

# 2023 ICPC 国际大学生程序设计竞赛亚洲区域赛

杭州站

12 月 10 日



杭州师范大学  
Hangzhou Normal University



浙江大学  
ZHEJIANG UNIVERSITY



HUAWEI



图森未来



亦赋量化



非凸科技  
non-convex ft.tech

## 题目集

- A 提交
- B 节日装饰
- C 又是最短路查询
- D 运算符优先级
- E 串的周期
- F Top Cluster
- G 控制贪吃蛇
- H 甜蜜的修噶 II
- I 梦游的 Putata
- J 神秘树
- K 卡牌游戏
- L 双修大师 V
- M V 图

请勿在比赛开始前打开

本资料仅作为补充材料，以英文题面为准

浙江大学程序设计竞赛出题组

## Problem A. 提交

Input file: *standard input*  
Output file: *standard output*

本题的背景故事，样例，和注意事项均为虚构。如果有任何和现实比赛，规则，提交，或者团队的雷同，纯属巧合。

在国际挑战谜题大赛中（International Challenging Puzzle Contest, ICPC），有  $m$  个提交。这  $m$  个提交的列表会按时间顺序给你。一个提交可以表示为一个元组  $(c, p, t, s)$ ，表示在时刻  $t$ ，团队  $c$  在问题  $p$  上作出了状态为  $s$  的提交。一个提交的状态要么是“accepted”，要么是“rejected”。

一个团队的分数是团队的解决问题数量和总消耗时间<sup>†</sup>的有序对。解决问题的数量越多，分数越高。如果相同，总消耗时间越小，分数越高。

如果团队  $c$  在问题  $p$  上作出了至少一个状态为“accepted”的提交，我们说团队  $c$  解决了问题  $p$ 。当获得更高分数的队伍数量小于  $\min(\lceil 0.1 \times n \rceil, 35)$  时，一个团队可以获得金牌，这里  $n$  表示解决了至少一个问题的队伍数量， $\lceil x \rceil$  表示最小的不小于  $x$  的整数。

你需要找到所有可以在  $m$  个提交中至多一个状态被改变时可以获得金牌的队伍。

<sup>†</sup> 总消耗时间是所有被解决问题的消耗时间之和（如果没有被解决问题则为 0）。一个被解决问题的消耗时间是第一个状态为“accepted”的提交时间加上这个提交之前的提交数量的 20 倍。请注意，我们称提交  $i$  在提交  $j$  之前，当且仅当提交  $i$  在  $m$  个提交的列表中出现于提交  $j$  之前。

### Input

每个测试点包含多个测试用例。第一行包含一个整数  $t$  ( $1 \leq t \leq 10^5$ )，表示测试用例的数量。

对于每个测试用例，第一行包含一个整数  $m$  ( $1 \leq m \leq 10^5$ )，表示提交的数量。

接下来第  $m$  行中的第  $i$  行包含  $c_i, p_i, t_i, s_i$ ，表示在时刻  $t_i$ ，团队  $c_i$  在问题  $p_i$  上作出了状态为  $s_i$  的提交。具体地：

- $c_i$  是一个长度在 1 和 20 之间的字符串，由大写字母，小写字母，数字和下划线 ‘\_’ 组成。注意不会存在两个团队有相同的名字。
- $t_i$  是一个小于 300 的非负整数。
- $p_i$  是一个大写字母。
- $s_i$  要么是字符串“accepted”，要么是字符串“rejected”。

保证对于所有  $i < j$  有  $t_i \leq t_j$ 。请注意，当  $t_i = t_j$  并且  $i < j$  时，我们仍然说第  $i$  个提交在第  $j$  个提交之前。

保证所有测试用例中  $m$  的和不超过  $10^5$ 。

### Output

对于每个测试用例：

在第一行中输出一个整数  $k$ ，表示可以在  $m$  个提交中至多一个状态被改变时可以获得金牌的队伍的数量。

在第二行，以任意顺序输出  $k$  个不同的字符串，表示这  $k$  个队伍的名字。

## Examples

<i>standard input</i>	<i>standard output</i>
2 5 TSxingxing10 G 0 rejected TSxingxing10 B 83 accepted aoliaoligeiliao J 98 accepted TS1 J 118 accepted TS1 B 263 accepted 12 AllWayTheNorth A 0 rejected YaoYaoLingXian Y 10 accepted XuejunXinyoudui1 X 200 rejected XuejunXinyoudui1 X 200 accepted LetItRot L 215 accepted AllWayTheNorth W 250 accepted ImYourFan I 257 accepted ImYourFan Y 257 accepted AllWayTheNorth T 264 accepted XuejunXinyoudui1 J 294 accepted LetItRot I 299 accepted LetItRot I 299 rejected	2 TSxingxing10 TS1 4 AllWayTheNorth XuejunXinyoudui1 LetItRot ImYourFan

<i>standard input</i>	<i>standard output</i>
2 2 jiangly_fan A 1 accepted jiangly B 23 accepted 3 conqueror_of_tourist A 1 accepted conqueror_of_tourist A 2 accepted tourist B 23 accepted	2 jiangly_fan jiangly 1 conqueror_of_tourist

