1. 请完成下列数制转换

十进制	二进制	十六进制
167		
	1111 0011	
		0xAA

2. 请计算各表达式的结果，其中 char a = 0x96, b = 0xAA;

表达式	结果	表达式	结果
~ a		a | b	
~ b		a || b	
a & b		a ^ b	
a && b		!a	
!b		~ (a & b)	
-1 < 0U		a / b	

3. 已知 int x = 0x18205643; 变量 x 存储在地址 0x100 的位置上。请填写大端（Big Endian）系统中地址 **0x100~0x103** 处各字节的内容(十六进制)。

	0x100	0x101	0x102	0x103	
…					…

4. 已知 int y = 0x56192036; 变量 y 存储在地址 0x200 的位置上。请填写小端（Little Endian）系统中地址 **0x200~0x203** 处各字节内容(十六进制)。

	0x200	0x201	0x202	0x203	
…					…

5. 已知寄存器 **%eax** 和 **%ecx** 的值和内存中地址 **0x100~0x120** 处的数据（见下表），请计算以下指令中寄存器 **%edx** 的值。

内存			
地址	**值**	**地址**	**值**
0x100	0x306	0x110	0
0x104	0x80C	0x114	0x4
0x108	0xFFFFFFFF	0x118	0x5555
0x10c	0xCCCCCCCC	0x11c	0x2017

寄存器	**值**
%eax	0x100
%ecx	0x4

指令	**%edx**
movl %eax, %edx	
movl (%eax), %edx	
movl 4(%eax), %edx	
movl (%eax, %ecx), %edx	
movl 16(%eax, %ecx), %edx	
leal 4(%eax, %ecx, 2), %edx	

6. 请根据左侧的汇编指令，补全其所对应的 C 语言语句

```
/* xp at %ebp+8, yp at %ebp+12 */
  movl  8(%ebp), %edx
  movl  12(%ebp), %ecx
  movl  (%edx), %ebx
  movl  (%ecx), %eax
  movl  %eax, (%edx)
  movl  %ebx, (%ecx)
```

```
void func(int *xp, int *yp)
{
  int t0 = *xp;
  int t1 = *yp;
  ____________________;
  ____________________;
}
```

7. 请根据左侧的 x86-32 汇编指令，补全其所对应的 C 语言语句。

```
absdiff:
  pushl  %ebp
  movl   %esp, %ebp
  movl   8(%ebp), %edx  // x in %ebp+8
  movl   12(%ebp), %eax // y in %ebp+12
  cmpl   %eax, %edx
  jle    .L6            // less or equal
  subl   %eax, %edx
  movl   %edx, %eax
  jmp    .L7
.L6:
  subl %edx, %eax
.L7:
  popl %ebp
  ret
```

```
int absdiff(int x, int y)

{

    int result;

    if (x > y) {

      ____________________;

    } else {

      ____________________;

    }

    return result;

}
```

8. 根据链接器符号解析的规则，将代码中出现的符号填写到右侧对应的位置

```
extern int buf[];

int *bufp0 = &buf[0];
static int *bufp1;
void swap()
{
  int temp;
  bufp1 = &buf[1];
  temp = *bufp0;
  *bufp0 = *bufp1;
  *bufp1 = temp;
}
```

全局符号（Global）

外部符号（External）

内部符号（Local）

9. 请画出下面代码中的数组数据在内存中的布局。

```
#define ZLEN   3
#define PCOUNT 4
typedef int zip_dig[ZLEN];
zip_dig pgh[PCOUNT] = { {2, 0, 6}, {2, 1, 3 },
                        {2, 1, 7}, {2, 2, 1 } };
```

10. 请分别画出下面结构在字节对齐为 1，2，4 三种情况下在内存中的布局。

```
struct S1 {
    char c;
    int i[2];
    double v;
};
```

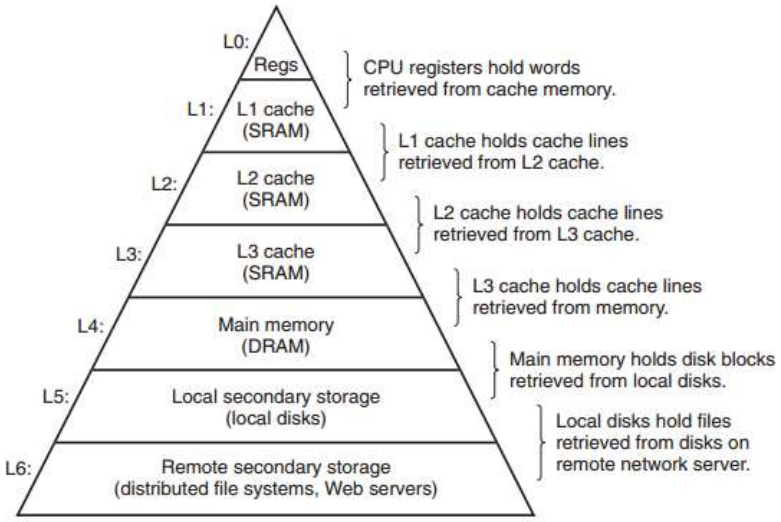
11. 请说明 C 语言中的 switch 语句在汇编语言水平上的主要实现思路。

