1. 判断以下描述更符合 SRAM 还是 DRAM，还是都符合.

	SRAM	DRAM
(1)访问速度更快	✔	
(2)每比特需要的晶体管数目少		✔
(3)单位容量造价更便宜		✔
(4)常用作主存		✔
(5)需要定期刷新		✔
(6)断电后失去存储的信息	✔	✔
(7)支持随机访问	✔	✔

2. 已知一个双面磁盘有 2 个盘片、10000 个柱面，每条磁道有 400 个扇区，每个扇区容量为 512 字节，则它的存储容量是_**8.192**_GB

   2*2*10000*400*512 = 8,192,000,000 Byte = 8.192GB

3. 已知一个磁盘的平均寻道时间为 6ms，旋转速度为 7500RPM，那么它的平均访问时间大约为_**10**_ms

   6ms + 0.5*(60/7500*1000)ms = 10ms

4. 已知一个磁盘每条磁道平均有 400 个扇区，旋转速度为 6000RPM，那么它的平均传送时间大约为_**0.025**_ms

5. 考虑如下程序

```
for (int i = 0; i < n; i++) {
   B[i] = 0;
   for (int j = 0; j < m; j++)
      B[i] += A[i][j];
}
```

判断以下说法的正确性

(**N**) 对于数组 A 的访问体现了时间局部性。

(**Y**) 对于数组 A 的访问体现了空间局部性。

(**Y**) 对于数组 B 的访问体现了时间局部性。

(**Y**) 对于数组 B 的访问体现了空间局部性。

6. 高速缓存

   a) 一个容量为 8K 的直接映射高速缓存，每一行的容量为 32B，那么它有_**256**_组，每组有_**1**_行。

   b) 一个容量为 8K 的全相联映射高速缓存，每一行的容量为 32B，那么它有_**1**_组，每组有_**256**_行。

   c) 一个容量为 8K 的 4 路组相联映射高速缓存，每一行的容量为 32B，那么它有_**64**_组，每组有_**4**_行。

   d) 一个容量为 16K 的 4 路组相联高速缓存，每一行的容量为 64B，那么一个 16 位地址 0xCAFE 应映射在第_**43**_组内。

7. 判断以下说法的正确性

(**Y**)	保持块大小与路数不变，增大组数，命中率一定不会降低。
(**N**)	保持总容量与块大小不变，增大路数，命中率一定不会降低。
(**N**)	保持总容量与路数不变，增大块大小，命中率一定不会降低。
(**Y**)	使用随机替换代替 LRU，期望命中率可能会提高。

8. 有以下定义：

```
// 以下都是局部变量
int i, j, temp, ians;
int *p, *q, *r;
double dans;
// 以下都是全局变量
int iMat[100][100];
double dMat[100][100];
// 以下都是函数
int foo(int x);
```

如果将下列左侧代码优化为右侧代码，有没有可能有副作用？如果有请说明.

(1)	`ians = 0;`
	`for (j = 0; j < 100; j++)`
	`  for (i = 0; i < 100; i++)`
	`  ians += iMat[i][j];`
		`ians = 0;`
		`for (i = 0; i < 100; i++)`
		`  for (j = 0; j < 100; j++)`
		`  ians += iMat[i][j];`

(2)	`dans = 0;`
	`for (j = 0; j < 100; j++)`
	`  for (i = 0; i < 100; i++)`
	`  dans += dMat[i][j];`
		`dans = 0;`
		`for (i = 0; i < 100; i++)`
		`  for (j = 0; j < 100; j++)`
		`  dans += dMat[i][j];`

(3)	`for (i = 0;i < foo(100);i++)`
	`  ians += iMat[0][i];`
		`temp = foo(100);`
		`for (i = 0; i < temp; i++)`
		`  ians += iMat[0][i];`

(4)	`*p += *q;` `*p += *r;`	`temp = *q + *r;` `*p += temp;`

答:
(1) 会
(2) 不会，因为浮点数不能结合
(3) 不会，因为foo 可能有副作用
(4) 不会，如果 pqr 指向同一个元素那么两个运算不等价

9. 假设已有声明 `int i, int sum, int *p, int *q, int *r, const int n = 100, float a[n], float b[n], float c[n], int foo(int), void bar()`, 以下哪种优化编译器总是可以进行?