<i>standard input</i>	<i>standard output</i>
2 13 A A 1 accepted A X 1 accepted K K 1 rejected B B 2 accepted C C 2 accepted D D 2 accepted E E 2 accepted F F 2 accepted G G 2 accepted H H 2 accepted I I 2 accepted J J 2 accepted K K 2 rejected 12 A A 1 accepted A X 1 accepted B B 2 accepted C C 2 accepted D D 2 accepted E E 2 accepted F F 2 accepted G G 2 accepted H H 2 accepted I I 2 accepted J J 2 rejected K K 2 rejected	11 A K B C D E F G H I J 1 A

<i>standard input</i>	<i>standard output</i>
2 11 A A 1 accepted B B 1 accepted C C 2 accepted D D 2 accepted E E 2 accepted F F 2 accepted G G 2 accepted H H 2 accepted I I 2 accepted J J 2 accepted K K 2 accepted 12 A A 1 accepted A X 1 accepted K K 1 rejected B B 2 accepted C C 2 accepted D D 2 accepted E E 2 accepted F F 2 accepted G G 2 accepted H H 2 accepted I I 2 accepted J J 2 accepted	2 A B 2 A K

## Note

在第一个样例的第一个测试用例中，TS1 解决了两个问题，所以他们可以获得金牌。TSxingxing10 可以通过改变他们第一个提交的状态为“accepted”来获得金牌。

在第一个样例的第二个测试用例中，AllWayTheNorth, XuejunXinyoudui1, LetItRot 和 ImYourFan 有同样的分数，解决问题数量为 2 以及总消耗时间为 514。如果没有提交发生改变，他们可以同时获得金牌。

## Problem B. 节日装饰

Input file: *standard input*  
Output file: *standard output*

为了庆祝即将到来的冬季节日，比特镇的中心大街（可以看作是  $x$  轴）装饰上了  $n$  盏彩色路灯，依次编号为  $1, 2, \dots, n$ 。第  $i$  盏灯的坐标为  $x_i$ 、颜色为  $c_i$ 。任意两盏灯的坐标两两不同。

你需要回答  $q$  个询问。对于第  $i$  个询问，你将获得一个整数  $d_i$  ( $1 \leq d_i \leq 250\,000$ )，你需要找到编号最小的路灯  $u$  ( $1 \leq u \leq n$ )，使得在坐标  $x_u + d_i$  处存在另一盏颜色与  $c_u$  不同的路灯，或确定无法找到这样的  $u$ 。你的答案被认为是正确的当且仅当它与标准答案的绝对或相对误差不超过  $0.5$ 。

### Input

第一行包含两个整数  $n$  和  $q$  ( $1 \leq n, q \leq 250\,000$ )，分别表示路灯和询问的数量。

接下来  $n$  行，每行包含两个整数  $x_i$  和  $c_i$  ( $1 \leq x_i \leq 250\,000$ ,  $1 \leq c_i \leq n$ )，依次表示每盏路灯的坐标和颜色。输入保证任意两盏灯的坐标两两不同。

接下来  $q$  行，每行一个整数  $d_i$  ( $1 \leq d_i \leq 250\,000$ )，依次描述每个询问。

### Output

对于每个询问，输出一行一个数，即你找到的最小编号  $u$ 。如果无法找到这样的  $u$ ，请输出  $0$ 。

你的答案被认为是正确的当且仅当它与标准答案的绝对或相对误差不超过  $0.5$ 。

更准确地来说，假设你的答案是  $x$ 、标准答案为  $y$ ，那么你的答案将被认为是正确的当且仅当  $\frac{|x-y|}{\max(1, |y|)} \leq 0.5$ 。

### Example

<i>standard input</i>	<i>standard output</i>
4 5	3
3 1	2
1 2	1
5 1	2
6 2	0
1	
2	
3	
4	
5	

## Problem C. 又是最短路查询

Input file: *standard input*  
Output file: *standard output*

在图论中，平面图是可以画在平面上并且使得不同的边可以互不交叠的图。而如果一个图无论怎样都无法画在平面上，并使得不同的边互不交叠，那么这样的图不是平面图，或者称为非平面图。一个将平面图画在平面上的方法称为平版图，又称为图的平面嵌入，更精确地说，平版图包含一个平面图与一个映射，此映射将平面图的顶点对应到平面上的一点，边对应到一条平面曲线段，满足边两端点对应到线段的两端点，并且线段之间除了在端点之外都不相交。

给定一张  $n$  个点  $m$  条边的无向平面图。点的编号依次为  $1, 2, \dots, n$ 。第  $i$  条边的两个顶点分别为  $u_i$  和  $v_i$ ，其边长为  $w_i$ 。

你需要回答  $q$  个询问。对于第  $i$  个询问，你将获得两个整数  $s_i$  和  $t_i$ 。请编写一个程序，计算从顶点  $s_i$  到顶点  $t_i$  的包含至多三条边的最短路径的长度，或判断这样的路径不存在。

