

全国青少年信息学奥林匹克（NOI2018）

山东省队第二轮选拔赛

第二试

考试时间：2018 年 5 月 16 日 8: 00 – 13: 00

对于 C/C++ 程序，最终测试将使用 C++11 标准并开启 -O2 优化

Problem A. 原题识别 (old.c/cpp/pas)

Input file: old.in
 Output file: old.out
 Time limit: 10 seconds
 Memory limit: 512 megabytes

“人肉题库”小 Q 刷题非常勤奋，题量破万。每当有人拿题目请教他时，小 Q 总能在 1 秒内报出这是哪个 OJ 的哪道题。因此，小 Q 是被当作“原题搜题机”一样的存在。

有一天，小 Q 来到了一棵 n 个节点的有根树下，这棵树的根节点为 1 号点，且每个节点都印着一道题目。凭借超大的题量，小 Q 迅速识别出了每道题的来源，并发现有些题目被搬运了好多次。他把每个节点的题目都做了一个分类，第 i 个节点的题目对应的题目种类为 a_i ，当且仅当 $a_i = a_j$ 时， i 点和 j 点的题目来源是相同的。

同一道题目做多次除了增加 AC 数以外，对本身的水平没有任何提高。为了调查这棵树的题目质量，小 Q 会不断提出以下两种询问共 m 次：

- 1 $x y$: 如果将 x 点到 y 点的最短路径上的所有点 (包括 x 和 y) 对应的题目都做一遍，那么一共可以做到多少道本质不同的题目？
- 2 $A B$: 如果在 A 点到根的最短路径上 (包括 A 点和根) 等概率随机选择一个点 x ，在 B 点到根的最短路径上 (包括 B 点和根) 等概率随机选择一个点 y ，那么询问 1 $x y$ 的答案期望是多少？

定义 cnt_x 表示 x 点到根最短路径上的节点个数，因为小 Q 不喜欢分数，而且第 2 类询问的答案一定可以表示成 $\frac{ans}{cnt_A \times cnt_B}$ 的形式，你只需要告诉他 ans 的值就可以了。

识别这些题目消耗了小 Q 太大的精力，他没有办法自己去计算这些简单的询问的答案。请写一个程序回答小 Q 的所有 m 个问题。

Input

第一行包含一个正整数 T ，表示测试数据的组数。

每组数据第一行包含 5 个正整数 n, p, SA, SB, SC ，其中 n 表示树的节点个数。

为了在某种程度上减少输入量，树边和每个点的题目种类 $a[]$ 将由以下代码生成：

```

1 unsigned int SA, SB, SC;
2 unsigned int rng61(){
3     SA ^= SA << 16;
4     SA ^= SA >> 5;
5     SA ^= SA << 1;
6     unsigned int t = SA;
7     SA = SB;
8     SB = SC;
9     SC ^= t ^ SA;
10    return SC;
11 }
```

```

12 void gen(){
13     scanf("%d%d%u%u%u", &n, &p, &SA, &SB, &SC);
14     for(int i = 2; i <= p; i++)
15         addedge(i - 1, i);
16     for(int i = p + 1; i <= n; i++)
17         addedge(rng61() % (i - 1) + 1, i);
18     for(int i = 1; i <= n; i++)
19         a[i] = rng61() % n + 1;
20 }

```

第二行包含一个正整数 m ，表示询问次数。

接下来 m 行，每行 3 个正整数，形式为 $1\ x\ y$ 或者 $2\ A\ B$ ，依次表示每个询问。

Output

对于每组数据，输出 m 行，每行一个整数，依次回答每个询问。

Examples

old.in	old.out
2	1
5 3 10000 12345 54321	5
3	1
1 2 3	4
2 1 3	34
1 3 2	61
10 6 23456 77777 55555	45
5	3
1 1 10	
2 3 5	
2 7 5	
2 5 4	
1 8 6	

Notes

- $1 \leq T \leq 3$, $2 \leq p \leq n \leq 100000$, $1 \leq m \leq 200000$,
- $10000 \leq SA, SB, SC \leq 1000000$, $1 \leq x, y, A, B \leq n$ 。

子任务 1 (30 分): 只含第 1 类询问。

子任务 2 (30 分): 满足 $p = n$ 。

子任务 3 (40 分): 没有任何附加的限制。

Problem B. 旧试题 (divsum.c/cpp/pas)

Input file: divsum.in
 Output file: divsum.out
 Time limit: 5 seconds
 Memory limit: 512 megabytes

时光匆匆，转眼间又是一年寒暑……

这是小 Q 同学第二次参加省队选拔赛。今年，小 Q 痛定思痛，不再冒险偷取试题，而是通过练习旧试题提升个人实力。可是旧试题太多了，小 Q 没日没夜地做题，却看不到前方的光明在哪里。

