第十五届山东省 CCPC 大学生程序设计竞赛

正式赛

2025 年 5 月 25 日

试题列表

A	项目管理
B	弹球
C	括号整数
D	分布式系统
E	最大公因数
F	ACE 字符串
G	装配线
H	最小生成树
I	方阵谜题
J	有用的算法
K	路径规划 2
L	恒星
M	三角剖分

本试题册共 13 题，19 页。
如果您的试题册缺少页面，请立即通知志愿者。

承办方

命题方

竞赛过程中访问非竞赛网页是违反竞赛规则的行为。
如果您有兴趣（我们很荣幸），
请在竞赛后扫描二维码。

Problem A. 项目管理

某公司里有 n 名员工，第 i 名员工的职级是 a_i。公司即将启动一个大项目，需要尽可能多的人参加。然而，员工们不太喜欢职级比他们高的同事。为了让项目里的每个人都愿意一起工作，对于所有 $1 \le i \le n$，若第 i 名员工参加了项目，则项目中至多只能有 b_i 个人职级大于 a_i。

作为公司的老板，您需要选择尽可能多的人参加项目。

Input

有多组测试数据。第一行输入一个整数 T（$1 \le T \le 10^4$）表示测试数据组数，对于每组测试数据：

第一行输入一个整数 n（$1 \le n \le 2 \times 10^5$），表示员工的数量。

对于接下来的 n 行，第 i 行输入两个整数 a_i 和 b_i（$1 \le a_i \le n$，$0 \le b_i < n$），表示第 i 名员工的职级，以及他/她最多愿意和多少个职级更高的人一起工作。

保证所有数据 n 之和不超过 2×10^5。

Output

对于每组数据，首先输出一行一个整数 c，表示可以参加项目的最大人数。接下来输出一行 c 个由单个空格分隔的不同整数 $p_1, p_2, \cdots, p_c$，表示参加项目的员工编号。如果有多种最优答案，您可以输出任意一种。

Example

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

Problem B. 弹球

二维平面上有两条水平线 $y = 0$ 和 $y = H$。在这两条线之间，一开始有 n 块小木板，每块可以被看作一个点。第 i 块木板位于 (x_i, y_i)。

维护 q 次操作，操作有以下三种。

- $+\ x\ y$：往平面上增加一块位于 (x, y) 的木板。
- $-\ x\ y$：从平面上移除一块位于 (x, y) 的木板。
- $?\ x\ y\ v_y\ g$：一颗弹球从 (x, y) 处被释放。

  令 $\vec{v} = (v_x, v_y)$ 表示球的速度（也就是说，若球现在位于 (x, y)，它会在 ϵ 秒后移动到 $(x + v_x\epsilon, y + v_y\epsilon)$）。若 $g \geq x$ 则 $v_x = 1$，否则 $v_x = -1$。v_y 已在询问中给出。v_y 的值要么是 1 要么是 -1。

  当球撞到一块木板或一条水平线时，v_y 将会被反转（也就是说，v_y 变为 $-v_y$），而 v_x 保持不变。

  当弹球的 x 坐标为 g 时，求它的 y 坐标。

Input

有多组测试数据。第一行输入一个整数 T（$1 \leq T \leq 10^4$）表示测试数据组数，对于每组测试数据：

第一行输入三个整数 H，n 和 q（$2 \leq H \leq 10^9$，$1 \leq n, q \leq 10^5$），表示位于上方的水平线的位置，初始的木板数量，以及操作的数量。

对于接下来 n 行，第 i 行输入两个整数 x_i 和 y_i（$1 \leq x_i \leq 10^9$，$1 \leq y_i < H$），表示第 i 块木板的位置。

对于接下来 q 行，第 i 行首先输入一个字符 op（$op \in \{+, -, ?\}$）表示第 i 次操作的种类。

- 若 $op = +$ 则再输入两个整数 x 和 y（$1 \leq x \leq 10^9$，$1 \leq y < H$），表示要添加木板的位置。保证当前没有木板位于 (x, y)。
- 若 $op = -$ 则再输入两个整数 x 和 y（$1 \leq x \leq 10^9$，$1 \leq y < H$），表示要移除的木板的位置。保证当前有一块木板位于 (x, y)。
- 若 $op = ?$ 则再输入四个整数 x，y，v_y 和 g（$1 \leq x, g \leq 10^9$，$1 \leq y < H$，$v_y \in \{-1, 1\}$），表示弹球的初始位置，弹球沿 y 轴方向的初始速度以及目标位置的 x 坐标。保证当前没有木板位于 (x, y)。