### Input

第一行包含两个正整数  $n$  和  $m$  ( $2 \leq n \leq 10^6, 1 \leq m \leq 10^6$ )，分别表示顶点和无向边的数量。

接下来  $m$  行，第  $i$  行包含三个正整数  $u_i, v_i$  和  $w_i$  ( $1 \leq u_i, v_i \leq n, u_i \neq v_i, 1 \leq w_i \leq 10^8$ )，表示第  $i$  条边。输入保证每对点之间存在至多一条无向边，且输入的图是平面图。

接下来一行包含一个正整数  $q$  ( $1 \leq q \leq 10^6$ )，表示询问的数量。

接下来  $q$  行，第  $i$  行包含两个正整数  $s_i$  和  $t_i$  ( $1 \leq s_i, t_i \leq n, s_i \neq t_i$ )，表示第  $i$  个询问。

### Output

对于每个询问，输出一行一个整数，即包含至多三条边的最短路径的长度。如果找不到这样的路径，请输出“-1”。

## Examples

<i>standard input</i>	<i>standard output</i>
6 9 1 2 4 2 3 6 3 6 5 6 5 3 5 4 2 4 1 3 3 4 9 1 3 100 5 3 1 5 1 3 1 6 3 4 3 5 2 5	6 8 3 1 7
6 4 1 2 1 2 3 1 3 4 1 4 5 1 3 1 4 1 5 1 6	3 -1 -1

## Problem D. 运算符优先级

Input file: *standard input*  
Output file: *standard output*

Randias 正在做他的小学作业:

找出长度为  $2n$  的非零整数序列  $a$  满足:

$$(a_1 \times a_2) + (a_3 \times a_4) + \cdots + (a_{2n-1} \times a_{2n}) \\ = a_1 \times (a_2 + a_3) \times (a_4 + a_5) \times \cdots \times (a_{2n-2} + a_{2n-1}) \times a_{2n} \neq 0$$

形式化地,  $\sum_{i=1}^n a_{2i-1} a_{2i} = a_1 a_{2n} \prod_{i=2}^n (a_{2i-2} + a_{2i-1}) \neq 0$ 。

Randias 当然知道怎么解。但他想考考你。你能解决上面的问题吗?

### Input

每个测试点包含多个测试用例。第一行包含一个整数  $t$  ( $1 \leq t \leq 10^5$ ), 表示测试用例的数量。

对于每个测试用例, 唯一一行包含一个整数  $n$  ( $2 \leq n \leq 10^5$ )。

保证所有测试用例中的  $n$  之和不超过  $2 \cdot 10^5$ 。

### Output

对于每个测试用例, 输出一行共  $2n$  个整数, 表示  $a_1, a_2, \dots, a_{2n}$  ( $1 \leq |a_i| \leq 10^{10}$ )。

可以证明答案总是存在的。

### Example

<i>standard input</i>	<i>standard output</i>
3	1 -3 -3 1
2	1 -10 6 6 -10 1
3	1 -15 10 -1 -1 10 -15 1
4	



## Problem E. 串的周期

Input file: *standard input*  
Output file: *standard output*

Randias 有  $n$  个字符串  $s_1, s_2 \dots s_n$ 。

对于两个字符串  $a = \overline{a_0 a_1 \dots a_{p-1}}$  和  $b = \overline{b_0 b_1 \dots b_{q-1}}$ ，如果对于所有的  $i$  ( $0 \leq i < q$ )， $b_i = a_{i \bmod p}$ ，我们说  $a$  是  $b$  的一个周期。

现在，Randias 可以进行如下操作：

- 选择一个串  $s_i$  和两个下标  $j, k$  ( $0 \leq j, k < |s_i|$ )，交换  $s_{i,j}$  和  $s_{i,k}$ 。

他可以进行这个操作任意多次。他想要在进行完所有操作后，对于所有  $1 < i \leq n$ ， $s_{i-1}$  是  $s_i$  的一个周期。

请帮他求出最终的字符串，或者指出这是不可能的。

### Input

每个测试点包含多个测试用例。第一行包含一个整数  $t$  ( $1 \leq t \leq 10^4$ )。

对于每个测试用例，第一行包含一个整数  $n$  ( $1 \leq n \leq 10^5$ )。

接下来的  $n$  行，第  $i$  行包含一个串  $s_i$  ( $1 \leq |s_i| \leq 5 \cdot 10^6$ )。保证字符串中只包含小写英文字母。

保证一个测试点的所有测试用例中  $n$  的和不超过  $10^5$ ， $|s_i|$  的和不超过  $5 \cdot 10^6$ 。

### Output

对于每个测试用例，如果可以使得对于所有  $1 < i \leq n$ ，满足  $s_{i-1}$  是  $s_i$  的一个周期，输出一行 “YES”。接下来输出  $n$  行  $n$  个串，第  $i$  个串  $s'_i$  代表操作后的  $s_i$ 。如果有多组解，输出任意一组。

