11 分）

阅读下面的 C 代码：

```
/*
 * Copyright (C) 2013 Davidlohr Bueso <davidlohr.bueso@hp.com>
 *
 * Based on the shift-and-subtract algorithm for computing integer
 * square root from Guy L. Steele.
 */

/**
 * int_sqrt - rough approximation to sqrt
 * @x: integer of which to calculate the sqrt
 *
 * A very rough approximation to the sqrt() function.
 */
unsigned long int_sqrt(unsigned long x)
{
    unsigned long b, m, y = 0;

    if (x <= 1)
        return x;

    m = 1UL << (BITS_PER_LONG - 2);
    while (m != 0) {
        b = y + m;
        y >>= 1;

        if (x >= b) {
            x -= b;
            y += m;
        }
        m >>= 2;
    }

    return y;
}
```

1) 在 64 位的机器上 BITS_PER_LONG 的定义为 long 类型的二进制位数，它是多少位？（1 分）

答：

2) 填写下面反汇编中的缺失的内容：（每空 1 分）

<int_sqrt>:

```

4004c4:    push    %rbp
4004c5:    mov     %rsp,%rbp
4004c8:    mov     %rdi,-0x28(%rbp)

4004cc:    movq    __ (1) __,-0x8(%rbp)
4004d4:    cmpq    $0x1,-0x28(%rbp)

4004d9:    ja      __ (2) __ <int_sqrt+??>
4004db:    mov     -0x28(%rbp),%rax

4004df:    jmp     __ (3) __ <int_sqrt+??>
4004e1:    movl    $0x0,-0x10(%rbp)

4004e8:    movl    __ (4) __,-0xc(%rbp)

4004ef:    jmp     __ (5) __ <int_sqrt+??>
4004f1:    mov     -0x10(%rbp),%rax
4004f5:    mov     -0x8(%rbp),%rdx

4004f9:    lea     __ (6) __,%rax
4004fd:    mov     %rax,-0x18(%rbp)
400501:    shrq    -0x8(%rbp)
400505:    mov     -0x28(%rbp),%rax
400509:    cmp     -0x18(%rbp),%rax

40050d:    jb      __ (7) __ <int_sqrt+??>
40050f:    mov     -0x18(%rbp),%rax
400513:    sub     %rax,-0x28(%rbp)
400517:    mov     -0x10(%rbp),%rax
40051b:    add     %rax,-0x8(%rbp)

40051f:    shrq    __ (8) __,-0x10(%rbp)
400524:    cmpq    $0x0,-0x10(%rbp)

400529:    jne     __ (9) __ <int_sqrt+??>

40052b:    mov     -0x8(%rbp),__ (10) __
40052f:    leaveq
400530:    retq

```


得分

第四题（10 分）

阅读下面的汇编代码：

<f>:

```

4004c4:    push    %rbp
4004c5:    mov     %rsp,%rbp
4004c8:    sub     $0x10,%rsp
4004cc:    mov     %edi,-0x4(%rbp)
4004cf:    cmpl    $0x1,-0x4(%rbp)
4004d3:    ja      4004dc <f+0x18>
4004d5:    mov     $0x1,%eax
4004da:    jmp     40052d <f+0x69>
4004dc:    mov     -0x4(%rbp),%eax
4004df:    and     $0x1,%eax
4004e2:    test    %eax,%eax
4004e4:    jne     4004f5 <f+0x31>
4004e6:    mov     0x200440(%rip),%eax    # 60092c <x.1604>
4004ec:    add     $0x1,%eax
4004ef:    mov     %eax,0x200437(%rip)    # 60092c <x.1604>
4004f5:    mov     -0x4(%rbp),%eax
4004f8:    and     $0x1,%eax
4004fb:    test    %al,%al
4004fd:    je      40050e <f+0x4a>
4004ff:    mov     0x20042b(%rip),%eax    # 600930 <y.1605>
400505:    add     $0x1,%eax
400508:    mov     %eax,0x200422(%rip)    # 600930 <y.1605>
40050e:    mov     -0x4(%rbp),%eax
400511:    sub     $0x1,%eax
400514:    mov     %eax,%edi
400516:    callq   4004c4 <f>
40051b:    mov     0x20040f(%rip),%edx    # 600930 <y.1605>
400521:    lea     (%rax,%rdx,1),%edx
400524:    mov     0x200402(%rip),%eax    # 60092c <x.1604>
40052a:    lea     (%rdx,%rax,1),%eax
40052d:    leaveq
40052e:    retq

```

1) 程序

```
main()
{
    unsigned int n;
    for (n=1; n< 4; n++) {
        printf("f(%d) = %x\n", n, f(n));
    }
}
```

的运行结果为: f(1)=1, f(2)=4e, f(3)=9f, 请填写 f 函数所需要的内容 (每空 1 分):

```
#define N   (1) _____
#define M   (2) _____
```

```
struct P1 {char c[N]; char *d[N]; char e[N]; } P1;
struct P2 {int i[M]; char j[M]; short k[M]; } P2;
```

```
unsigned int f(unsigned int n)
{
    (3) _____ unsigned int x = sizeof(P1);
    (4) _____ unsigned int y = sizeof(P2);

    if ( (5) _____ )
        return 1;

    if ( (6) _____ )
        x++;

    if ( (7) _____ )
        y++;

    return (8) _____;
}
```

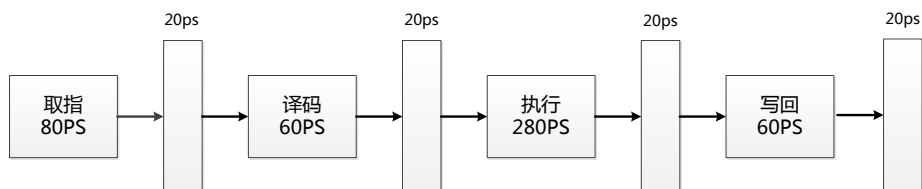
2、程序

```
main()
{
    printf("%x, %x\n", f(2), f(2));
}
```

的运行结果为: _____ (2 分)

得分

第五题（9分）



在“取指-译码-执行-写回”的四级流水线中，各流水级的工作内容和延迟如上图所示，寄存器的延迟也已标出。数据和指令分别存放在不同的存储器中。Cycle N 写入寄存器文件的数据 Cycle N+1 才可读出。请问：

