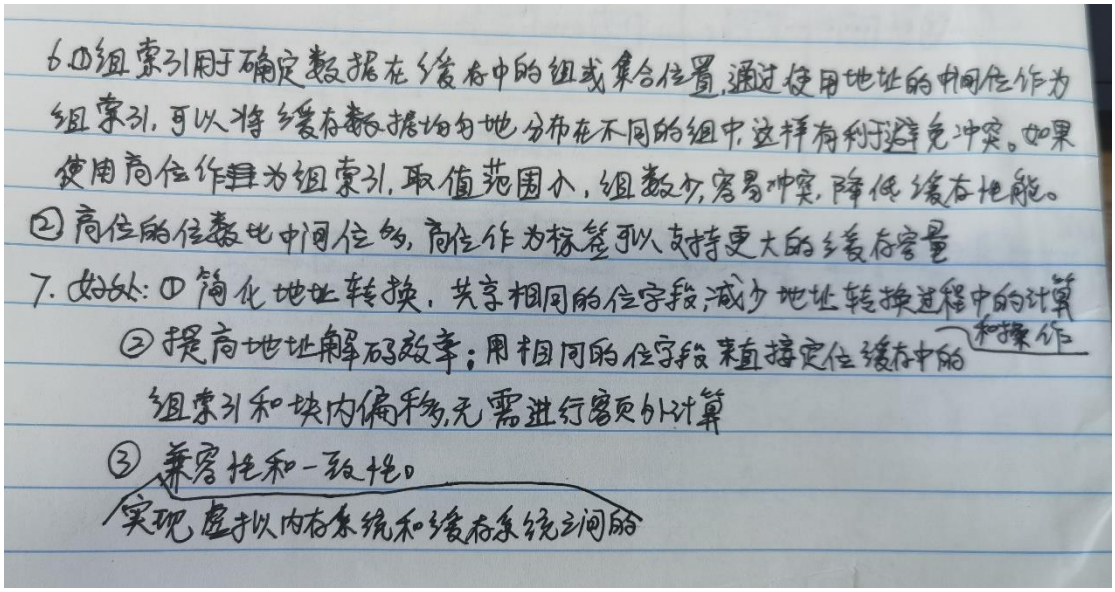

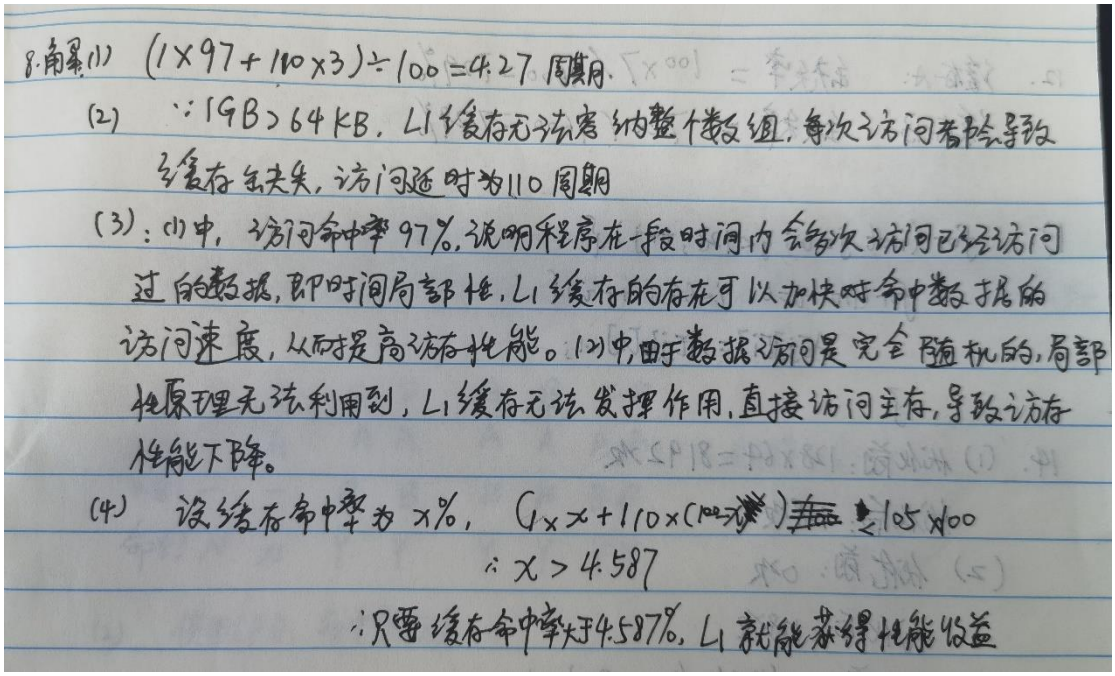

9. 根据给出的不同缓存配置，补全下表中缺失的字段。

编号	地址位数 Bit	缓存大小 KB	块大小 Byte	相联度	组数量	组索引位数 Bit	标签位数 Bit	偏移位数 Bit
1	32	4	64	2	32	5	21	6
2	32	4	64	8	8	3	23	6
3	32	4	64	全相联	1	0	26	6
4	32	16	64	1	256	8	18	6
5	32	16	128	2	64	6	19	7
6	32	64	64	4	256	8	18	6
7	32	64	64	16	64	6	20	6
8	32	64	128	16	32	5	20	7

10. 解：(1) A 时间：$0.22(1-P_1)+P_1\times 100 = 0.22+99.78P_1$

B 时间：$0.52(1-P_2)+P_2\times 100 = 0.52+99.48P_2$

$0.22+99.78P_1 < 0.52+99.48P_2$

$\Rightarrow P_1 < 0.003+0.997P_2$

(2) A时间：$0.22(1-P_1)+0.22kP_1 = 0.22+0.22(k-1)P_1$

B时间：$0.52+0.52(k-1)P_2$

$0.22+0.22(k-1)P_1 < 0.52+0.52(k-1)P_2$

$\Rightarrow P_1 < \frac{1.36}{k-1}+2.36P_2$

11. 直接映射：0次　　4路组相联：4次

2路组相联：2次　　8路组相联：7次

12. 缓存A：缺失率 $= 100\times 7/9600 = 7.29\%$

缓存B：缺失率 $= 700/9600 = 7.29\%$

13.
```
for (int j=0; j<128; ++j) {
    for (int i=0; i<64; ++i) {
        A[j][i] = A[j][i]+1;
    }
}
```

14. (1) 优化前：$128\times 64 = 8192$次

优化后：128次

(2) 优化前：0次