如果不可能满足该条件，输出 “NO”。

### Example

<i>standard input</i>	<i>standard output</i>
4	NO
2	YES
abc	abbca
abcd	abbc
4	abbcabb
bbcaa	a
cabb	YES
acabbbb	ab
a	aba
3	abaabaab
ab	NO
aab	
bbaaaaab	
3	
ab	
aab	
bbaaaaaa	

## Problem F. Top Cluster

Input file: *standard input*  
Output file: *standard output*

Top Cluster 是一种非常有用的数据结构，用于维护树上的信息。利用 Top Cluster，我们可以高效地进行范围查询。

可爱的 EMmm 非常喜欢数据结构科技。她现在正在学习 Top Cluster，并尝试解决一道数据结构题。在座的你能不能写个程序和 EMmm 一起解决这道数据结构题？

你将得到一棵  $n$  个点的树，点的编号依次为  $1, 2, \dots, n$ 。第  $i$  个点的点权为非负整数  $w_i$ 。所有点的点权都是两两不同的。

你需要回答  $q$  个询问。对于第  $i$  个询问，你将获得两个整数  $x_i$  和  $k_i$  ( $1 \leq x_i \leq n, 0 \leq k_i \leq 10^{15}$ )，你需要回答  $\text{mex}(\{w_u \mid \text{dist}(u, x_i) \leq k_i \wedge 1 \leq u \leq n\})$  的值。

在这里， $\text{dist}(u, v)$  表示点  $u$  到点  $v$  的最短路径的长度。在数学中，一个集合的  $\text{mex}$  (“minimum excluded value”) 被定义为不属于该集合的最小非负整数。

EMmm 擅长解决  $\text{mex}$  问题。她发现，当所有点的点权两两不同时，上述问题等价于找到最小的非负整数，满足它要么出现在给定邻域外，即  $\text{dist}(x_i, u) > k_i$ ，要么从未在整棵树中出现过。然而，她的思路无法继续深入下去了。你能帮她解决这个问题吗？

### Input

第一行包含两个正整数  $n$  和  $q$  ( $1 \leq n, q \leq 5 \cdot 10^5$ )，分别表示点数和询问数。

第二行包含  $n$  个整数  $w_1, w_2, \dots, w_n$  ( $0 \leq w_i \leq 10^9$ )，依次表示每个点的点权。输入保证所有点的点权都是两两不同的。

接下来  $n - 1$  行，每行包含三个整数  $u_i, v_i$  和  $l_i$  ( $1 \leq u_i, v_i \leq n, u_i \neq v_i, 1 \leq l_i \leq 10^9$ )，表示一条连接点  $u_i$  和点  $v_i$  的长度为  $l_i$  的双向树边。保证输入的是一棵合法的树。

接下来  $q$  行，每行两个整数  $x_i$  和  $k_i$  ( $1 \leq x_i \leq n, 0 \leq k_i \leq 10^{15}$ )，依次描述每个询问。

### Output

对于每个询问，输出一行一个整数，即  $\text{mex}$  的值。

### Example

<i>standard input</i>	<i>standard output</i>
5 4	1
3 9 0 1 2	0
1 2 10	3
3 1 4	4
3 4 3	
3 5 2	
3 0	
1 0	
4 6	
4 7	

## Problem G. 控制贪吃蛇

Input file: *standard input*

Output file: *standard output*

Putata 正在他的笔记本电脑上玩一个网红贪吃蛇游戏，控制一条贪吃蛇在  $n \times m$  的网格上移动。蛇可以表示为有序对的序列  $(x_1, y_1) \rightarrow (x_2, y_2) \rightarrow \cdots \rightarrow (x_k, y_k)$ 。这里  $k$  表示蛇的长度，蛇头在  $(x_1, y_1)$  处，蛇尾在  $(x_k, y_k)$  处。

在这个游戏中，玩家可以发送一串字符串来控制贪吃蛇行动，每个字符表示一个命令。一共有以下五种类型的命令：

- ‘L’：控制蛇头往左移动一步至  $(x_1, y_1 - 1)$ 。
- ‘R’：控制蛇头往右移动一步至  $(x_1, y_1 + 1)$ 。
- ‘U’：控制蛇头往上移动一步至  $(x_1 - 1, y_1)$ 。
- ‘D’：控制蛇头往下移动一步至  $(x_1 + 1, y_1)$ 。
- ‘S’：缩短一节蛇身。蛇尾的那节将消失。蛇的长度将变为  $k - 1$ 。玩家不能在  $k = 1$  时进行该操作。

当蛇头移动时，身体的各个部分也会相应地移动。具体来说，第  $i$  ( $2 \leq i \leq k$ ) 节身体将移动至第  $(i - 1)$  节身体在移动之前所处的位置。蛇不能碰到任何障碍，也不能移动到网格之外。此外，蛇不能与自己相撞，所以玩家应该确保任意两节身体不会处于同一个位置。

考虑下面的边界情况：头部位于  $(x_1, y_1)$ ，尾部位于  $(x_k, y_k)$ ，头部将要移动至  $(x'_1, y'_1)$ 。那么如果  $(x'_1, y'_1) = (x_k, y_k)$ ，则这次移动是合法的。因为在现实世界中，头部永远不可能追上尾部。所以当  $k = 2$  时，玩家可以通过一条指令完成头部和尾部的交换。