1) 若不考虑流水线填充和清空时间，请计算该处理器的吞吐率。（1分）

答：

2) 若将该处理器改造为单周期（SEQ），请计算 SEQ 处理器的吞吐率。（1分）

答：

3) 在上述流水线中，执行阶段包含了访问数据存储的时间。对于如下 Y86 程序段，指令间存在哪些数据相关（dependence），会引起哪些数据冒险（hazard）？（5分）

Prog:

```

irmovl $128, %edx      #instr1
irmovl $3, %ecx         #instr2
rmmovl %ecx, 0(%edx)    #instr3
irmovl $10, %ebx        #instr4
mrmovl 0(%edx), %eax    #instr5
addl   %ebx, %eax       #instr6
  
```

答：

4) 以上的数据冒险，可以通过转发（forward）的方法解决。请结合上述程序代码和流水线结构图逐个说明解决方案。（2分）

答：

得分

第六题（9 分）

请分析Y86 ISA中定义的两条指令（cmovXX、call）和一条新加入Y86 ISA的IA32指令（decl：将操作数减1）。若在教材所描述的SEQ处理器上执行这些指令，请按下表填写每个阶段进行的操作。如果在某一阶段没有任何操作，请填写none指明。（每条指令3分）

注1、所用到的指令编码为：

cmovXX rA, rB	2	fn	rA	rB	
call Dest	8	0	Dest		
decl rA	C	0	rA	F	

注2、需说明的信号包括：icode, ifun, rA, rB, valA, valB, valC, valE, valP; the register file R[], data memory M[], Program counter PC, condition codes CC。

Stage	cmovXX rA, rB	call Dest	decl rA
Fetch			
Decode			
Execute			
Memory			
Write back			
PC update			

得分

第七题（10 分）

已知如下的汇编程序实现了函数 transform(char* src, char* tgt, char delta)

transform:

 jmp L2

L1:

 add %edx, %eax

 add \$1, %rdi

 mov %al, (%rsi)

 add \$1, %rsi

L2:

 movzbl (%rdi), %eax

 test %al, %al

 jne L1

 movb \$0, (%rsi)

 retq

注：x86-64 位指令中传递前三个参数分别使用寄存器%rdi, %rsi 和%rdx

1) 写出 transform 函数对应的 C 语言版本（2 分）

2) 假设读写访存指令延迟为 20 个时钟周期，其他指令延迟为 2 个时钟周期，所有分支预测都成功。同时 CPU 包含足够多的部件来实现指令并行，那么 CPE 最低应该是多少（2 分）？为什么（2 分）？

3) 已知 src 对应字符串中每个字符 c 都满足 $0 < c \leq 80$ 且 $0 \leq \text{delta} \leq 5$ 。通过下面的改写，可以把 transform 程序 CPE 的理论下限降低一半，请填空。假设程序运行在小端法机器上。（每空 1 分）

void transform(char* src, char* tgt, char delta) {

 short x = _____;

 while(*src && _____) {

 (short)tgt = *(short*)src + x;

 src += 2;

 tgt += 2;

 }

 (short)tgt = _____ ? *(short*)src + delta : *(short*)tgt & _____;

}

得分

第八题（9 分）

假设存在一个能够存储四个数据的 Cache，每一行（line）的数据块长度（B）为 2 字节。假设内存空间一共是 32 个字节，即内存空间地址长度一共是 5 个比特：从 0（00000₂）到 31（11111₂）。某程序片段一共有 8 个数据读取操作，每个操作的地址按顺序如下所示（单位是字节），

数据访问地址序列：1→4→17→2→8→16→9→0

数据替换采用 LRU（least recently used）策略，即“最久没有被访问”的 cache line 作为替换对象。例如：如果 A、B、C、D 四个 cache line 被访问的顺序是 A→B→A→C→D，则在这四个 cache line 里面，B 是最久没有被访问的 cache line。

1) 如果 Cache 的结构是直接映射（directed mapped）（S=4，E=1），如下图所示。请在下图空白处填入，访问上述数据序列访问后 Cache 的状态。（注：TAG 使用二进制格式，V=1 代表数据有效，用[A-B]表示地址 A 到 B 之间对应的数据）（4 分）

V	TAG	DATA

2) 如果 cache 的结构如下图所示（S=2，E=2），请填入访问后的状态（2 分）

V	TAG	DATA	V	TAG	DATA

在这种情况下，数据访问一共产生了多少次 Miss：_____（1 分）

3) 如果 cache 的结构变成（S=1，E=4），最终存储在 Cache 里面的数据有哪些？（注：只需要填写数据部分，顺序不限）

_____, _____, _____, _____（2 分）

北京大学信息科学技术学院考试试卷

考试科目： 计算机系统导论 姓名： _____ 学号： _____

考试时间： 2014 年 11 月 13 日 任课教师： _____

题号	一	二	三	四	五	六	七	八	总分
分数									
阅卷人									

北京大学考场纪律

1、考生进入考场后，按照监考老师安排隔位就座，将学生证放在桌面上。无学生证者不能参加考试；迟到超过 15 分钟不得入场。在考试开始 30 分钟后方可交卷出场。

2、除必要的文具和主考教师允许的工具书、参考书、计算器以外，其它所有物品（包括空白纸张、手机、或有存储、编程、查询功能的电子用品等）不得带入座位，已经带入考场的必须放在监考人员指定的位置。

3、考试使用的试题、答卷、草稿纸由监考人员统一发放，考试结束时收回，一律不准带出考场。若有试题印制问题请向监考教师提出，不得向其他考生询问。提前答完试卷，应举手示意请监考人员收卷后方可离开；交卷后不得在考场内逗留或在附近高声交谈。未交卷擅自离开考场，不得重新进入考场答卷。考试结束时间到，考生立即停止答卷，在座位上等待监考人员收卷清点后，方可离场。

4、考生要严格遵守考场规则，在规定时间内独立完成答卷。不准交头接耳，不准偷看、夹带、抄袭或者有意让他人抄袭答题内容，不准接传答案或者试卷等。凡有违纪作弊者，一经发现，当场取消其考试资格，并根据《北京大学本科考试工作与学术规范条例》及相关规定严肃处理。

5、考生须确认自己填写的个人信息真实、准确，并承担信息填写错误带来的一切责任与后果。

学校倡议所有考生以北京大学学生的荣誉与诚信答卷，共同维护北京大学的学术声誉。

以下为试题和答题纸，共 15 页。

得分

第一题 单项选择题（每小题 2 分，共 32 分）

1、假设下列 unsigned 和 int 数均为 32 位，

unsigned x = 0x00000001;

int y = 0x80000000;

int z = 0x80000001;