A	`for(i = 0; i < n; ++i){`
	`  a[i] += b[i];`
	`  a[i] += c[i];`
	`}`
		`float temp;`
		`for(i = 0; i < n; ++i){`
		`  temp = b[i] + c[i];`
		`  a[i] += temp;`
		`}`

B	`*p += *q;`
	`*p += *r;`
		`int temp;`
		`temp = *q + *r;`
		`*p += temp;`

C	`for(i = 0; i < n; ++i)` `  sum += i*4;`	`int N = n * 4;` `for(i = 0; i < N; i += 4)`

		sum += i;

D	for(i = 0; i < foo(n); ++i)
	bar();
		int temp = foo(n);
		for(i = 0; i < temp; ++i)
		bar();

C

10. 阅读下列 C 代码以及它编译生成的汇编语言

```
long func() {
    long ans = 1;
    long i;
    for (i = 0; i < 1000; i += 2)
        ans = ans ?? (A[i] ?? A[i+1]);
    return ans;
}
```

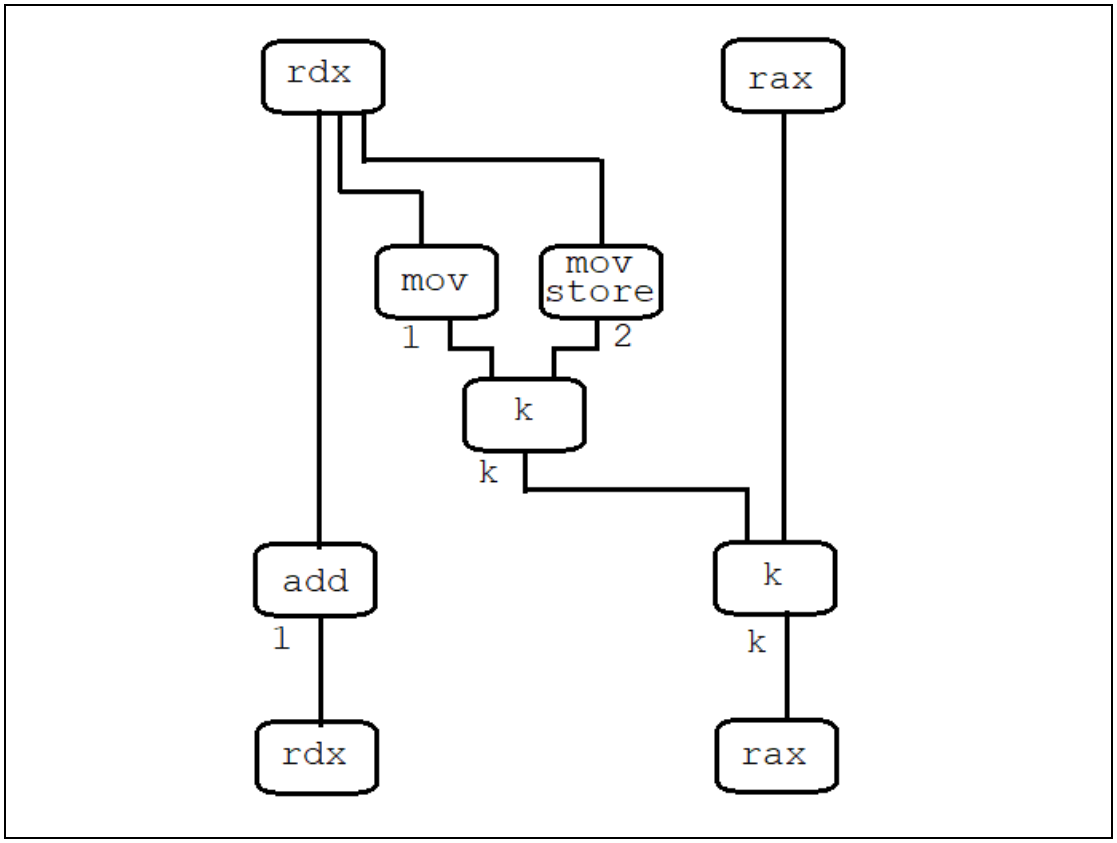
```
func:
    movl $0, %edx
    movl $1, %eax
    leaq A(%rip), %rsi
    jmp .L2
.L3:
    movq 8(%rsi,%rdx,8), %rcx // 2 cycles
    ?? (%rsi,%rdx,8), %rcx    // k + 1 cycles
    ?? %rcx, %rax             // k cycles
    addq $2, %rdx             // 1 cycles
.L2:
    cmpq $999, %rdx           // 1 cycles
    jle .L3
    rep ret
```