一天，因做题过度而疲惫入睡的小 Q 梦到自己在考场上遇到了一道好像做过的题目，却怎么也想不起曾经自己是怎么解决它的，直到醒来还心有余悸。

小 Q 眉头一皱，感觉事情不妙，于是他找到了你，希望你能教他解决这道题目。小 Q 依稀记得题目要计算如下表达式的值

$$\left(\sum_{i=1}^A \sum_{j=1}^B \sum_{k=1}^C d(ijk) \right) \bmod (10^9 + 7)$$

其中 $d(ijk)$ 表示 ijk 的约数个数。

Input

第一行包含一个正整数 T ，表示有 T 组测试数据。

接下来 T 行，每行描述一组测试数据，包含三个整数 A, B 和 C ，含义见题目描述。

Output

对于每组测试数据，输出一行，包含一个整数，表示所求表达式的值。

Examples

divsum.in	divsum.out
5	11536
10 10 10	51103588
100 100 100	165949340
1000 1000 1000	19234764
10000 10000 10000	176764584
100000 100000 100000	

Notes

对于 30% 的数据， $1 \leq A, B, C \leq 5000$ 。

对于 100% 的数据， $1 \leq T \leq 10, 1 \leq A, B, C \leq 10^5, 1 \leq \sum \max(A, B, C) \leq 2 \cdot 10^5$ 。

Problem C. 荣誉称号 (title.c/cpp/pas)

Input file: title.in
 Output file: title.out
 Time limit: 10 seconds
 Memory limit: 512 megabytes

休闲游戏玩家小 Q 不仅在算法竞赛方面取得了优异的成绩，还在一款收集钻石的游戏中排名很高。

这款游戏一共有 n 种不同类别的钻石，编号依次为 1 到 n 。小 Q 已经玩了这款游戏很久了，对于第 i 种钻石，他已经收集到了 a_i 个。这款游戏最大的亮点就是，钻石只有一种获得途径，那就是从商城中购买。具体来说，第 i 种钻石的单价为 b_i 点券。为了鼓励玩家充值，每种钻石都没有数量上限，只要肯充钱，就可以拥有任意多的钻石。但是这款游戏并没有开发“丢弃道具”功能，因此小 Q 不能通过丢弃钻石去完成任务。

最近这款游戏推出了一个限时成就任务，完成任务的玩家可以获得荣誉称号，而完成任务条件则是：给定正整数 k 和 m ，对于任意一个整数 $x(2^k \leq x \leq n)$ ， $a_x + a_{\lfloor \frac{x}{2} \rfloor} + a_{\lfloor \frac{x}{4} \rfloor} + a_{\lfloor \frac{x}{8} \rfloor} + \dots + a_{\lfloor \frac{x}{2^k} \rfloor}$ 都要是 m 的倍数。

高玩小 Q 当然想完成这个限时成就任务，但是在充钱之前他想知道他究竟需要多少点券才能完成这个任务。请写一个程序帮助小 Q 计算最少需要的点券数量。

Input

第一行包含一个正整数 T ，表示测试数据的组数。

每组数据第一行包含 9 个正整数 $n, k, m, p, SA, SB, SC, A, B$ ，其中 n 表示钻石种类数， k, m 表示任务条件。

为了在某种程度上减少输入量， $a[]$ 和 $b[]$ 由以下代码生成：

```

1 unsigned int SA, SB, SC; int p, A, B;
2 unsigned int rng61(){
3     SA ^= SA << 16;
4     SA ^= SA >> 5;
5     SA ^= SA << 1;
6     unsigned int t = SA;
7     SA = SB;
8     SB = SC;
9     SC ^= t ^ SA;
10    return SC;
11 }
12 void gen(){
13     scanf("%d%d%d%d%u%u%u%d", &n, &k, &m, &p, &SA, &SB, &SC, &A, &B);
14     for(int i = 1; i <= p; i++) scanf("%d", &a[i], &b[i]);
15     for(int i = p + 1; i <= n; i++){
16         a[i] = rng61() % A + 1;
17         b[i] = rng61() % B + 1;
18     }
19 }

```

Output

对于每组数据，输出一行一个整数，即最少需要的点券数量。

Examples

title.in	title.out
2	3
3 1 2 3 11111 22222 33333 1 1	14
1 5	
2 3	
3 6	
7 2 3 7 11111 22222 33333 1 1	
6 9	
4 5	
3 7	
5 2	
2 4	
1 7	
9 6	

Notes

- $1 \leq T \leq 10$,
- $1 \leq k \leq 10$ 且 $2^k \leq n$,
- $1 \leq p \leq \min(n, 100000)$, $10000 \leq SA, SB, SC \leq 1000000$,
- $1 \leq A, B, a_i, b_i \leq 10^7$ 。

子任务 1 (30 分): 满足 $1 \leq n \leq 1000$ 且 $m = 2$ 。

子任务 2 (40 分): 满足 $1 \leq n \leq 10^5$ 且 $m \leq 200$ 。

子任务 3 (30 分): 满足 $1 \leq n \leq 10^7$ 且 $m \leq 200$ 。