

Problem A. 树异或价值

Input file: standard input
Output file: standard output
Time limit: 3 seconds
Memory limit: 512 megabytes

曾经有一棵 n 个节点的有根树，其根节点的编号为 1。同时，小 ω 有一个大小为 n 的数组 a_1, a_2, \dots, a_n 。

定义数组 a 的价值为：

$$\sum_{i=1}^n \sum_{j=1}^n (a_i \oplus a_j) \times dep_{LCA(i,j)}$$

相关定义解释：

- \oplus 表示按位异或。
- dep_x 表示 x 的深度，即 x 到根节点的路径上节点的数量。
- $LCA(i, j)$ 表示节点 i 和 j 的最近公共祖先。

不幸的是，很多年之后小 ω 忘记了数组 a 。但是他记得：

- $\forall 1 \leq i \leq n, 0 \leq a_i \leq 2^k - 1$ ，其中 k 是一个给定的常数。
- 在所有 2^{kn} 种可能的 a 数组中，小 ω 的 a 数组的价值是最大的。

小 ω 并不满足于找到一个合法的 a 数组，他更想知道有多少种 a 数组能满足上述条件。答案可能很大，请输出其对 998244353 取模的结果。

Input

输入包含多组测试数据。

第一行包含一个整数 T ($1 \leq T \leq 20$)，表示测试数据的组数。

对于每组测试数据：

第一行包含两个整数 n, k ($1 \leq n \leq 2 \times 10^5, 1 \leq k \leq 10^9$)，表示树与 a 数组的大小， a 数组元素的上界。

第二行包含 $n - 1$ 个整数 p_2, p_3, \dots, p_n ($1 \leq p_i < i$)，其中 p_i 表示树上节点 i 的父亲。

Output

对于每组测试数据：

输出一行一个整数，表示合法 a 数组数量对 998244353 取模的结果。

Example

standard input	standard output
1 3 3 1 1	216

Problem B. 树上询问

Input file: **standard input**
Output file: **standard output**
Time limit: 10 seconds
Memory limit: 256 megabytes

给定一棵大小为 n 的树，节点的点权 $p: p_1, p_2, \dots, p_n$ 是一个排列。

你需要处理共 m 次两种类型的操作：

- 给定节点 x 和 y ，交换 p_x 和 p_y 。
- 给定区间 $[l, r]$ ，询问是否存在两个节点 x 和 y ，使得从 x 到 y 路径上的点权恰好为 $[l, r]$ 区间内的所有整数。

Input

输入包含多组测试数据。

第一行包含一个整数 T ($1 \leq T \leq 20$)，表示测试数据的组数。

对于每组测试数据：

第一行包含一个整数 n ($1 \leq n \leq 10^5$)，表示树的大小。

第二行包含 n 个整数 p_1, p_2, \dots, p_n ($1 \leq p_i \leq n$)，表示节点的点权。保证 p 是一个排列。

接下来 $n - 1$ 行，每行两个整数 x_i, y_i ($1 \leq x_i, y_i \leq n$)，表示存在一条连接节点 x_i 与节点 y_i 的边。保证所有边构成一棵树。

接下来一行输入一个整数 m ($1 \leq m \leq 10^5$)，表示操作的数量。

接下来 m 行表示操作，第 i 次操作为两种操作中的一种：

- “1 x_i y_i ”，表示交换节点 x_i 和 y_i 的点权。注意， x_i 和 y_i 可能相同。
- “2 l_i r_i ”，表示询问的区间。

Output

对于每次操作 2，输出一行一个字符串 “Yes” 或 “No”（不包含引号），分别表示树上存在/不存在满足条件的节点 x 与 y 。

Example

standard input	standard output
2	Yes
5	No
1 2 3 4 5	Yes
1 2	
1 3	
3 4	
3 5	
3	
2 1 3	
1 3 4	
2 1 3	
4	
1 2 3 4	
1 2	
2 3	
3 4	
1	
2 1 4	