以下表达式正确的是

A. $(-1) < x$

B. $(-y) > -1$

C. $\sim y + y == -1$

D. $(z << 4) > (z * 16)$

答：()

2、下面说法正确的是：

A. 数 0 的反码表示是唯一的

B. 数 0 的补码表示不是唯一的

C. 1000, 1111, 1110, 1111, 1100, 0000, 0000, 0000 表示唯一的整数是 0x8FEFC000

D. 1000, 1111, 1110, 1111, 1100, 0000, 0000, 0000 如果是单精度浮点表示，则表示的是 $-(1.110111111)_2 * 2^{31-127}$

答：()

3、下面表达式中为“真”的是：

A. $(\text{unsigned}) -1 < -2$

B. $2147483647 > (\text{int}) 2147483648U$

C. $(0x80005942 \gg 4) == 0x09005942$

D. $2147483647 + 1 != 2147483648$

答：()

4、下列的指令组中，那一组指令只改变条件码，而不改变寄存器的值？

A. CMP, SUB

B. TEST, AND

C. CMP, TEST

D. LEAL, CMP

答：()

5、下列指令中，寻址方式不正确的是

- A. `MOVB %ah, 0x20(, %ecx, 8)`
- B. `LEAL (0xA, %eax), %ebx`
- C. `SUBB 0x1B, %bl`
- D. `INCL (%ebx, %eax)`

答：()

6、有如下定义的结构，在 x86-64 下，下述结论中错误的是？

```
struct {
    char c;
    union {
        char vc;
        double value;
        int vi;
    } u;
    int i;
} sa;
```

- A. `sizeof(sa) == 24`
- B. `(&sa.i - &sa.u.vi) == 8`
- C. `(&sa.u.vc - &sa.c) == 8`
- D. 优化成员变量的顺序，可以做到“`sizeof(sa) == 16`”

答：()

7、关于如何避免缓冲区溢出带来的程序风险，下述错误的做法为？

- A. 编程时定义大的缓冲区数组
- B. 编程时避免使用 `gets`，而采用 `fgets`
- C. 程序运行时随机化栈的偏移地址
- D. 在硬件级别引入不可执行代码段的机制

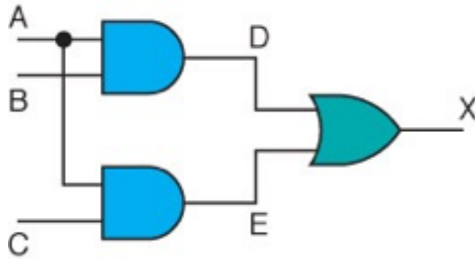
答：()

8、对简单的 `switch` 语句常采用跳转表的方式实现，在 x86-64 系统中，下述最有可能正确的 `switch` 分支跳转汇编指令为哪个？

- A. `jmp .L3(, %eax, 4)`
- B. `jmp .L3(, %eax, 8)`
- C. `jmp *.L3(, %eax, 4)`
- D. `jmp *.L3(, %eax, 8)`

答：()

9、对应下述组合电路的正确 HCL 表达式为



- A. `Bool X = (A || B) && (A || C)`
- B. `Bool X = A || (B && C)`
- C. `Bool X = A && (B || C)`
- D. `Bool X = A || B || C`

答：()

10、若处理器实现了三级流水线，每一级流水线实际需要的运行时间分别为 2ns、2ns 和 1ns，则此处理器不停顿地执行完毕 10 条指令需要的时间为：

- A. 21ns
- B. 22ns
- C. 23ns
- D. 24ns

答：()

11、关于 RISC 和 CISC 的描述，正确的是：

- A. CISC 指令系统的指令编码可以很短，例如最短的指令可能只有一个字节，因此 CISC 的取指部件设计会比 RISC 更为简单。
- B. CISC 指令系统中的指令数目较多，因此程序代码通常会比较长；而 RISC 指令系统中通常指令数目较少，因此程序代码通常会比较短。
- C. CISC 指令系统支持的寻址方式较多，RISC 指令系统支持的寻址方式较少，因此用 CISC 在程序中实现访存的功能更容易。
- D. CISC 机器中的寄存器数目较少，函数参数必须通过栈来进行传递；RISC 机器中的寄存器数目较多，只需要通过寄存器来传递参数。

答：()

12、关于流水线技术的描述，正确的是：

- A. 指令间数据相关引发的数据冒险，一定可以通过暂停流水线来解决。
- B. 流水线技术不仅能够提高执行指令的吞吐率，还能减少单条指令的执行时间。
- C. 增加流水线的级数，一定能获得性能上的提升。
- D. 流水级划分应尽量均衡，不均衡的流水线会增加控制冒险。

答：()

13、下面关于程序性能的说法中，哪种是正确的？

- A. 处理器内只要有多个功能部件空闲，就能实现指令并行，从而提高程序性能。
- B. 同一个任务采用时间复杂度为 $O(\log N)$ 算法一定比采用复杂度为 $O(N)$ 算法的执行时间短
- C. 转移预测总是能带来好处，不会产生额外代价，对提高程序性能有帮助。
- D. 增大循环展开 (loop unrolling) 的级数，有可能降低程序的性能 (即增加执行时间)

答：()

14、仅考虑以下代码，哪些程序优化总是被编译器自动进行？（假设 `int i, int j, int A[N], int B[N], int m, int *p` 都是局部变量，`N` 是一个整数型常量，`int foo(int)` 是一个函数）

优化前	优化后
A. <pre>for (j = 0 ; j < N ; j ++) B[i] *= A[j];</pre>	<pre>int temp = B[i]; for (j= 0 ; j < N ; j ++) temp *= A[j]; B[i] = temp;</pre>
B. <pre>for (j = 0 ; j < N ; j ++) m + = i*N*j;</pre>	<pre>int temp = i*N; for (j= 0 ; j < N ; j ++) m + = temp * j;</pre>
C. <pre>i = foo(N); j = foo(N); if (*p != 0) m = j ;</pre>	<pre>j = foo(N); if (*p != 0) m = j ;</pre>
D. <pre>for (j = 0 ; j < foo(N) ; j ++) m ++;</pre>	<pre>int temp = foo(N); for (j= 0 ; j < temp ; j ++) m ++;</pre>

答：()