优化后：128次

(3) 优化前：$64\times 4/32 = 8$个块

优化后：$256/32 = 8$个块

15.

	input 数组				output 数组			
	列 0	列 1	列 2	列 3	列 0	列 1	列 2	列 3
行 0	miss	miss	miss	miss	miss	miss	miss	miss
行 1	miss	miss	miss	miss	miss	miss	miss	miss
行 2	miss	miss	miss	miss	miss	miss	miss	miss
行 3	miss	miss	miss	miss	miss	miss	miss	miss

16. (1) 512÷16=32 块，1个块可以存储2个元素，且两个元素属于不同的组

∴命中率为100%

(2) 不会。因为程序访问input是按行访问，每次访问都会读取2个元素，而缓存的块大小已经足够存储两个连续的元素，增加缓存的总大小并不会改变程序的访存模式和访存方式，因此对命中率没有影响。

(3) 可以。增加块大小，使得每个块存储更多元素，从而提高命中率。

17. (1) ∵虚拟地址 0x05A4

∴页号 0x05，偏移量 0xA4

∴组号为1，标签为0x05 命中，物理地址0x0D组0xA4得0x0DA4 ~~0x0DA4~~

(2) $2^{14-6}=2^8=256$ 个条目

(3) 物理地址0x0DA4，对应页号0x0D，没有块偏移，有效位为0，命中缓存

18. (1)

	A	B	C	D	A	B	C	D
way 0	A	A	A	A	A	A	A	A
way 1	-	-	B	B	B	B	B	B
命中?	N	N	Y	Y	Y	Y	Y	Y

命中率 50%

(2) 使用LFU，命中率75%

19. (1) 原因：同一缓存组内的不同块需要通过低位标签来进行区分和匹配

(2) 影响：影响替换策略，影响缓存性能

(3) 8KB/4=2KB，$\log_2(2KB)=\log_2(2048)=11$ 比特

20. 监听一致性，优点：实现简单，通信开销较低

缺点：总线开销较大，无法利用局部性原则

目录一致性：优点：通信开销较低，利用了局部性原则

缺点：实现复杂，延迟增加

缓存一致性的实现代价体现在：软硬件开销，通信开销，一致性维护开销