Problem C. 黑洞合并

Input file: **standard input**
Output file: **standard output**
Time limit: 7.5 seconds
Memory limit: 512 megabytes

宇宙中初始有 n 个黑洞，从左到右编号为 1 到 n ，初始质量依次为 w_1, w_2, \dots, w_n 。
黑洞间即将发生 $n - 1$ 次合并，每次将两个黑洞合并为一个。合并遵循的规律如下：

- 第 i 次合并开始前，剩余的黑洞数量为 $n - i + 1$ ，从左到右**重新编号**为 $1, 2, \dots, n - i + 1$ ；
- 第 i 次合并时，**随机**选取两个编号为 x_i 和 y_i ，满足 $x_i + y_i = n - i + 2$ 的黑洞进行合并，合并后的黑洞**随机**占据原先黑洞 x_i 或 y_i 的位置，其质量为 $w_{x_i} + w_{y_i}$ ；
- 第 i 次合并会释放出 $w_{x_i} \cdot w_{y_i} \cdot (w_{x_i} + w_{y_i})$ 的能量。

$n - 1$ 次合并后，只剩下一个黑洞，请你计算 $n - 1$ 次合并中释放能量之和的期望。答案可能很大，请输出答案对 998244353 取模后的结果。

Input

输入包含多组测试数据：
输入的第一行包含一个整数 T ($1 \leq T \leq 10$)，表示测试数据的组数。
对于每组测试数据：
第一行包含一个整数 n ($1 \leq n \leq 10^6$)，表示初始黑洞的数量。
第二行包含 n 个整数 w_1, w_2, \dots, w_n ($1 \leq w_i \leq 10^6$)，表示黑洞的初始质量。

Output

对于每组测试数据：
一行包含一个整数，表示答案对 998244353 取模后的结果。

Example

standard input	standard output
1 3 1 1 1	8

Problem D. 亡语

Input file: **standard input**
Output file: **standard output**
Time limit: 3 seconds
Memory limit: 512 megabytes

你在玩一款在线卡牌游戏《炉石传说》。

你有 n 个随从，随从死亡当且仅当其生命值小于等于 0。一个随从死亡后，将会触发 亡语。随从分为三类，每类随从具有不同的 存活效果 与 亡语效果：

- 存活效果：随从存活时产生的效果，死亡但未结算亡语的随从**不具有**存活效果
- 亡语效果：一个随从**首次**生命值降到小于等于 0 时触发的效果

类型	初始信息	存活效果	死亡效果
1	h - 初始生命值	无	对所有随从造成 1 点伤害
2	h - 初始生命值 x - 初始等级 k - 亡语召唤数量	无	召唤 k 个 [编号继承此随从 & 初始生命值为该随从初始生命值 2 倍 & 初始等级为 $x - 1$ & 亡语召唤数量为 k] 的 第二类 随从 (也具有第二类随从的亡语效果)。 特别地 ，等级 0 的随从死亡时 不会召唤新的随从。
3	h - 初始生命值	所有随从受到的伤害增加一倍。 假设当前有 m 个第三类随从存活， 那么所有随从 (包括自己) 受到 的伤害会变为 $m + 1$ 倍。	无