保证所有数据 n 之和与 q 之和均不超过 2×10^5。

Output

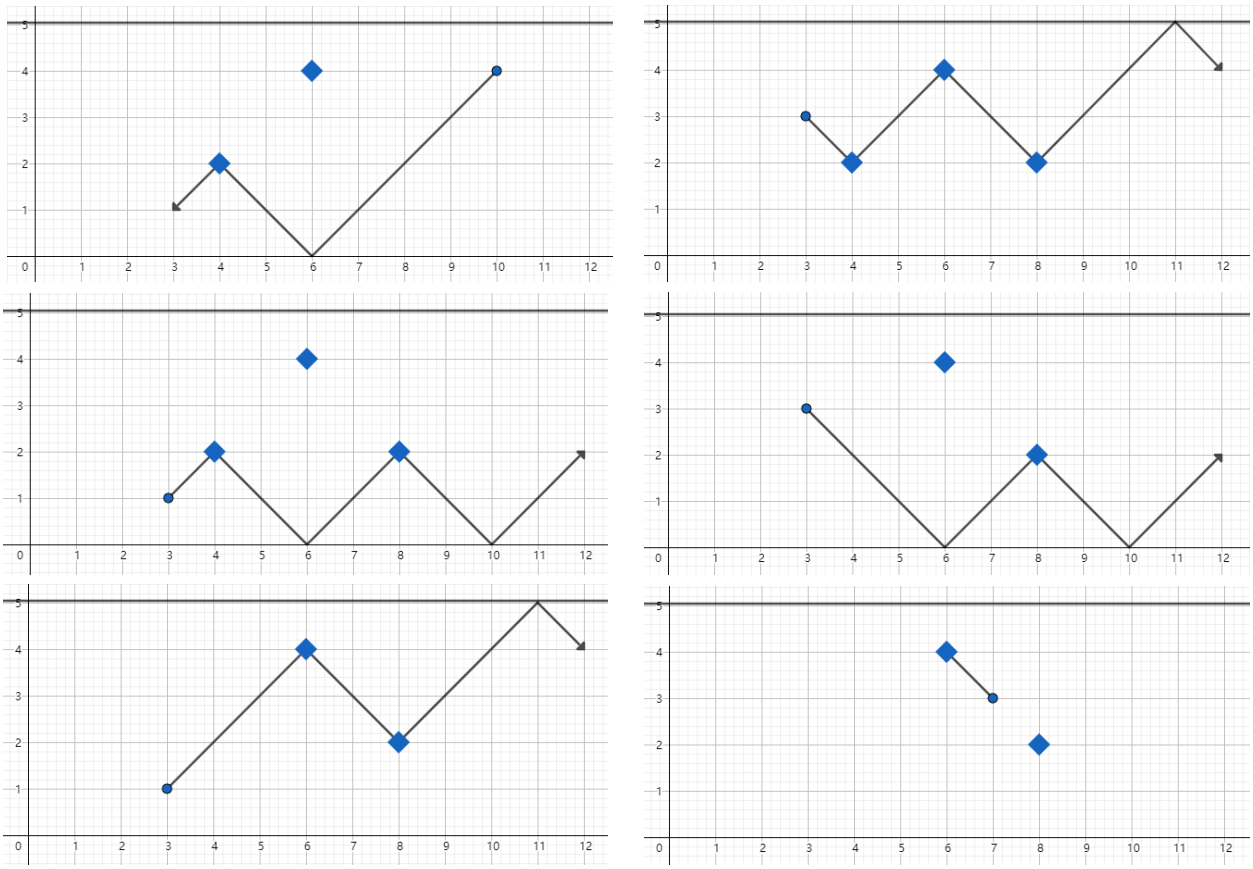
每次第三种操作输出一行一个整数表示答案。

Example

standard input	standard output
2 5 2 8 4 2 6 4 ? 10 4 -1 3 + 8 2 ? 3 3 -1 12 ? 3 1 1 12 - 4 2 ? 3 3 -1 12 ? 3 1 1 12 ? 7 3 1 6 10 1 2 5 5 ? 9 2 1 9 ? 9 2 -1 9	1 4 2 2 4 4 2 2

Note

第一组样例数据的询问如下图所示。木板用菱形表示。



Problem C. 括号整数

令 $({}_w$ 和 $)_w$ 表示权重为 w 的圆括号。平衡带权括号序列（BWBS）的定义如下：

- 空序列是 BWBS。
- 若 P 是 BWBS，则 $({}_wP)_w$ 也是 BWBS。也就是说，开头和结尾的两个括号有相同的权重。
- 若 P 和 Q 都是 BWBS，则 PQ 也是 BWBS。