12. 什么是中断？请说明中断的工作过程。

13. 请说明 call（过程调用）指令和 ret（过程返回）指令的使用方法和具体工作过程，并说明执行这些指令后相关寄存器的变化情况。

14. 什么是程序的局部性原理（locality）？

15. 请根据下图说明计算机系统中存储器层次结构的特点。



16. 请论述攻击缓冲区溢出漏洞所采用的主要手段，并从程序设计的角度以及编译器的角度讨论可以采取何种手段避免或防止缓冲区溢出攻击。

17. 请对以下程序链接问题进行说明和讨论：

（a） 什么是强符号（Strong Symbols）？什么是弱符号(Weak Symbols)？这两类符号在链接时遵循何种规则？

（b） 以下每组包含两个代码文件，请对以下各组代码文件在链接时可能出现的问题或存在的风险进行分析和讨论。

第一组

```
int x;
p1() {…}
```

```
p1() {…}
```

第二组

```
int x;
p1() {…}
```

```
int x;
p2() {…}
```

第三组

```
int x, y;
p1() {…}
```

```
double x;
p2() {…}
```

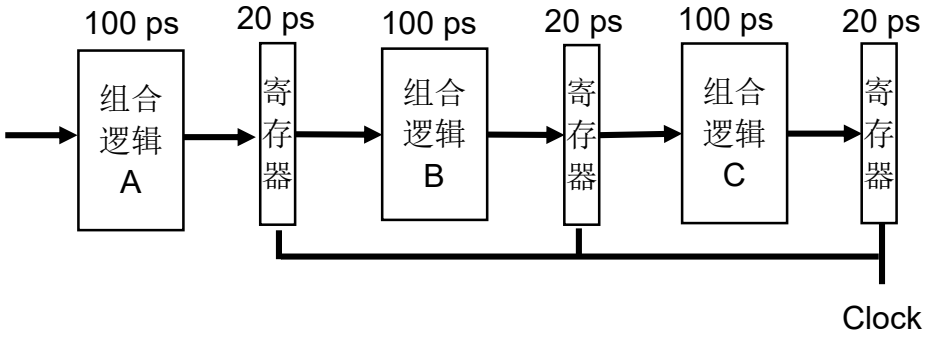
第四组

```
int x=7,  y=5;
p1() {…}
```

```
double x;
p2() {…}
```

18. 请说明陷阱/软中断（Trap）异常产生后的主要处理流程，并说明陷阱/软中断异常的主要用途。

19. 在某具有三级流水线结构的处理器中，每级流水线中的组合逻辑和寄存器的延迟时间如下图所示。



请计算：

（1）在流水线满负载条件下，求该处理器的指令吞吐量。

（2）如果组合逻辑 B 的延迟时间变为 360ps，其他部件延迟时间不变，求该处理器的指令吞吐量。

提示：吞吐量单位为 GIPS（十亿条指令/秒）；1ps = 10^{-12}s。

20. 综合分析题

某个具有内存管理单元（MMU）的计算机系统拥有以下特性：

（1）14-bit 的虚拟地址，12-bit 的物理地址，每个页包含 64 字节（Page Size）。

（2）后备缓冲表（TLB）采用 4 路组相连的方式进行组织，可以缓存 16 个页表项。

（3）高速缓存（Cache）采用直接映射的方式进行组织，缓存共分为 16 组，每组包含 1 个块，每个块大小为 4 个字节。高速缓存只用于缓存通用数据，不缓存页表。

（4）页表只有一级。

在当前时刻，系统中 TLB 的状态如下：

组索引	*Tag*	*PPN*	*Valid*	*Tag*	*PPN*	*Valid*	*Tag*	*PPN*	*Valid*	*Tag*	*PPN*	*Valid*
0	03	–	0	09	0D	1	00	–	0	07	02	1
1	03	2D	1	02	–	0	04	–	0	0A	–	0
2	02	–	0	08	–	0	06	–	0	03	–	0
3	07	–	0	03	0D	1	0A	34	1	02	–	0

页表状态如下(只列出了前 16 项)：

VPN	*PPN*	*Valid*
00	28	1
01	–	0
02	33	1
03	02	1
04	–	0
05	16	1
06	–	0
07	–	0

VPN	*PPN*	*Valid*
08	13	1
09	17	1
0A	09	1
0B	–	0
0C	–	0
0D	2D	1
0E	11	1
0F	0D	1

Cache 状态如下：

组索引	*Tag*	*Valid*	*Data*（0-4）			
0	19	1	99	11	23	11
1	15	0	–	–	–	–
2	1B	1	00	02	04	08
3	36	0	–	–	–	–
4	32	1	43	6D	8F	09
5	0D	1	36	72	F0	1D
6	31	0	–	–	–	–
7	16	1	11	C2	DF	03

组索引	*Tag*	*Valid*	*Data*（0-4）			
8	24	1	3A	00	51	89
9	2D	0	–	–	–	–
A	2D	1	93	15	DA	3B
B	0B	0	–	–	–	–
C	12	0	–	–	–	–
D	16	1	04	96	34	15
E	13	1	83	77	1B	D3
F	14	0	–	–	–	–

基于以上材料请作答：

（1） 请计算在 14-bit 的虚拟地址中，VPN 的位长、VPO 的位长。并计算在 12-bit 的物理地址中 PPN 的位长和 PPO 的位长。

（2） 当处理器加载虚拟地址为 0x036a 的 short 类型数据时，请说明 MMU 与 TLB 和 Cache 交互过程。并计算最终加载到的 short 类型数据的值。

（3） 当处理器加载虚拟地址为 0x0020 的 short 类型数据时，请说明在数据加载后 TLB 的变化情况。