该程序每轮循环处理两个元素。在理想的机器上（执行单元足够多），每条指令消耗的时间周期如右边所示。

(1) 当问号处为乘法时，k = 8。此时这段程序的 CPE 为__**4**__

(2) 当问号处为加法时，k = 1。此时这段程序的 CPE 为__**0.5**__

数据相关图：


rdx
rax
mov
1
mov
store
2
k
k
add
1
k
k
rdx
rax

得分

第五题（15 分）

Cache 为处理器提供了一个高性能的存储器层次框架。下面是一个 8 位存储器地址引用的列表（地址单位为字节，地址为 10 进制表示）：

3，180，43，2，191，88，190，14，181，44

1. (7 分)考虑如下 cache（S=2, E=2），每个 cache block 大小为 2 个字节。假设 cache 初始状态为空，替换策略为 LRU。请填补下表：
(Tag 使用二进制格式；Data 使用十进制格式，例：M[6-7]表示地址 6 和 7 对应的数据)

	V	Tag	Data	V	TAG	Data
SET 0	1			1		
SET 1	1			1		

共命中_____次（1 分），分别访问地址_____（地址用 10 进制表示，2 分）

解答：

答案：

	V	Tag	Data	V	TAG	Data
SET 0	1	101101 M[180-181]		1	001011 M[44-45]	
SET 1	1	000011 M[14-15]		1	101111 M[190-191]	

共命中 3 次

（表格上每空格 1 分，tag 和 data 都正确才得分；）

命中次数回答正确得 1 分；

分别为访问地址 2、190、181（三个地址完全正确得 2 分，答对两个得 1 分）

2. (5 分)现在有另外两种直接映射的 cache 设计方案 C1 和 C2，每种方案的 cache 总大小都为 8 个字节，C1 块大小为 2 个字节，C2 块大小为 4 个字节。假设从内存加载一次数据到 cache 的时间为 25 个周期，访问一次 C1 的时间为 3 个周期，访问一次 C2 的时间为 5 个周期。针对第一问的地址访问序列，哪一种 cache 的设计更好？（请分别给出两种 cache 访问第一问地址序列的总时间以及 miss rate）

答案：

Address	Binary address	C1 hit/miss	C2 hit/Miss
3	000000 11	M	M
180	101101 00	M	M
43	001010 11	M	M

2	000000 10	M	M
191	101111 11	M	M
88	010110 00	M	M
190	101111 10	H	H
14	000011 10	M	M
181	101101 01	H	M
44	001011 00	M	M

C1 更好。(1 分)

C1: miss rate = 8/10 = 80%, (1 分) total cycles = 8 * 25 + 10 * 3 = 230 (1 分)

C2: miss rate = 9/10 = 90%, (1 分) total cycles = 9 * 25 + 10 * 5 = 275 (1 分)

3. (3 分)现在考虑另外一个计算机系统。在该系统中，存储器地址为 32 位，并采用如下的 cache：

Cache datasize	Cache block size	Cache mode
32 KiB	8 Bytes	直接映射

此 cache 至少要占用_____Bytes. (datasize + (valid bit size + tag size) * blocks)

答案：

cache block 为 8 bytes，所以 b=3；

cache block 一共 32*1024 / 8 = 4096 个，又因为是直接映射，所以 s=12；于是 tag 位一共 t = 32 - s - t = 17。所以总大小为：

totalsize = datasize + (valid bit size + tag size) * blocks

= 32 * 1024 + (1 + 17) * 4096 / 8 = 41984 (bytes)