15、以下关于存储结构的讨论，那个是正确的

- A. 增加额外一级存储，数据存取的延时一定不会下降
- B. 增加存储的容量，数据存取的延时一定不会下降
- C. 增加额外一级存储，数据存取的延时一定不会增加
- D. 以上选项都不正确

答：()

16、关于 cache 的 miss rate，下面那种说法是错误的。

- A. 保持 E 和 B 不变，增大 S，miss rate 一定不会增加
- B. 保持总容量和 B 不变，提高 E，miss rate 一定不会增加
- C. 保持总容量和 E 不变，提高 B，miss rate 一定不会增加
- D. 如果不采用“LRU”，使用“随机替换策略”，miss rate 可能会降低

答：()

得分

第二题 (10 分)

1) 假设下列 unsigned 和 int 数均为 5 位 (有符号整型用补码运算表示), 在下表中填入正确答案 (每空 1 分, 共 6 分)

```
int y = - 7;
unsigned z = y;
```

	Decimal Representation	Binary Representation
z		
$y - z$		
TMin		

2) 请按 IEEE 浮点标准的单精度浮点数表示下表中的数值, 首先写出形如 $(-1)^s \times M \times 2^E$ 的表达式, 然后给出十六进制的表示。(每格 1 分, 共 4 分)

注: 单精度浮点数的字段划分如下:

符号位 (s): 1-bit; 阶码字段 (exp): 8-bit; 小数字段 (frac): 23-bit; 偏置值 (bias): 127。

Value	$(-1)^s \times M \times 2^E, 1 \leq M < 2$	Hex representation
0.375		
-12.5		

得分

第三题（10 分）

阅读以下代码。假设代码运行在 IA32 的计算机上，字长为 4，请给出各个变量在内存中的十六进制字节表示（地址从小到大）。

注意 tiny_float 是一种 8 位的浮点数，1 个符号位，4 个指数位，3 个尾数位。

```
int main()
{
    int a = 0x15213;
    unsigned char b = ((char)-5);
    tiny_float c = 19;
    float d;
    if (b < 4)
        d = -0.96875;
    else
        d = 769;
}
```

答：

得分

第四题（10 分）

一个函数如下，其中部分代码被隐去，请通过gdb调试信息补全代码（4分）。

```
int f(int n, int m) {
    if (m > 0) {
        if (_____) {
            int r = _____;
            return _____;
        }
        else if (_____) {
            return 1;
        }
    }
    return 0;
}
```

如下是通过“gcc -g -O2”命令编译后，在gdb中通过“disas f”命令得到的反汇编代码，其中有两个汇编指令不全，请补全这两条汇编指令（2分）。

```
0x00000000004004e0 <f+0>:      mov     %rbx,-0x10(%rsp)
0x00000000004004e5 <f+5>:      mov     _____
0x00000000004004ea <f+10>:     xor     %eax,%eax
0x00000000004004ec <f+12>:     sub     $0x10,%rsp
0x00000000004004f0 <f+16>:     test    %esi,%esi
0x00000000004004f2 <f+18>:     mov     %edi,%ebp
0x00000000004004f4 <f+20>:     mov     %esi,%ebx
0x00000000004004f6 <f+22>:     jle     0x400513 <f+51>
0x00000000004004f8 <f+24>:     cmp     $0x1,%edi
0x00000000004004fb <f+27>:     jle     0x400521 <f+65>
0x00000000004004fd <f+29>:     lea     -0x1(%rbp),%edi
0x0000000000400500 <f+32>:     callq  0x4004e0 <f>
0x0000000000400505 <f+37>:     lea     -0x1(%rax,%rbx,1),%edx
0x0000000000400509 <f+41>:     mov     %edx,%eax
0x000000000040050b <f+43>:     sar     $0x1f,%edx
0x000000000040050e <f+46>:     idiv    %ebp
0x0000000000400510 <f+48>:     lea     0x1(%rdx),%eax
```

```

0x00000000000400513 <f+51>:      mov      _____
0x00000000000400517 <f+55>:      mov      0x8(%rsp),%rbp
0x0000000000040051c <f+60>:      add      $0x10,%rsp
0x00000000000400520 <f+64>:      retq
0x00000000000400521 <f+65>:      sete    %al
0x00000000000400524 <f+68>:      movzbl  %al,%eax
0x00000000000400527 <f+71>:      jmp     0x400513 <f+51>

```

已知在调用函数f(4, 3)时，我们在函数f中指令retq处设置了断点，下面列出的是程序在第一次运行到断点处暂停时时，相关通用寄存器的值。请根据你对函数及其汇编代码的理解，填写当前栈中的内容。如果某些内存位置处内容不确定，请填写x。（4分）

```

rax      0x1
rbx      0x3
rcx      0x3
rdx      0x309c552970
rsi      0x3
rdi      0x1
rbp      0x2
rsp      0x7fffffff340
rip      0x400520

```

0x7fffffff38c	
0x7fffffff388	
0x7fffffff384	
0x7fffffff380	
0x7fffffff37c	
0x7fffffff378	
0x7fffffff374	
0x7fffffff370	
0x7fffffff36c	
0x7fffffff368	
0x7fffffff364	
0x7fffffff360	
0x7fffffff35c	
0x7fffffff358	
0x7fffffff354	
0x7fffffff350	
0x7fffffff34c	
0x7fffffff348	
0x7fffffff344	
0x7fffffff340	
0x7fffffff33c	
0x7fffffff338	
0x7fffffff334	
0x7fffffff330	
0x7fffffff32c	
0x7fffffff328	
0x7fffffff324	
0x7fffffff320	

得分

第五题（8 分）

阅读下面的汇编代码，根据汇编代码填写 C 代码中缺失的部分，然后描述该程序的功能。

pushl %ebp	int fun(_____x) {
movl %esp,%ebp	int bit_sum = 0;
movl \$0x0, %ecx	while (_____) {
cmpl \$0x0, 8(%ebp)	_____;
jle .L1	_____;
.L2	}
movl \$0x0, %edx	if (_____)
movl 8(%ebp), %eax	return 1;
divl \$0x0a	else
addl %edx, %ecx	return 0;
movl %eax, 8(%ebp)	
cmpl \$0x0, 8(%ebp)	
jg .L2	
.L1	
movl 0x0, %edx	
movl %ecx, %eax	}
divl 0x3	
cmpl 0x0, %edx	
jne .L3	
movl 0x1, %eax	
jmp .L4	
.L3	
movl 0x0, %eax	
.L4	