给定网格地图以及蛇的身体序列。令  $f(i, j)$  表示要让蛇头位于  $(i, j)$ ，Putata 需要发送的最小命令数。你需要计算：

$$\left( \sum_{i=1}^n \sum_{j=1}^m f(i, j)^2 \right) \bmod 2^{64}$$

注意如果蛇头无法到达  $(i, j)$ ， $f(i, j)$  将会被定义为 0。

### Input

第一行包含三个整数  $n, m$  和  $k$  ( $1 \leq n, m \leq 3000, 1 \leq k \leq \min\{nm, 10^5\}$ )，分别表示网格的行列数以及蛇身的长度。

接下来  $k$  行，第  $i$  行包含两个整数  $x_i$  和  $y_i$  ( $1 \leq x_i \leq n, 1 \leq y_i \leq m, |x_i - x_{i+1}| + |y_i - y_{i+1}| = 1$ )，表示第  $i$  节身体的位置。输入保证任意两节身体不会处于同一个位置，且每节身体一定位于写着 ‘.’ 的格子上。

接下来  $n$  行，第  $i$  行包含一个仅由字符 ‘.’ 和 ‘#’ 组成的长度为  $m$  的字符串。格子  $(i, j)$  是障碍当且仅当第  $i$  行的第  $j$  个字符是 ‘#’。

### Output

输出一行一个整数，即答案。

## Examples

<i>standard input</i>	<i>standard output</i>
4 5 5 3 5 3 4 3 3 3 2 4 2 ..... ..... ..... .....	293
2 2 4 1 1 1 2 2 2 2 1 .. ..	14
5 5 3 1 2 1 1 2 1 ..... .###. .#.#. .###. .....	407

## Problem H. 甜蜜的修噶 II

Input file: *standard input*  
Output file: *standard output*

修噶很甜。

有  $n$  个小朋友想要糖。Prof. Chen 给小朋友们分糖。第  $i$  个小朋友最开始有  $a_i$  包糖。有  $n$  个事件以均匀等概率随机顺序发生。第  $i$  个事件为：

- 如果第  $i$  个小朋友的糖的包数严格少于第  $b_i$  个小朋友，那么第  $i$  个小朋友会得到额外的  $w_i$  包糖。

现在，由于事件是随机顺序发生的，Prof. Chen 的助手 Randias 希望知道每个小朋友在所有事件结束后得到糖果数量的期望。

可以证明答案总可以被表示为一个最简分数  $\frac{x}{y}$ ，其中  $x$  和  $y$  为整数并且  $y \not\equiv 0 \pmod{10^9 + 7}$ 。你需要输出等于  $x \cdot y^{-1} \pmod{10^9 + 7}$  的整数。换言之，输出一个整数  $a$  满足  $0 \leq a < 10^9 + 7$  并且  $a \cdot y \equiv x \pmod{10^9 + 7}$ 。

### Input

每个测试点包含多个测试用例。第一行包含一个整数  $t$  ( $1 \leq t \leq 5 \cdot 10^5$ )，代表测试用例的数量。

对于每个测试用例，第一行包含一个整数  $n$  ( $1 \leq n \leq 5 \cdot 10^5$ )，代表小朋友的数量。

接下来一行包含  $n$  个整数  $a_i$  ( $1 \leq a_i \leq 10^9$ )，代表小朋友初始的糖果数量。

接下来一行包含  $n$  个整数  $b_i$  ( $1 \leq b_i \leq n$ )。

接下来一行包含  $n$  个整数  $w_i$  ( $1 \leq w_i \leq 10^9$ )。

保证每个测试点中  $n$  的总和不超过  $5 \cdot 10^5$ 。

### Output

对于每个测试用例，在一行内输出  $n$  个整数，代表每个小朋友会得到糖果数量的期望。输出答案对  $10^9 + 7$  取模的结果。

### Example

<i>standard input</i>	<i>standard output</i>
4	500000007 5 5 6
4	5 10 9
2 5 5 2	166666673 5 6
4 2 1 3	500000006 4 3 4 5
3 2 1 4	
3	
5 4 3	
1 1 1	
6 6 6	
3	
5 4 3	
2 3 1	
1 2 3	
5	
2 1 3 2 1	
5 1 1 3 4	
1 3 4 2 4	

## Problem I. 梦游的 Putata

Input file: *standard input*

Output file: *standard output*

Putata 梦见他在一个大小为  $n \times m$  的幻影网格世界中迷路了。在这个网格世界中，行从上到下依次被编号为 0 到  $n-1$ ，列从左到右依次被编号为 0 到  $m-1$ 。Putata 不知道如何逃离幻影世界，所以他决定随机游走。假设 Putata 现在位于  $(x, y)$ ，他将会：