例如，$({}_1({}_3)_3)_1$ 和 $({}_5({}_7)_7({}_2)_2)_5$ 都是 BWBS，而 $({}_1({}_3)_1)_3$ 和 $)_5({}_5({}_7)_7$ 不是。

考虑一个有 n 位数的，不含前导零的正整数，设它的第 i 高位为 d_i（也就是说，该整数等于 $\sum_{i=1}^{n} d_i \times 10^{n-i}$）。称该整数是一个括号整数，若存在一个平衡带权括号序列，使得第 i 个括号的权重为 d_i。例如，1000022122 是一个括号整数，因为存在一个 BWBS $({}_1({}_0({}_0)_0)_0({}_2)_2)_1({}_2)_2$。

给定一个整数 A，求小于等于 A 的最大括号整数。

Input

有多组测试数据。第一行输入一个整数 T（$1 \le T \le 10^4$）表示测试数据组数，对于每组测试数据：

第一行输入一个整数 A（$11 \le A < 10^{2\times 10^5}$）。

保证所有数据 A 的位数之和不超过 2×10^5。

Output

每组数据输出一行一个整数，表示小于等于 A 的最大括号整数。因为 11 是一个括号整数，因此答案总是存在。请注意，括号整数不能有前导零。

Example

standard input	standard output
4 20250525 11451419198100 1001 1000	20244202 11451418814154 1001 99

Problem D. 分布式系统

某分布式系统有 n 个编号从 0 到 $(n-1)$ 的工作节点。该系统将处理 q 项任务，其中第 i 项任务可以记为两个整数 a_i 与 b_i，表示该任务有 a_i 项编号从 0 到 (a_i-1) 的子任务，且子任务 j 会分配给工作节点 $(b_i+j) \bmod n$。

对于每个 $0 \le k < n$，求所有任务处理完成后，工作节点 k 一共被分配了几项子任务。

Input

有多组测试数据。第一行输入一个整数 T（$1 \le T \le 10^4$）表示测试数据组数，对于每组测试数据：

第一行输入两个整数 n 和 q（$1 \le n, q \le 2 \times 10^5$），表示工作节点的数量以及任务的数量。

对于接下来 q 行，第 i 行输入两个整数 a_i 和 b_i（$1 \le a_i \le 10^9$，$0 \le b_i < n$），表示第 i 项任务。

保证所有数据 n 之和与 q 之和均不超过 2×10^5。

Output

每组数据输出一行 n 个由单个空格分隔的整数 $v_0, v_1, \cdots, v_{n-1}$，其中 v_k 表示所有任务处理完成后，工作节点 k 一共被分配了几项子任务。

Example

standard input	standard output
2 7 3 10 0 4 2 21 1 1 2 200 0 100 0	5 5 6 5 5 5 4 300

Problem E. 最大公因数

给定长度为 n 的正整数序列 $a_1, a_2, \cdots, a_n$，您需要执行恰好 k 次操作。每次操作，您需要选择一个元素，并将它的值增加 1。

在操作结束后，最大化所有元素的最大公因数。

Input

有多组测试数据。第一行输入一个整数 T（$1 \le T \le 10^3$）表示测试数据组数，对于每组测试数据：

第一行输入两个整数 n 和 k（$1 \le n \le 10^6$，$1 \le k \le 10^{12}$）表示序列的长度以及操作的次数。

第二行输入 n 个整数 $a_1, a_2, \cdots, a_n$（$1 \le a_i \le 10^6$）表示序列。

保证所有数据 n 之和不超过 10^6，所有数据 $\max a_i$ 之和不超过 10^6，所有数据 k 之和不超过 10^{12}。

Output

每组数据输出一行一个整数，表示恰好 k 次操作后，所有元素的最大公因数最大可能是多少。

Example

standard input	standard output
2 3 6 2 9 8 3 7 2 9 8	5 2

Note

对于第一组样例数据，可以将序列变为 $5, 10, 10$，最大公因数为 5。

对于第二组样例数据，可以将序列变为 $6, 10, 10$，最大公因数为 2。

Problem F. ACE 字符串

称长度为 k 的字符串 $s_1 s_2 \cdots s_k$ 是 ACE 字符串，若它可以被分成五个非空子串，使得第一个子串、第三个子串和第五个子串相等。

更正式地，称该字符串是 ACE 字符串，若存在两个正整数 p 和 q 满足以下所有限制：

- $3p + 2 \leq k$。
- $p + 2 \leq q \leq k - 2p$。
- 对于所有 $1 \leq i \leq p$，$s_i = s_{q+i-1} = s_{k-p+i}$。