答：

得分

第六题（10 分）

请分析Y86 ISA中新加入的一条指令：caddXX，条件加法。其功能可以参考add和cmovXX两条指令。

caddXX	C	fn	rA	rB
--------	---	----	----	----

若在教材所描述的SEQ处理器上执行这条指令，请按下表填写每个阶段进行的操作。需说明的信号包括：icode, ifun, rA, rB, valA, valB, valC, valE, valP, Cnd; the register file R[], data memory M[], Program counter PC, condition codes CC。其中对存储器的引用必须标明字节数。如果在某一阶段没有任何操作，请填写none指明。

Stage	caddXX rA, rB
Fetch	
Decode	
Execute	
Memory	
Write back	
PC update	

得分

第七题（10 分）

如下是使用 C 语言描述的链表结构的声明，链表的结尾使用空指针来表示。同时使用函数 `int length (List *p)` 来计算链表的长度。为简化起见，假设该链表是非循环的。

```
typedef struct LIST {
    struct LIST *next;
    int data;
} List;
```

- 1) 函数 `count_pos1` 用来计算链表中 `data` 为正数的元素个数，并将结果存放在地址 `k`。以下的程序可能存在问题导致效率很低或程序出错，请指出并修改。

（4 分）

```
void count_pos1 (List *p, int *k) {
    int i;
    for (i = 0; i < length(p); i++) {
        if (p->data > 0)
            *k++;
        p = p->next;
    }
}
```

- 2) 为提高程序性能, 可以考虑删除变量 `i` 以消除函数调用。请修改上述程序达到该目的。(2 分)

- 3) 上述程序内层循环的汇编片段如下所示。假设该链表不为空且大部分数据都为正数, 转移预测全部正确, 设计中有足够多的部件来实现指令并行。其中访存操作全部 `cache` 命中, 时延为 3 `cycle`, 其他指令时延为 1`cycle`。请计算以下程序的 CPE 下限, 并给出文字说明。(4 分)

```
.L1:
    movl    4(%eax), %ecx
    testl   %ecx, %ecx
    jle     .L2
    incl    %edx

.L2:
    movl    (%eax), %eax
    testl   %eax, %eax
    jne     .L1
```

得分

第八题（10 分）

假设存在一个能够存储四个 Block 的 Cache，每一个 Block 的长度为 2Byte。假设内存空间大小共是 16Byte，即内存空间地址长度一共是 4bit，可访问地址为 (0~15)，数据访问地址序列如下所示，访问数据单位是 Byte，默认替换策略是 LRU。

2 3 10 9 6 8

1) 如果 Cache 的结构是下图所示 (S=2, E=2)，请在下图空白处填入访问上述六次数据访问后 Cache 的状态。注：用 [0-1] 表示地址 0 至 1 上对应的数据（4 分）

	V	TAG	Block	V	TAG	Block
set0						
set1						

2) 这六次数据访问一共产生了多少次 Miss _____（2 分）

3) 如果 Cache 的替换策略改成 MRU（即，最近使用的数据被替换出去），请在下图空白处填入访问上述六次数据访问后 Cache 的状态（2 分）。

	V	TAG	Block	V	TAG	Block
set0						
set1						

4) 这六次数据访问一共产生了多少次 Miss _____（2 分）

北京大学信息科学技术学院考试试卷

考试科目：计算机系统导论 姓名：_____ 学号：_____

考试时间：2015 年 11 月 09 日 小班教师：_____

题号	一	二	三	四	五	总分
分数						
阅卷人						

北京大学考场纪律

1、考生进入考场后，按照监考老师安排隔位就座，将学生证放在桌面上。无学生证者不能参加考试；迟到超过 15 分钟不得入场。在考试开始 30 分钟后方可交卷出场。

2、除必要的文具和主考教师允许的工具书、参考书、计算器以外，其它所有物品（包括空白纸张、手机、或有存储、编程、查询功能的电子用品等）不得带入座位，已经带入考场的必须放在监考人员指定的位置。

3、考试使用的试题、答卷、草稿纸由监考人员统一发放，考试结束时收回，一律不准带出考场。若有试题印制问题请向监考教师提出，不得向其他考生询问。提前答完试卷，应举手示意请监考人员收卷后方可离开；交卷后不得在考场内逗留或在附近高声交谈。未交卷擅自离开考场，不得重新进入考场答卷。考试结束时间到，考生立即停止答卷，在座位上等待监考人员收卷清点后，方可离场。

4、考生要严格遵守考场规则，在规定时间内独立完成答卷。不准交头接耳，不准偷看、夹带、抄袭或者有意让他人抄袭答题内容，不准接传答案或者试卷等。凡有违纪作弊者，一经发现，当场取消其考试资格，并根据《北京大学本科考试工作与学术规范条例》及相关规定严肃处理。

5、考生须确认自己填写的个人信息真实、准确，并承担信息填写错误带来的一切责任与后果。

学校倡议所有考生以北京大学学生的荣誉与诚信答卷，共同维护北京大学的学术声誉。

以下为试题和答题纸，共 页。

得分

第一题 单项选择题（每小题 1 分，共 20 分）

注：请将选择题答案填写在下表中

题号	1	2	3	4	5	6	7	8	9	10
答案										
题号	11	12	13	14	15	16	17	18	19	20
答案										

1. 给定一个实数，会因为该实数表示成单精度浮点数而发生误差。不考虑 NaN 和 Inf 的情况，该绝对误差的最大值为

A. 2^{103}
 B. 2^{104}
 C. 2^{230}
 D. 2^{231}

参考信息：单精度浮点数阶码 8 位，尾数 23 位

2. 以下说法错误的是

A. 负数加上负数结果可能为正数
 B. 正数加上正数结果可能为负数
 C. 用&和~可以表示所有的逻辑与或非操作
 D. 用&和|可以表示所有的逻辑与或非操作

3. 在 32 位平台上，按 C90 标准以下语句中，结果为假的是

A. `return INT_MIN < INT_MAX;`
 B. `return -2147483648 < 2147483647;`
 C. `int a = -2147483648; return a < 2147483647;`
 D. `return -2147483647-1 < 2147483647;`