- 以  $\frac{l(x,y)}{100}$  的概率移动至  $(x, (y-1) \bmod m)$ 。
- 以  $\frac{r(x,y)}{100}$  的概率移动至  $(x, (y+1) \bmod m)$ 。
- 以  $\frac{u(x,y)}{100}$  的概率移动至  $((x-1) \bmod n, y)$ 。
- 以  $\frac{d(x,y)}{100}$  的概率移动至  $((x+1) \bmod n, y)$ 。

你需要执行  $q$  个操作，每个操作是下面两种之一：

- “1  $x\ y\ L\ R\ U\ D$ ” ( $0 \leq x < n, 0 \leq y < m, 1 \leq L, R, U, D \leq 100, L + R + U + D = 100$ ): 将  $l(x, y), r(x, y), u(x, y)$  和  $d(x, y)$  的值分别修改为  $L, R, U$  和  $D$ 。
- “2  $sx\ sy\ tx\ ty$ ” ( $0 \leq sx, tx < n, 0 \leq sy, ty < m, (sx, sy) \neq (tx, ty)$ ): 假设 Putata 现在位于  $(sx, sy)$ ，请帮他计算出第一次到达目标位置  $(tx, ty)$  所需的期望步数。

### Input

第一行包含两个整数  $n$  和  $m$  ( $3 \leq n \leq 10^5, 3 \leq m \leq 5$ )，分别表示幻影网格世界的行数和列数。

接下来  $n$  行，第  $i$  行包含  $m$  个整数  $l(i-1, 0), l(i-1, 1), \dots, l(i-1, m-1)$  ( $1 \leq i \leq n, 1 \leq l(\cdot, \cdot) \leq 100$ )。

接下来  $n$  行，第  $i$  行包含  $m$  个整数  $r(i-1, 0), r(i-1, 1), \dots, r(i-1, m-1)$  ( $1 \leq i \leq n, 1 \leq r(\cdot, \cdot) \leq 100$ )。

接下来  $n$  行，第  $i$  行包含  $m$  个整数  $u(i-1, 0), u(i-1, 1), \dots, u(i-1, m-1)$  ( $1 \leq i \leq n, 1 \leq u(\cdot, \cdot) \leq 100$ )。

接下来  $n$  行，第  $i$  行包含  $m$  个整数  $d(i-1, 0), d(i-1, 1), \dots, d(i-1, m-1)$  ( $1 \leq i \leq n, 1 \leq d(\cdot, \cdot) \leq 100$ )。

输入数据保证对于任意一个格子  $(i, j)$  ( $0 \leq i < n, 0 \leq j < m$ ) 都有  $l(i, j) + r(i, j) + u(i, j) + d(i, j) = 100$ 。

接下来一行包含一个整数  $q$  ( $1 \leq q \leq 3 \cdot 10^4$ )，表示操作的数量。

接下来  $q$  行，每行描述一个操作。

### Output

对于每个询问，输出一行一个整数，表示 Putata 第一次到达目标位置  $(tx, ty)$  所需的期望步数。

更确切地说，假设答案的最简分数形式是  $\frac{p}{q}$ ，你应该输出最小的非负整数  $r$ ，使得  $q \cdot r \equiv p \pmod{10^9 + 7}$ 。你可以认为在所有的测试数据中，这样的  $r$  始终存在。

## Example

<i>standard input</i>	<i>standard output</i>
4 3	76426175
1 2 3	344136684
4 5 6	555192113
7 8 9	
10 11 12	
23 24 25	
26 27 28	
29 30 31	
32 33 34	
10 11 12	
13 14 15	
16 17 18	
19 20 21	
66 63 60	
57 54 51	
48 45 42	
39 36 33	
4	
2 0 1 1 1	
2 0 0 3 2	
1 1 1 25 25 25 25	
2 0 0 3 2	

## Problem J. 神秘树

Input file: *standard input*  
Output file: *standard output*

这是一道交互题。

Randias 有一个未知的  $n$  个顶点的树。这棵树可以是一个菊花或者链。现在, Randias 需要确定这棵树是菊花还是链。他可以问以下问题至多  $\lceil \frac{n}{2} \rceil + 3$  次:

- 是否存在一条边连接顶点  $u$  和顶点  $v$  ( $1 \leq u, v \leq n, u \neq v$ )?

他需要确定树的类型。帮助他提出这些问题并确定答案。

如果存在一个顶点  $u$ , 使得对于每个其他顶点  $v$ , 都存在边  $(u, v)$ , 则树被称为菊花。

如果存在一个排列  $p_1, p_2, \dots, p_n$ , 使得对于每个  $i$  ( $1 \leq i < n$ ), 都存在边  $(p_i, p_{i+1})$ , 则树被称为链。

长度为  $n$  的排列是一列数, 其中 1 到  $n$  的每个整数恰好出现一次。

请注意, 交互器是自适应的, 这意味着答案不一定是预先确定好的, 交互器可以任意选择, 但答案不会在交互过程中与之前返回的结果产生矛盾。

### Input

每个测试点包含多个测试用例。对于每个测试点, 第一行包含一个整数  $t$  ( $1 \leq t \leq 250$ ), 表示测试用例数量。

对于每个测试用例, 第一行包含一个整数  $n$  ( $4 \leq n \leq 1000$ ), 表示顶点的数量。保证每个测试点中  $n$  的总和不超过 1000。