假设触发亡语的顺序依次为 亡语 1，亡语 2，那么因为亡语 1 或 2 死亡的随从，其亡语触发顺序遵循如下规则：

- 因亡语 1 死亡的随从，其亡语结算在 因亡语 2 死亡的随从 之前；（可以理解为 BFS）
- 因同一个亡语死亡的随从，优先结算编号较小的随从的亡语。

给定 n 个随从的初始信息，请你求出当第一个随从死亡（强制死亡）并依次结算所有亡语后，所有随从的状态，具体见输出格式。

Input

输入包含多组测试数据：

输入的第一行包含一个整数 T ($1 \leq T \leq 10$)，表示测试数据的组数。

对于每组测试数据：

第一行包含一个整数 n ($1 \leq n \leq 10^5$)，表示随从个数。

接下来 n 行，描述 n 个随从，第 i 个随从为三种类型中的一种：

- “1 h_i ” ($1 \leq h_i \leq 10^9$)，表示第 i 个随从属于第一类，初始生命值为 h_i
- “2 h_i x_i k_i ” ($1 \leq h_i \leq 10^9, 0 \leq x_i \leq 10^5, 1 \leq k_i \leq 10$)，表示第 i 个随从属于第二类，初始生命值为 h_i ，等级为 x_i ，亡语召唤数量为 k_i
- “3 h_i ” ($1 \leq h_i \leq 10^9$)，表示第 i 个随从属于第三类，初始生命值为 h_i

Output

对于每组测试数据:

输出 n 行, 第 i 行表示第 i 个随从的状态:

若为第一类随从, 输出一个整数 h_i 表示它的剩余生命值 ($h_i = 0$ 表示其死亡) 。

若为第二类随从, 输出三个整数 m_i, x_i, h_i :

- 若第 i 个随从未死亡, 则 h_i 等于其剩余生命值, $m_i = 1$, x_i 等于其初始等级;
- 若第 i 个随从已死亡, 则 m_i, x_i, h_i 表示其衍生随从的状态: m_i 个等级为 x_i , 剩余生命值为 h_i 的随从;
- 特殊地, 若第 i 个随从及其衍生随从均死亡, $m_i = x_i = h_i = 0$ 。

若为第三类随从, 输出一个整数 h_i 表示它的剩余生命值 ($h_i = 0$ 表示其死亡) 。

Example

standard input	standard output
3	0
3	2 1 2
1 1	5
2 1 2 2	0
3 7	9 0 4
5	0
1 1	3 1 2
2 1 2 3	3
1 1	0
2 1 2 3	3 1 1
3 7	0
5	3 1 2
1 1	0
2 1 2 3	
1 1	
2 1 2 3	
3 1	

Note

对于第二组样例, 完整过程如下:

1. 随从 1 死亡, 由于随从 5 未死亡, 所以对所有随从造成 2 点伤害, 随从 2,3,4 死亡;
2. 随从 2 亡语结算, 召唤三个等级为 1, 血量为 2 的随从;
3. 随从 3 亡语结算, 由于随从 5 未死亡, 所以对所有随从造成 2 点伤害, 新随从 2 死亡;
4. 随从 4 亡语结算, 召唤三个等级为 1, 血量为 2 的随从;
5. 新随从 2 亡语结算, 召唤九个等级为 0, 血量为 4 的随从。

对于第三组样例, 完整过程如下:

1. 随从 1 死亡, 由于随从 5 未死亡, 所以对所有随从造成 2 点伤害, 随从 2,3,4 死亡;
2. 随从 2 亡语结算, 召唤三个等级为 1, 血量为 2 的随从;

3. 随从 3 亡语结算，由于随从 5 死亡，所以对所有随从造成 1 点伤害；
4. 随从 4 亡语结算，召唤三个等级为 1，血量为 2 的随从。

Problem E. 怪物猎人

Input file: standard input
Output file: standard output
Time limit: 1 second
Memory limit: 256 megabytes