参考信息：

C90 的转换顺序：int -> long -> unsigned -> unsigned long
 $2^{31} = 2147483648$

4. 关于浮点数，以下说法正确的是
- A. 给定任意浮点数 a, b 和 x ，如果 $a > b$ 成立（求值为 1），则一定 $a+x > b+x$ 成立
 - B. 不考虑结果为 NaN、Inf 或运算过程发生溢出的情况，高精度浮点数一定得到比低精度浮点数更精确或相同的结果
 - C. 不考虑输入为 NaN、Inf 的情况，高精度浮点数一定得到比低精度浮点数更精确或相同的结果
 - D. 给定任意浮点数 a, b 和 x ，如果 $a > b$ 不成立（求值为 0），则一定 $a+x > b+x$ 不成立。
5. 已知下面的数据结构，假设在 Linux/IA32 下要求对齐，这个结构的总的大小是多少个字节？如果重新排列其中的字段，最少可以达到多少个字节？
- ```
struct {
 char a;
 double *b;
 double c;
 short d;
 long long e;
 short f;
};
```
- A. 32, 28
  - B. 36, 32
  - C. 28, 26
  - D. 26, 26
6. 下列寻址模式中，正确的是：
- A. (%eax, , 4)
  - B. (%eax, %esp, 3)
  - C. 123
  - D. \$1(%ebx, %ebp, 1)
7. 假设存储器按“大端法”存储数据对象，已知如下的 C 语言数据结构：union { char c[2]; int i; }; 当 c 的值为 0x01, 0x23 时，i 的值为：
- A. 0x0123
  - B. 0x2301
  - C. 0x01230000
  - D. 不确定



8. 假设某条 C 语言 switch 语句编译后产生了如下的汇编代码及跳转表:

|                     |           |
|---------------------|-----------|
| movl 8(%ebp), %eax  | .L7:      |
| subl \$48, %eax     | .long .L3 |
| cmpl \$8, %eax      | .long .L2 |
| ja .L2              | .long .L2 |
| jmp *.L7(, %eax, 4) | .long .L5 |
|                     | .long .L4 |
|                     | .long .L5 |
|                     | .long .L6 |
|                     | .long .L2 |
|                     | .long .L3 |

在源程序中, 下面的哪些(个)标号出现过:

- A. '2', '7'
- B. 1
- C. '3'
- D. 5

9. x86 体系结构中, 下面哪个说法是正确的?

- A. leal 指令只能够用来计算内存地址
- B. x86\_64 机器可以使用栈来给函数传递参数
- C. 在一个函数内, 改变任一寄存器的值之前必须先将其原始数据保存在栈内
- D. 判断两个寄存器中值大小关系, 只需要 SF 和 ZF 两个条件码

10. 下列的指令组中, 哪一组指令只改变条件码, 而不改变寄存器的值?

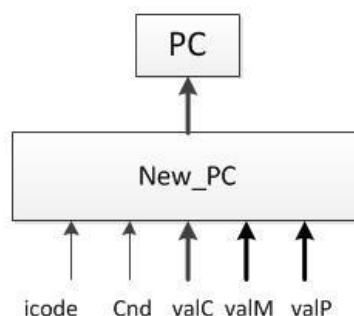
- A. CMP, SUB
- B. TEST, AND
- C. CMP, TEST
- D. LEAL, CMP

11. 下面有关指令系统设计的描述正确的是:

- A. 采用 CISC 指令比 RISC 指令代码更长。
- B. 采用 CISC 指令比 RISC 指令运行时间更短
- C. 采用 CISC 指令比 RISC 指令译码电路更加复杂
- D. 采用 CISC 指令比 RISC 指令的流水线吞吐更高

12. 一个功能模块包含组合逻辑和寄存器，组合逻辑单元的总延迟是 100ps，单个寄存器的延时是 20ps，该功能模块执行一次并保存执行结果，理论上能达到的最短延时和最大吞吐分别是多少？
- A. 20ns, 50GIPS  
B. 120ns, 50GIPS  
C. 120ns, 10GIPS  
D. 20ps, 10GIPS
13. 关于流水线技术的描述，错误的是：
- A. 流水线技术能够提高执行指令的吞吐率，但也同时增加单条指令的执行时间。  
B. 减少流水线的级数，能够减少数据冒险发生的几率。  
C. 指令间数据相关引发的数据冒险，都可以通过 data forwarding 来解决。  
D. 现代处理器支持一个时钟内取指、执行多条指令，会增加控制冒险的开销。
14. 在 Y86 的 SEQ 实现中，PC (Program Counter, 程序计数器) 更新的逻辑结构如下图所示，请根据 HCL 描述为①②③④选择正确的数据来源。

```
Int new_pc = [
 # Call.
 Icode = ICALL : ①;
 # Taken branch.
 Icode = IJXX && Cnd : ②;
 # Completion of RET instruction.
 Icode = IRET : ③;
 # Default.
 1 : ④;
]
```



其中：Icode 为指令类型，Cnd 为条件是否成立，valC 表示指令中的常数值，valM 表示来自返回栈的数据，valP 表示 PC 自增。

- A. valC, valM, valP, valP  
B. valC, valC, valP, valP  
C. valC, valC, valM, valP  
D. valM, valC, valC, valP
15. 下面关于存储器的说法，错误的是\_\_\_\_\_
- A. SDRAM 的速度比 FPM DRAM 快  
B. SDRAM 的 RAS 和 CAS 请求共享相同的地址引脚

- C. 磁盘的寻道时间和旋转延迟大致在一个数量级
- D. 固态硬盘的随机读写性能基本相当

16. 某磁盘的旋转速率为 7200 RPM，每条磁道平均有 400 扇区，则一个扇区的平均传送时间为\_\_\_\_\_

- A. 0.02 ms
- B. 0.01 ms
- C. 0.03 ms
- D. 0.04 ms

17. 某高速缓存 (E=2, B=4, S=16)，地址宽度 14，当引用地址 0x9D28 处的 1 个字节时，tag 位应为：

- A. 01110100
- B. 001110000
- C. 1110000
- D. 10011100

18. 关于高速缓存的说法正确的是 \_\_\_\_\_

- A. 直写 (write through) 比写回 (write back) 在电路实现上更复杂
- B. 固定的高速缓存大小，较大的块可提高时间局部性好的程序的命中率
- C. 随着高速缓存组相联度的不断增大，失效率不断下降
- D. 以上说法全不正确