### Interaction Protocol

在每个测试用例中, 你可以进行不超过  $\lceil \frac{n}{2} \rceil + 3$  次询问。对于一个询问, 单独输出一行 “?  $u$   $v$ ” ( $1 \leq u, v \leq n, u \neq v$ ), 接下来你要从标准输入读取回应。

对于一个询问, 如果  $u$  与  $v$  之间没有边, 交互器会输出“0”, 如果  $u$  与  $v$  之间有边, 交互器会输出“1”。

如果要给出答案, 当这棵树是一条链时, 单独输出一行 “! 1”, 当这棵树是一个菊花时, 输出 “! 2”。输出答案不会被计算在限制的  $\lceil \frac{n}{2} \rceil + 3$  次询问之中。

输出完答案后, 你的程序应当处理下一个测试用例, 如果这是最后一个测试用例, 你的程序应当终止。

在输出一个询问之后, 不要忘记输出换行符并清空输出缓冲区。要做到这一点, 可以在 C++ 中使用 `fflush(stdout)` 或 `cout.flush()`, 在 Java 中使用 `System.out.flush()`, 或者在 Python 中使用 `stdout.flush()`。



## Example

<i>standard input</i>	<i>standard output</i>
2	
4	? 1 2
1	? 2 3
1	? 3 4
1	! 1
4	? 1 3
1	? 2 4
0	? 1 2
0	? 1 4
0	! 2

## Problem K. 卡牌游戏

Input file: *standard input*  
Output file: *standard output*

Randias 正在玩一种卡牌游戏。在这个游戏中，每张卡牌上都写有一个数字。对于卡牌  $a_1, a_2 \dots a_m$ ，Randias 将按照以下方式进行游戏。

最初，所有卡牌都在他的手中。Randias 将维护一个卡牌序列（最初为空）。在第  $i$  次操作中，Randias 将把第  $i$  张卡牌（这张卡牌上写有数字  $a_i$ ）放在卡牌序列的末尾。然后：

- 如果在序列中没有其他写有数字  $a_i$  的卡牌，则第  $i$  次操作结束。
- 否则，设卡牌序列中第  $j$  张卡牌上写有数字  $a_i$ 。Randias 将拿走从第  $j$  张卡牌到新放置的卡牌之间的所有卡牌，包括第  $j$  张卡牌和新放置的卡牌。

例如，假设  $a = [2, 1, 3, 1, 2, 3]$ ，并且卡牌序列  $s = []$  最初为空。

在第 1 次操作后， $s = [2]$ 。

在第 2 次操作后， $s = [2, 1]$ 。

在第 3 次操作后， $s = [2, 1, 3]$ 。

在第 4 次操作后， $s = [2]$ （卡牌 1, 3, 1 被拿走）。

在第 5 次操作后， $s = []$ （卡牌 2, 2 被拿走）。

在第 6 次操作后， $s = [3]$ 。

现在，给定  $n$  张卡牌  $a_1, a_2 \dots a_n$ 。Randias 有  $q$  次询问，第  $i$  次询问给出两个整数  $l_i, r_i$ ，他想知道如果初始卡牌列表是  $a_{l_i}, a_{l_i+1} \dots a_{r_i}$ ，那么最终的卡牌序列上将剩下多少张卡牌？

出于一些原因，Randias 希望你在线地回答这些询问。即你需要通过上一次询问的答案来解码出下一次询问。

### Input

第一行包含两个整数  $n, q$  ( $1 \leq n, q \leq 3 \cdot 10^5$ )，表示卡牌数量和询问数量。

接下来一行包含  $n$  个整数  $a_1, a_2 \dots a_n$  ( $1 \leq a_i \leq n$ )。

接下来的  $q$  行，每行包含两个整数  $l'_i, r'_i$  ( $0 \leq l'_i, r'_i \leq 2n$ )。设上一次询问的答案为  $lastans$ ，则  $l_i = l'_i \oplus lastans$ ， $r_i = r'_i \oplus lastans$ ， $1 \leq l_i \leq r_i \leq n$ ，其中  $l_i$  和  $r_i$  代表当前的询问。如果之前没有回答过任何询问， $lastans = 0$ 。

### Output

对于每次询问，输出一个整数代表答案。

## Examples

<i>standard input</i>	<i>standard output</i>
5 5 3 3 1 1 1 5 5 3 4 3 3 0 5 3 5	1 2 1 0 1
7 7 2 4 1 2 3 1 2 1 6 0 4 3 3 0 4 0 3 0 6 2 7	2 1 1 1 2 3 0

## Note

第一个样例中，询问是  $[5, 5]$ ,  $[2, 5]$ ,  $[1, 1]$ ,  $[1, 4]$ ,  $[3, 5]$ 。

## Problem L. 双修大师V

Input file: *standard input*  
Output file: *standard output*

Prof. Chen 是数据结构和计算几何方面的大师。最近，他教会了 Putata 和 Budada 凸多边形的定义。凸多边形是一个简单多边形（即没有两个顶点重合，并且除了相邻边的端点相交之外，没有两条边相交），并且所有内角都小于  $\pi$ 。