给定一个长度为 n 的字符串，求最长 ACE 子串的长度，或表明这种子串不存在。

Input

有多组测试数据。第一行输入一个整数 T（$1 \leq T \leq 3 \times 10^5$）表示测试数据组数，对于每组测试数据：

第一行输入一个整数 n（$1 \leq n \leq 3 \times 10^5$）表示字符串的长度。

第二行输入一个由小写英文字母构成的长度为 n 的字符串 $s_1 s_2 \cdots s_n$。

保证所有数据 n 之和不超过 3×10^5。

Output

每组数据输出一行一个整数，表示最长 ACE 子串的长度。若这种子串不存在，则输出 `0`。

Example

standard input	standard output
3 9 abcabcabc 6 abaaaa 1 a	8 6 0

Note

对于第一组样例数据，最长的 ACE 子串是 `abcabcab`，因为它可以被分成 `ab`，`c`，`ab`，`c` 和 `ab`，其中第一个子串、第三个子串和第五个子串相等。

对于第二组样例数据，最长的 ACE 子串是 `abaaaa`，因为它可以被分成 `a`，`b`，`a`，`aa` 和 `a`，其中第一个子串、第三个子串和第五个子串相等。

Problem G. 装配线

一条装配线上有 k 名工人，编号从 1 到 k，每名工人都有一个收件箱。工人们将要加工 n 个工件，第 i 个工件将在第 t_i 分钟的开头加入到第 w_i 名工人的收件箱。每分钟内，以下事件将按顺序发生：

1. 新工件（如果有的话）被加入到一些工人的收件箱里。
2. 如果一名工人的收件箱不是空的，他/她会从收件箱里拿起一个工件并加工。每名工人的该事件同时发生。
3. 如果一名工人刚刚加工了工件：
   - 对于工人 $1 \le i < k$，他/她会将该工件放入工人 $(i+1)$ 的收件箱。
   - 对于工人 k，他/她会将该工件放入出货箱，该工件加工完成。

   每名工人的该事件也是同时发生的。

求完成所有 n 个工件的加工需要几分钟。

Input

有多组测试数据。第一行输入一个整数 T（$1 \le T \le 10^4$）表示测试数据组数，对于每组测试数据：

第一行输入两个整数 n 和 k（$1 \le n \le 2 \times 10^5$，$1 \le k \le 10^9$），表示工件的数量以及工人的数量。

对于接下来 n 行，第 i 行输入两个整数 w_i 和 t_i（$1 \le w_i \le k$，$1 \le t_i \le 10^9$），表示第 i 个工件将在第 t_i 分钟的开头加入到第 w_i 名工人的收件箱。

保证所有数据 n 之和不超过 2×10^5。

Output

每组数据输出一行一个整数，表示完成所有 n 个工件的加工需要几分钟。

Example

standard input	standard output
2 4 3 3 2 2 1 3 2 1 2 2 10 1 7 4 20	5 26

Note

第一组样例数据解释如下。下表展示了每分钟第 1 个以及第 3 个事件结束后每个收件箱里的工件数量。数量由一个序列 $\{c_1, c_2, c_3\}$ 给出，其中 c_i 表示第 i 名工人的收件箱里的工件数量。

分钟	事件 1	事件 3
1	$\{0, 1, 0\}$	$\{0, 0, 1\}$
2	$\{1, 0, 3\}$	$\{0, 1, 2\}$
3	$\{0, 1, 2\}$	$\{0, 0, 2\}$
4	$\{0, 0, 2\}$	$\{0, 0, 1\}$
5	$\{0, 0, 1\}$	$\{0, 0, 0\}$

Problem H. 最小生成树

给定一张有 n 个节点与 m 条带权边的无向连通图，您可以往图中再加入至多 k 条边。如果您添加的边连接了节点 u 和 v，则它的边权为 $|u-v|$。

您需要最小化最小生成树里所有边的权重之和。

请注意：图可能含有自环或重边。另外，即使两个点原来已经被一条或多条边相连，也可以选择在两个点间添加一条边。

Input

有多组测试数据。第一行输入一个整数 T（$1 \le T \le 10^4$）表示测试数据组数。对于每组测试数据：

第一行输入三个整数 n，m 和 k（$2 \le n \le 2 \times 10^5$，$n-1 \le m \le 2 \times 10^5$，$0 \le k \le 2 \times 10^5$）表示图的点数，边数，和您能添加的边数。