19. 关于局部性 (locality) 的描述，不正确的是：

- A. 循环通常具有很好的时间局部性
- B. 循环通常具有很好的空间局部性
- C. 数组通常具有很好的时间局部性
- D. 数组通常具有很好的空间局部性

20. 以下哪些程序优化编译器总是可以自动进行？（假设 int i, int j, int A[N], int B[N], float m 都是局部变量，N 是一个整数型常量，int foo(int) 是一个函数）

|    | 优化前                                       | 优化后                                                             |
|----|-------------------------------------------|-----------------------------------------------------------------|
| A. | for (j = 0 ; j < N ; j ++)<br>m += i*N*j; | int temp = i*N;<br>for (j = 0 ; j < N ; j ++)<br>m += temp * j; |

|    |                                                                       |                                                                                           |
|----|-----------------------------------------------------------------------|-------------------------------------------------------------------------------------------|
| B. | <pre> for (j = 0 ; j &lt; N ; j ++)     B[i] *= A[j]; </pre>          | <pre> int temp = B[i]; for (j= 0 ; j &lt; N ; j ++)     temp *= A[j]; B[i] = temp; </pre> |
| C. | <pre> for (j = 0 ; j &lt; N ; j ++)     m = (m + A[j]) + B[j]; </pre> | <pre> for (j = 0 ; j &lt; N ; j ++)     m = m + (A[j] + B[j]); </pre>                     |
| D. | <pre> for (j = 0 ; j &lt; foo(N) ; j ++)     m++; </pre>              | <pre> int temp = foo(N); for (j= 0 ; j &lt; temp ; j ++)     m++; </pre>                  |

|    |
|----|
| 得分 |
|    |

## 第二题 (20 分)

1. 对于下面的每一个表达式, 请选择以下选项中的一个或多个 (即“不定项”), 使得该表达式恒成立, 如果没有满足条件的选项则选 E。

A. <      B. >      C. ==      D. !=      E. none

题目中出现的变量定义如下 (浮点数保证不是 NaN 或者 Inf):

```
int x, y;
unsigned ux = x;
double d;
```

- 1) 如果  $x > 0$ , 则  $x + 1$  \_\_\_\_ 0
- 2) 如果  $x > y$ , 则  $ux$  \_\_\_\_  $y$
- 3) 如果  $((x \ll 31) \gg 31) < 0$ , 则  $x \& 1$  \_\_\_\_ 0
- 4) 如果  $((\text{unsigned char})x \gg 1) < 64$ , 则  $(\text{char})x$  \_\_\_\_ 0
- 5) 如果  $d < 0$ , 则  $d * 2$  \_\_\_\_ 0
- 6) 如果  $d < 0$ , 则  $d * d$  \_\_\_\_ 0
- 7)  $x \wedge y \wedge (\sim x) - y$  \_\_\_\_  $y \wedge x \wedge (\sim y) - x$
- 8)  $((!!ux) \ll 31) \gg 31$  \_\_\_\_  $((!!x) \ll 31) \gg 31$

2. 考虑一种 12-bit 长的浮点数, 此浮点数遵循 IEEE 浮点数格式, 请回答下列问题。

注: 浮点数的字段划分如下:

符号位 (s): 1-bit; 阶码字段 (exp): 4-bit; 小数字段 (frac): 7-bit。

1) 请写出在下列区间中包含多少个用上面规则精确表示的浮点数

A:  $[1, 2)$  \_\_\_\_\_

B:  $[2, 3)$  \_\_\_\_\_

2) 请写出下面浮点数的二进制表示

| 数字                | 二进制表示 |
|-------------------|-------|
| 最小的正规格化数          |       |
| 最大的非规格化数          |       |
| $17\frac{1}{16}$  |       |
| $-\frac{1}{8192}$ |       |

|                 |  |
|-----------------|--|
| $20\frac{3}{8}$ |  |
| $-\infty$       |  |

|    |
|----|
| 得分 |
|    |

### 第三题（20 分）

1. 考虑下面的union的声明，回答后面的问题。

```
union ELE {
 struct {
 int x;
 int *p;
 }e1;
 struct {
 union ELE * next;
 int y;
 }e2;
};
```

(注：32位机器)

这个union总共大小为多少\_\_\_\_\_字节

2. 假设编译器为process的主体产生了如下代码，请补充完整下面的过程：  
（只有一个不需要任何强制类型转换且不违反任何类型限制的答案）

```
movl 8(%ebp),%eax
movl (%eax),%ecx
movl 4(%ecx),%edx
movl (%edx),%edx
subl 4(%eax),%edx
movl %edx, (%ecx)
```

```
void process(union ELE * up)
{
 up->_____ = _____ - _____;
}
```

3. 请查看下文完成如下功能的汇编代码，定位错误语句并进行更正：（6 分）

给出  $n$  (在 `%ebp+8` 位置,  $n \geq 1$ ), `up` (在 `%ebp+12` 位置, `ELE*` 类型), 假设以 `*up` 为头元素 (设 `*up` 为第 0 个), 由声明中的 `next` 连接形成了一个链表, 请将第  $n$  个元素 (假设链表足够长) 的 `x` 的值放入 `%eax` 中

x86 代码

```
xorl %ecx,%ecx
movl 8(%edx),%ebp
movl 12(%ebp),%eax
```

LOOP:

```
movl (%eax),%eax
add $1,%ecx
test %ecx,%edx
jne LOOP
```

```
movl (%eax),%eax
```

y86 代码

```
mrmovl 8(%edx),%ebp
mrmovl 12(%ebp),%eax
irmovl $1,%ecx
```

LOOP:

```
mrmovl (%eax),%eax
test %ecx,%edx
jne LOOP
```

```
mrmovl (%eax),%eax
```

4. 阅读下列代码, 回答后面的问题

```
typedef struct {
 short x[A][B];
 int y;
}str1;
```



```

typedef struct {
 char array[B];
 int t;
 short s[B];
 int u;
}str2;

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

```

(short 以 2 字节计算)

GCC 为 setVal 的主体产生下面的代码：

```

movl 12(%ebp),%eax
movl 28(%eax),%edx
addl 8(%eax),%edx
movl 8(%ebp),%eax
movl %edx,44(%eax)

```

请直接写出A和B的值各是多少？

|    |
|----|
| 得分 |
|    |

第四题（20 分）

请分析Y86 ISA中新加入的一条指令：NewJE，其格式如下。

|       |   |   |    |    |      |
|-------|---|---|----|----|------|
| NewJE | C | 0 | rA | rB | Dest |
|-------|---|---|----|----|------|