你有两只宠物，从第一轮开始，它们将轮流攻击一个生命值为 k 的怪物。

- 在奇数轮次中，第一只宠物**随机**对怪物造成 x 或 y 的伤害；
- 在偶数轮次中，第二只宠物**随机**对怪物造成 x 或 y 的伤害。

请你回答，对于第 i ($i \in \{1, 2\}$) 只宠物，其是否能给予怪物最后一击（首次将其生命值减少到小于等于 0）。

Input

输入包含多组测试数据。

第一行包含一个整数 T ($1 \leq T \leq 5 \times 10^5$)，表示测试数据的组数。

对于每组测试数据：

一行包含三个整数 k, x, y ($1 \leq k, x, y \leq 10^{18}$)，表示怪物的生命值和两种攻击的伤害。

Output

对于每组测试数据：

输出两行，

第 i 行包含一个字符串 "Yes"或 "No"（不包含引号），表示对于第 i 只宠物的答案。

Examples

standard input	standard output
1	Yes
2 3 4	No
2	Yes
9 4 2	Yes
7 3 4	Yes
	Yes

Problem F. 融合矿石

Input file: standard input
Output file: standard output
Time limit: 1 second
Memory limit: 256 megabytes

在遥远的古代，有一个被遗忘的王国，其地下埋藏着无尽的宝藏。这些宝藏并非普通的金银财宝，而是由一种特殊合金制成，合金中蕴含着不同比例的"金辉石"。

王国中散落着 n 种不同的矿石，每种矿石的数量都是无限的。对于第 i 种矿石，一块该矿石的质量为 a_i ，其中含有"金辉石"的质量为 b_i ($0 < b_i \leq a_i$)。探险家们发现，矿石之间可以相互融合：将两块矿石融合后，会得到一块新的矿石，其质量为两者质量之和，含有"金辉石"的质量也是两者之和。融合后的矿石仍可与其他矿石融合。

为了评估这些矿石的价值，探险家们制定了一套价值体系：对于任意一块矿石，其单位质量的价值由其中"金辉石"的占比 $\frac{b}{a}$ 决定。这个价值体系由 10 个整数 $v_1, v_2, v_3, \dots, v_{10}$ 组成，表示对于"金辉石"占比 $\in (\frac{i-1}{10}, \frac{i}{10}]$ 的矿石，其单位质量的价值是 v_i 。保证随着"金辉石"占比的增大，矿石的单位价值是单调不减的，即 $\forall i \in [1, 10), v_i \leq v_{i+1}$ 。

现在，探险家们携带了一个最大承重为 m 的背包，在进行任意次融合矿石的操作后，选择总质量不超过 m 的矿石装入背包中，请你计算背包内矿石的最大总价值。

Input

输入包含多组测试数据。

输入的第一行包含一个整数 T ($1 \leq T \leq 10$)，表示数据组数。

对于每组测试数据：

第一行包含 10 个整数 $v_1, v_2, v_3, \dots, v_{10}$ ($1 \leq v_i \leq 10^5, \forall i \in [1, 10), v_i \leq v_{i+1}$)，表示价值体系。

第二行包含两个整数 n, m ($1 \leq n, m \leq 3000$)，表示矿石的种类数和背包的最大承重。

接下来 n 行，每行两个整数 a_i, b_i ($0 < b_i \leq a_i \leq 3000$)，表示一块第 i 种矿石的质量和其中"金辉石"的质量。

保证所有测试数据中 n 的总和与 m 的总和都不超过 10^4 。

Output

对于每组测试数据：

输出一行一个整数，表示最大总价值。

Example

standard input	standard output
1 1 1 1 1 1 3 3 3 4 5 2 10 4 2 6 5	30

Problem G. 小猫钓鱼

Input file: standard input
Output file: standard output
Time limit: 1 second
Memory limit: 256 megabytes

shuishui 和 sha7dow 在玩 小猫钓鱼。
游戏规则如下：

- 游戏开始时，shuishui 和 sha7dow 各有 n 张牌，每张牌上标有 $[1, n]$ 中的一个整数，且 $[1, n]$ 中每个整数各出现 2 次；
- shuishui 初始牌为 $a : a_1, a_2, \dots, a_n$ ，sha7dow 初始牌为 $b : b_1, b_2, \dots, b_n$ ；
- 游戏开始后，两人轮流操作，将手中的任意一张牌置于牌堆顶部（牌堆初始为空）。假设放置的牌的点数为 x ，若放置前牌堆中已存在点数为 x 的牌，则放置该牌的玩家 **从牌堆顶往下**，直到牌堆中原有的点数为 x 的牌，将这些牌全部拿走，放入己方手牌中（包括 放置的点数为 x 的牌 和原有的点数为 x 的牌），并获得等同于拿走牌数量的分数。
- 轮到一个玩家放置手牌时，若此时其手牌为空，则游戏结束。
- 游戏结束后，分数较高的玩家获胜，若分数相同或者游戏不会结束，则游戏没有获胜者。