Putata 和 Budada 解决了凸包判定问题，但是 Prof. Chen 希望他们更进一步，即他们需要维护一个线段的多重集合  $S$ ，并且支持以下两种询问：

- $+ px\ py\ qx\ qy$ , 将以  $(px, py), (qx, qy)$  为端点的线段加入到多重集合  $S$ 。
- $- i$ , 删除第  $i$  个询问加入的线段。保证第  $i$  个询问是一个加入类型的询问，并且该线段还没有被删除。

在每次询问之后，Putata 和 Budada 需要回答是否存在一个凸多边形  $\mathcal{C}$ ，其中的顶点按照逆时针顺序为  $p_0, p_1, p_2, \dots, p_{m-1}$ ，并且满足对于所有  $u \in S$ ，存在  $j \in \{0, 1, 2, \dots, m-1\}$  满足  $u \subseteq p_j p_{(j+1) \bmod m}$ 。对于两个线段  $e, f$ ，我们有  $e \subseteq f$  当且仅当对于所有点  $z \in e$ ，满足  $z \in f$ 。

请帮助 Putata 和 Budada 解决这个问题。

### Input

每个测试点包含多组测试数据。第一行包含一个整数  $t$  ( $1 \leq t \leq 5 \cdot 10^5$ )，代表测试数据的组数。

对于每组测试数据，第一行包含一个整数  $n$  ( $1 \leq n \leq 5 \cdot 10^5$ )，代表询问的数量。

接下来  $n$  行中的每一行都包含一个询问。询问的开头为一个字符  $op$  ( $op \in \{+, -\}$ )。

如果  $op = +$ ，接下来包含四个整数  $px, py, qx, qy$  ( $-10^9 \leq px, py, qx, qy \leq 10^9$ )，代表一个加入询问。保证  $px \neq qx$  或  $py \neq qy$ 。

否则接下来包含一个整数  $i$  ( $1 \leq i \leq n$ )，代表一个删除询问。保证第  $i$  个询问是一个加入类型的询问，并且该线段还没有被删除。

保证所有数据的  $n$  之和不超过  $5 \cdot 10^5$ 。

### Output

对于每组测试数据，在一行内输出一个包含 '0' 和 '1' 的字符串。第  $i$  个字符为 '1' 当且仅当在第  $i$  次询问后答案为真，否则为 '0'。

## Example

<i>standard input</i>	<i>standard output</i>
4	11000001
8	11011
+ 0 0 1 0	1101
+ 5 5 1 3	1111
+ 2 0 2 1	
+ 9 7 6 2	
+ 1 2 2 2	
- 4	
+ 0 1 0 2	
- 2	
5	
+ 0 0 1 1	
+ 0 1 1 2	
+ 0 2 1 3	
- 2	
+ 1 1 10 10	
4	
+ 0 0 1 1	
+ 0 0 1 0	
+ 0 0 0 1	
- 1	
4	
+ 0 0 1 1	
+ 0 0 1 1	
- 1	
- 2	

## Problem M. V图

Input file: *standard input*  
Output file: *standard output*

对于一个 1 起始下标的长度为  $n$  的整数序列  $a$ , 当  $n \geq 3$  并且存在下标  $i$  ( $1 < i < n$ ) 满足下述条件时, 这个整数序列是 V 图:

- 对于  $1 \leq j < i$ ,  $a_j > a_{j+1}$ ;
- 对于  $i < j \leq n$ ,  $a_j > a_{j-1}$ 。

给定一个 V 图  $a$ , 找到平均值最大的 V 图  $b$ , 满足  $b$  是  $a$  的连续子序列。

注意一个序列的连续子序列可以通过删除这个序列的开始和结尾的若干（可能是零个）元素得到。

### Input

每个测试点包含多个测试用例。第一行包含一个整数  $t$  ( $1 \leq t \leq 10^5$ ), 表示测试用例的数量。

对于每个测试用例, 第一行包含一个整数  $n$  ( $3 \leq n \leq 3 \cdot 10^5$ ), 表示整数序列  $a$  的长度。

第二行包含  $n$  个整数  $a_1, a_2, \dots, a_n$  ( $1 \leq a_i \leq 10^9$ ), 表示整数序列  $a$ 。

保证  $a$  是 V 图, 并且所有测试用例的  $n$  之和不超过  $3 \cdot 10^5$ 。

### Output

对于每个测试用例, 输出一个实数表示平均值的最大值。

你的答案被认为是正确的当且仅当它与标准答案的绝对或相对误差不超过  $10^{-9}$ 。

更准确地来说, 假设你的答案是  $x$ 、标准答案为  $y$ , 那么你的答案将被认为是正确的当且仅当  $\frac{|x-y|}{\max(1, |y|)} \leq 10^{-9}$ 。

### Example

<i>standard input</i>	<i>standard output</i>
2	6.75000000000000000000
4	5.83333333333333303727
8 2 7 10	
6	
9 6 5 3 4 8	