其功能为：如果 $R[rA] = R[rB]$ ，则跳转到Dest继续执行，否则顺序执行。

1. 若在教材所描述的SEQ处理器上执行这条指令，请按下表补全每个阶段的操作。需说明的信号可能会包括：icode, ifun, rA, rB, valA, valB, valC, valE, valP, Cnd; the register file R[], data memory M[], Program counter PC, condition codes CC。其中对存储器的引用必须标明字节数。如果在某一阶段没有任何操作，请填写none指明。

| Stage      | NewJE rA, rB, Dest                                                                                        |
|------------|-----------------------------------------------------------------------------------------------------------|
| Fetch      | icode:ifun $\leftarrow M_1[PC]$<br>rA:rB $\leftarrow M_1[PC+1]$<br>valC $\leftarrow$<br>valP $\leftarrow$ |
| Decode     | valA $\leftarrow R[rA]$<br>valB $\leftarrow R[rB]$                                                        |
| Execute    | valE $\leftarrow$                                                                                         |
| Memory     |                                                                                                           |
| Write back | none                                                                                                      |
| PC update  | PC $\leftarrow$ valE==0 ?        :                                                                        |

2. 若在教材所描述的PIPE处理器上执行NewJE指令，如果跳转条件不满足，一共会错误执行\_\_\_\_条指令。

为了减小错误预测的代价，现将教材所描述的PIPE处理器做如下改进：在

Decode阶段增加一个比较器，用于判断  $(R[rA] = R[rB])$  条件，比较器的输出信号为d\_equal。如果相等，则d\_equal = 1，反之 d\_equal = 0。

此时，如果执行NewJE指令时跳转条件不满足，一共会错误执行\_\_\_\_条指令。

3.在教材所描述的PIPE处理器上执行JXX指令时，发生转移预测错误的判断条件和各级流水线寄存器的控制信号如下所示：

| Condition           | Trigger                 |
|---------------------|-------------------------|
| Mispredicted Branch | E_icode = IJXX & !e_Cnd |

| Condition           | F      | D      | E      | M      | W      |
|---------------------|--------|--------|--------|--------|--------|
| Mispredicted Branch | normal | bubble | bubble | normal | normal |

在第（2）小题所述的改进后的处理器上执行 NewJE 指令，发生转移预测错误的判断条件和各级流水线寄存器的控制信号应如何设置？

| Condition           | Trigger    |
|---------------------|------------|
| Mispredicted Branch | = INewJE & |

| Condition           | F | D | E | M      | W      |
|---------------------|---|---|---|--------|--------|
| Mispredicted Branch |   |   |   | normal | normal |

4.在第（2）小题所述的改进后的处理器上执行如下代码，

```

0x000: mrmovl 0(%eax), %edx
0x006: NewJE %edx, %eax, t
0x00c: irmovl $1, %eax # Fall through
0x012: nop
0x013: nop

```

```
0x014: nop
0x015: halt
0x016: t:irmovl $3, %edx # Target (Should not execute)
0x01c: irmovl $4, %ecx # Should not execute
0x022: irmovl $5, %edx # Should not execute
```

会发生 load-use 和 misprediction 组合的 hazard 情况，如下图所示



请问此时，各级流水线寄存器的控制信号应如何设置？

| Condition     | F | D | E | M      | W      |
|---------------|---|---|---|--------|--------|
| Combination C |   |   |   | normal | normal |

5.在教材 PIPE 处理器设计中，data memory 实际是高速缓存(cache)。假设在执行上述(4)中代码时，0x000 指令中的 0(%eax)地址中的数据不在 data memory 中，则 data memory 会将输出信号 m\_datamiss 置为 1，直到数据从内存中取回到 data memory，再将 m\_datamiss 置为 0。(m\_datamiss 的默认值为 0)

这种情况的判断条件如下，请问各级流水线寄存器的控制信号应如何设置？

| Condition | Trigger                                     |
|-----------|---------------------------------------------|
| Data Miss | M_icode in { IMRMOVL, IPOPL } && m_datamiss |

| Condition | F | D | E | M | W |
|-----------|---|---|---|---|---|
| data miss |   |   |   |   |   |



|    |
|----|
| 得分 |
|    |

第五题（20 分）

1. 仔细阅读下面的程序，根据条件回答下列各题

- 地址宽度为 7，数组的起始地址为 0x1000000
- Block Size = 4 Byte, Set = 4, 两路组相连.
- 替换算法位 LRU（最近最少使用）

```
#define LENGTH 8
void clear4x4 (char array[LENGTH][LENGTH]) {
 int row, col ;
 for (col = 0 ; col < 4; col++) {
 for (row = 0; row < 4 ; row++) {
 array[row] [col] = 0;
 }
 }
}
```

- 1) 以上程序执行会引起多少次失效？  
\_\_\_\_\_
- 2) 如果 LENGTH 改为 16，会引起多少次失效？  
\_\_\_\_\_
- 3) 如果 LENGTH 变为 17，与 2) 相比，下面描述正确的是：\_\_\_\_，会引起\_\_\_\_次失效。  
 A) 16×16 比 17×17 产生更多的失效次数  
 B) 16×16 和 17×17 产生的失效次数相同  
 C) 16×16 比 17×17 产生更少的失效次数
- 4) 请画出 3) 运行后 cache 中 set0 和 set1 的最终状态。

| V | Tag | Data | V | Tag | Data |
|---|-----|------|---|-----|------|
| 1 |     |      | 1 |     |      |
| 1 |     |      | 1 |     |      |

2. 改变假设条件，回答下列各题

- 地址宽度为 8，数组的起始地址为  $0 \times 10000000$
- Cache 容量为 16 Byte，Block Size = 4 Byte，全相联 Cache
- 替换算法位 LRU（最近最少使用）

1) Tag 的位数为\_\_\_\_\_

2) 如果执行上述程序，当 LENGTH=8 时，会引起多少次失效？

3) 如果 LENGTH 改为 16，会引起多少次失效？

4) 如果 LENGTH 变为 17，与 3) 相比，下面描述正确的是：\_\_\_\_\_

- A)  $16 \times 16$  比  $17 \times 17$  产生更多的失效次数
- B)  $16 \times 16$  和  $17 \times 17$  产生的失效次数相同
- C)  $16 \times 16$  比  $17 \times 17$  产生更少的失效次数

5) 请画出 4) 执行后 cache 的最终状态

|   |     |      |
|---|-----|------|
| V | Tag | Data |
| 1 |     |      |
| V | Tag | Data |
| 1 |     |      |
| V | Tag | Data |
| 1 |     |      |
| V | Tag | Data |
| 1 |     |      |