给定 shuishui 和 sha7dow 的初始手牌，shuishui 先手操作。假设 shuishui 和 sha7dow 足够聪明，且都以获胜为目的，请你求出游戏的胜利者，或者报告游戏没有获胜者。

Input

输入包含多组测试数据。
第一行包含一个整数 T ($1 \leq T \leq 5000$)，表示测试数据的组数。
对于每组测试数据：
第一行包含一个整数 n ($1 \leq n \leq 2 \times 10^5$)，表示初始双方的手牌数；
第二行包含 n 个正整数 a_1, a_2, \dots, a_n ($1 \leq a_i \leq n$)，表示 shuishui 的初始手牌；
第三行包含 n 个正整数 b_1, b_2, \dots, b_n ($1 \leq b_i \leq n$)，表示 sha7dow 的初始手牌；
输入保证在 a 和 b 中， $[1, n]$ 中每个数各出现 2 次。保证所有的测试数据中 n 的总和不超过 5×10^5 。

Output

对于每组测试数据：
输出一行一个字符串：
若 shuishui 获胜，输出 “shuishui”；
若 sha7dow 获胜，输出 “sha7dow”；
若游戏没有获胜者，输出 “Tie”。（所有字符串都不包含引号）

Example

standard input	standard output
1 3 1 1 2 3 3 2	shuishui

Problem H. 最佳选手

Input file: standard input
Output file: standard output
Time limit: 5 seconds
Memory limit: 256 megabytes

第17届 Culinary Combat Professional Contest (CCPC) 已经结束，赛事组织者将选出本次比赛的最佳选手。

共有 n 名选手，编号从 1 到 n ，一共进行了 m 场 1 vs 1 的对决：

- 第 i 场对决的是选手 a_i 和 b_i ；
- 每场对决分为上半场和下半场：
 - 在对决的上半场，选手 a_i 的得分为 x_i ，选手 b_i 的得分为 y_i ；
 - 在对决的下半场，选手 a_i 和 b_i 的准确得分未知，但两者得分之和为 z_i ；
 - 选手在整场对决中的得分等于上半场得分与下半场得分之和。换句话说，在第 i 场对决中， a_i 和 b_i 的可能得分分别为 $x_i + p_i$ 和 $y_i + q_i$ ，其中 $0 \leq p_i, q_i \leq z_i, p_i + q_i = z_i$ 。

注意，所有得分均为非负整数，并且每位选手至少参加了一场对决。

定义一位选手的关键对决为：这位选手参加的对决中，得分 **最小** 的对决；一位选手的最终得分为：其在关键对决中获得的得分。如果一名选手的最终得分 **严格大于** 所有其他选手的最终得分，则该选手将获得最佳选手奖。

由于 m 场对决下半场得分的不确定，最佳选手也可能不同。请找出所有可能成为最佳选手的选手。

Input

输入包含多组测试数据。

第一行包含一个整数 T ($1 \leq T \leq 2 \times 10^5$)，表示测试数据的组数。

对于每组测试数据：

第一行包含两个整数 n, m ($2 \leq n \leq 10^6, \lceil \frac{n}{2} \rceil \leq m \leq 10^6$)，表示选手数和对决数。

接下来的 m 行，第 i 行包含五个整数 a_i, b_i, x_i, y_i, z_i ($1 \leq a_i, b_i \leq n, a_i \neq b_i, 0 \leq x_i, y_i, z_i \leq 10^9$)，具体含义见题面。保证每位选手至少参加了一场对决。

保证所有测试数据中 n 的总和与 m 的总和都不超过 10^6 。

Output

对于每组测试数据：

第一行输出一个整数 k ，表示可能成为最佳选手的选手数量。

第二行输出 k 个整数，按 **升序** 输出所有可能成为最佳选手的选手编号。**特别地**，当 $k = 0$ 时，输出一个空行。

Example

standard input	standard output
3	1
3 2	3
1 2 2 3 3	1
2 3 6 7 1	3
3 2	3
1 2 2 3 3	1 2 3
2 3 6 7 2	
3 2	
1 2 2 3 6	
2 3 7 7 2	

Note

对于第三组测试数据，3 名选手都有可能成为最佳选手。选手 1 只有在如下情况中才能成为最佳选手：

- 在第一场对决中，选手 1 得分为 $2 + 6 = 8$ ，选手 2 得分为 $3 + 0 = 3$ ；
- 在第二场对决中，选手 2 得分为 $7 + 2 = 9$ ，选手 3 得分为 $7 + 0 = 7$ 。

在这种情况下，选手 1 的最终得分为 8，选手 2 的最终得分为 $\min(3, 9) = 3$ ，选手 3 的最终得分为 7，因此选手 1 为最佳选手。

Problem I. 长期素食

Input file: **standard input**
Output file: **standard output**
Time limit: 2 seconds
Memory limit: 256 megabytes

Mutsumi 是一名素食主义者。为了保持长期素食，她准备在接下来的 k 天中只吃自己种的黄瓜。

Mutsumi 有 n 片黄瓜田，编号从 1 到 n 。在接下来 k 天中的每一天，她都会选择一片黄瓜田，收获其中的黄瓜。如果她在第 i 天选择了编号为 j 的黄瓜田，她会获得 $a_j + i \times b_j$ 点幸福值。为了保证黄瓜的正常生长，她不会连续两天选择同一片黄瓜田。

Mutsumi 想知道她在 k 天中获得的幸福值之和的最大值。

Input

输入包含多组测试数据：

输入的第一行包含一个整数 T ($1 \leq T \leq 20$)，表示测试数据的组数。

对于每组测试数据：

第一行包含两个整数 n, k ($2 \leq n, k \leq 10^5$)，表示 Mutsumi 拥有的黄瓜田数量和总天数。

接下来 n 行，第 i 行包含两个整数 a_i, b_i ($1 \leq a_i \leq 10^9, 1 \leq b_i \leq 10^4$)，具体含义见题面。

Output

对于每组测试数据：

输出一行一个整数，表示 Mutsumi 在 k 天中获得的幸福值之和的最大值。

Example

standard input	standard output
1 4 6 10 1 7 2 2 3 9 1	86

Problem J. 收集签名

Input file: standard input
Output file: standard output
Time limit: 1 second
Memory limit: 256 megabytes

在荒野乱斗的世界中共有 n 座城市，由 $n - 1$ 条长短不一的双向道路连通。
收集狂科莱特在 1 号城市中的礼物店内工作。为了收集她最爱的英雄们的签名，她希望开始一段从 1 号城市开始，经过每个城市至少一次，并最后回到 1 号城市的旅程。她的移动速度为 1 米每秒。
除了缓慢地行走外，她也可以使用她的超级技能来收集签名。具体来说，假设她现在处于 x 号城市，她使用超级技能的过程如下：

1. 选择一个终止城市 y 。
2. 从城市 x 冲刺到城市 y ，收集沿途所有城市中英雄的签名，再从城市 y 冲刺回城市 x 。
3. 使用超级技能经过每条道路的所需的时间为 k 秒，无视道路的长度。

她只能在一些特殊的城市中使用超级技能，且每个特殊城市至多发动一次。你能告诉她整个旅程所需的最小时间吗？

Input

输入的第一行包含一个整数 T ($1 \leq T \leq 4 \times 10^3$) —— 表示该组数据包含的测试点组数。
对于每组测试点：
输入的第一行包含 n, k ($1 \leq n \leq 4 \times 10^3, 1 \leq k \leq 10^9$)，表示城市的数量以及使用超级技能经过一条道路所需的时间。
输入的第二行包含 n 个整数 b_1, b_2, \dots, b_n ($b_i \in \{0, 1\}$)，表示科莱特能在某个城市发动超级技能的次数。
接下来 $n - 1$ 行每行包含三个整数 x_i, y_i, w_i ($1 \leq x_i, y_i \leq n, 1 \leq w_i \leq 10^9$)，表示第 i 条道路的长度为 w_i 米，连接的城市为 x_i 号城市与 y_i 号城市。
保证所有测试点的 $\sum n$ 之和小于等于 2×10^4 。

Output

对于每组测试点，输出一行一个整数，表示科莱特旅途所需的最短时间。

Example

standard input	standard output
2	6
4 1	14
0 0 0 0	
1 2 1	
1 3 1	
3 4 1	
5 2	
0 0 1 0 0	
1 2 6	
1 3 1	
3 4 1	
3 5 1	

Problem K. 地牢谜题：更高还是更低

Input file: standard input
Output file: standard output
Time limit: 12 seconds
Memory limit: 1024 megabytes

在地牢谜题游戏 更高还是更低 中，初始有 n 只血量不同烈焰人排列成一列，第 i 只烈焰人是所有烈焰人中血量第 p_i 小的。你需要按血量从低到高或从高到低的顺序（由谜题规则确定）击杀所有的烈焰人。

你有一把魔法弓，每次射击可以击杀不超过 k 只相邻的烈焰人，你可以自由决定在同一次射击中烈焰人死亡的顺序。两只烈焰人是相邻的当且仅当它们之间的烈焰人都已经死亡。

你和你的同伴需要完成该地牢谜题 m 次。作为一个队伍，你自然不需要独自解决谜题。在第 i 次进行谜题游戏时，你的任务是击杀血量第 l_i 小，第 $l_i + 1$ 小，...，第 r_i 小的烈焰人。这意味着，在你开始任务之前，若谜题规则是按照血量从低到高击杀烈焰人，则血量第 $1, 2, \dots, l_i - 1$ 小的烈焰人已经全部被队友击杀；若谜题规则是按照血量从高到低击杀烈焰人，则血量第 $r_i + 1, r_i + 2, \dots, n$ 小的烈焰人已经全部被队友击杀。

同时，你可以超额完成任务。例如，若谜题规则是按血量从低到高击杀烈焰人，且你的任务是击杀血量第 3 小和第 4 小的烈焰人，那么，击杀血量第 3, 4 小或击杀血量第 3, 4, 5 小的烈焰人都被视为完成任务，而击杀血量第 3, 5 和击杀血量第 3, 4, 6 的烈焰人被视作任务失败。

你惊奇地发现，每次谜题游戏中，烈焰人的血量关系 p 是固定的，区别在于谜题规则和需要击杀的烈焰人范围不同。请你对于每次地牢谜题，计算完成任务所需的最小射击数。

Input

输入包含多组测试数据。

第一行包含一个整数 T ($1 \leq T \leq 10$)，表示测试数据的组数。

对于每组测试数据：

第一行包含两个整数 n, k ($1 \leq k \leq n \leq 2 \times 10^5$)，表示烈焰人的数量和魔法弓每次击杀烈焰人的最大数量。

第二行包含 n 个整数 p_1, p_2, \dots, p_n ($1 \leq p_i \leq n$)，表示烈焰人血量大小关系。保证 p 是一个排列。

第三行包含一个整数 m ($2 \leq m \leq 2 \times 10^5$)，表示需要完成谜题的次数。

接下来 m 行，第 i 行包含两个整数与一个字符 l_i, r_i, c_i ($1 \leq l_i \leq r_i \leq n, c_i \in \{H, L\}$)，表示在第 i 次谜题游戏中，你需要击杀的烈焰人范围，以及谜题规则。其中， $c_i = H$ 表示你需要按血量从低到高击杀烈焰人； $c_i = L$ 表示你需要按血量从高到低击杀烈焰人。

Output

对于每组测试数据：

输出 m 行，第 i 行包含一个整数，表示在第 i 次谜题游戏中，完成任务所需的最小射击数。

Example

standard input	standard output
2	2
5 3	1
1 4 5 3 2	2
2	1
1 3 H	
3 4 L	
5 4	
1 2 5 3 4	
2	
1 4 H	
1 4 L	

Problem L. 地牢谜题：三个怪人

Input file: standard input
Output file: standard output
Time limit: 10 seconds
Memory limit: 512 megabytes

好的，现在你身处一个房间内。在你的面前有 3 个宝箱以及 3 个怪人。其中只有一个宝箱是真的宝箱，其他都是假的，你必须一次打开真宝箱。房间里的每个怪人会告诉你以下两种消息之一，但不一定是实话：

- 有 k ($1 \leq k \leq 3$) 个怪人会说实话。
- 真宝箱是第 i ($1 \leq i \leq 3$) 个宝箱。

这似乎是地牢谜题中最困难的问题之一，因为玩家总是要花时间去考虑究竟谁在说实话。但更为困难的是，作为谜题的设计者，如何设计出合法的谜题。

现在假设有 n 个不同的宝箱以及 m 个相同的怪人，询问有多少种本质不同的合法谜题。一个谜题合法，当且仅当只有一个真宝箱作为前提条件时，玩家能够通过怪人们的话唯一确定真实的宝箱。两个谜题本质不同，当且仅当满足下面两个条件之一：

- 存在 $1 \leq k \leq m$ ，说"有 k 个怪人会说实话"的怪人数量不同。
- 存在 $1 \leq i \leq n$ ，说"真宝箱是第 i 个宝箱"的怪人数量不同。

由于答案可能很大，请输出其对某质数 P 取模的结果。

Input

输入第一行包含一个整数 T ($1 \leq T \leq 10$)，表示该组数据的测试点组数。

对于每组测试点，

输入第一行包含三个整数 n, m, P ($1 \leq n, m \leq 200, 10^8 \leq P < 10^9$)，表示宝箱的数量，怪人的数量，以及取模的质数。

Output

对于每组测试点，输出一行一个整数，表示合法谜题的数量对 P 取模的结果。

Example

standard input	standard output
3	6
2 3 998244353	200
200 1 998244353	0
2 1 998244353	

Note

对于第一组测试点，合法的谜题有：

- 1 个人说第一个宝箱是真的，2 个人说有一个人说实话。这样能够确定第二个宝箱是真的。
- 2 个人说第一个宝箱是真的，1 个人说有两个人说实话。这样能够确定第二个宝箱是真的。
- 1 个人说第一个宝箱是真的，1 个人说有一个人说实话，1 个人说有三个人说实话。这样能够确定第二个宝箱是真的。

- 1个人说第二个宝箱是真的，2个人说有一个人说实话。这样能够确定第一个宝箱是真的。
- 2个人说第二个宝箱是真的，1个人说有两个人说实话。这样能够确定第一个宝箱是真的。
- 1个人说第二个宝箱是真的，1个人说有一个人说实话，1个人说有三个人说实话。这样能够确定第一个宝箱是真的。

对于第二组测试点，唯一的人一旦指定了真宝箱，那么就可以确定那就是真的宝箱。

对于第三组测试点，即使唯一的人指定了真宝箱，由于可能他在说谎且真宝箱为另一个宝箱，所以不存在合法的谜题。