对于接下来的 m 行，第 i 行输入三个整数 u_i，v_i 和 w_i（$1 \le u_i, v_i \le n$，$0 \le w_i \le 10^9$）表示一条连接节点 u_i 和 v_i，权重为 w_i 的边。

保证给定的图是连通的。另外保证所有数据 n 之和，m 之和，以及 k 之和均不超过 2×10^5。

Output

对于每组数据：

首先输出一行一个整数 c（$0 \le c \le k$），表示您想要添加的边数。

然后输出 c 行，其中第 i 行包含两个整数 u_i 和 v_i（$1 \le u_i, v_i \le n$），表示您添加的第 i 条边连接了节点 u_i 和 v_i，权重为 $|u_i - v_i|$。

接下来输出一行一个整数，表示最小生成树里所有边的权重之和的最小值。

然后输出一行，包含 $(n-1)$ 个由单个空格分隔的不同整数 $e_1, e_2, \cdots, e_{n-1}$（$1 \le e_i \le m+c$），表示最小生成树中边的编号。$1 \le e_i \le m$ 表示一条原图中的边，$e_i > m$ 表示您添加的第 $(e_i - m)$ 条边。

Example

standard input	standard output
3 5 6 2 1 2 3 2 3 5 1 4 7 4 2 4 5 4 8 3 5 1 4 5 100 1 2 2 2 3 0 2 4 0 4 1 1 3 4 3 3 2 0 1 2 100 2 3 100	2 2 3 3 4 6 8 1 6 7 0 1 2 3 4 0 200 1 2

Problem I. 方阵谜题

考虑一个 3 行 3 列的网格。网格里的每个格子都写着一个整数。从 1 到 9（含两端）的每个整数在网格中恰好出现一次。

您可以对网格执行以下三种操作。

操作	描述	范例
右移一行	选择一行，将最右边的元素移动到最左边。	右移第三行 2 9 3 → 2 9 3 6 7 8 → 6 7 8 5 1 4 → 4 5 1
下移一列	选择一列，将最下边的元素移动到最上边。	下移第一列 2 9 3 → 4 9 3 6 7 8 → 2 7 8 4 5 1 → 6 5 1
顺时针旋转	将整个网格顺时针旋转 90 度。	顺时针旋转 4 9 3 → 6 2 4 2 7 8 → 5 7 9 6 5 1 → 1 8 3

给定两个这样的网格，求将第一个网格转换为第二个网格所需的最小操作次数。

Input

有多组测试数据。第一行输入一个整数 T（$1 \le T \le 2 \times 10^5$）表示测试数据组数。对于每组测试数据：

对于前三行，第 i 行输入一个字符串 $a_{i,1}a_{i,2}a_{i,3}$ $(1 \le a_{i,j} \le 9)$，表示第一个网格的第 i 行。

对于接下来的三行，第 i 行输入一个字符串 $b_{i,1}b_{i,2}b_{i,3}$ $(1 \le b_{i,j} \le 9)$，表示第二个网格的第 i 行。

Output

每组数据输出一行。如果可以将第一个网格转换为第二个网格，则输出一个整数，表示所需的最小操作次数；如果无法完成转换，则输出 `-1`。

Example

standard input	standard output
4 293 678 514 624 579 183 624 579 183 293 678 514 123 456 789 321 456 789 123 456 789 123 456 789	3 5 -1 0

Note

对于第一组样例数据，如题目描述中所示，我们可以先右移第三行，然后下移第一列，最后顺时针旋转。

Problem J. 有用的算法

> "Stop learning useless algorithms, go and solve some problems, learn how to use binary search."
>
> – Um_nik

二分法是一种非常有用的算法，在多种问题中均有应用。以下伪代码展示了二分法的一种实现方式。函数 BINARYSEARCH 接受两个参数 A 和 k，其中 $A = a_1, a_2, \cdots, a_n$ 是一个长度为 n 且元素严格单调递增的整数序列，k 是一个在序列 A 中出现的整数。该函数返回整数 k 在序列 A 中的下标。请注意，在以下伪代码中，$\lfloor x \rfloor$ 表示小于等于 x 的最大整数。

Algorithm 1 The Binary Search Algorithm

```
1: function BinarySearch(A, k)
2:     l ← 1
3:     r ← n
4:     while l < r do
5:         m ← ⌊(l+r)/2⌋
6:         if a_m ≥ k then
7:             r ← m
8:         else
9:             l ← m + 1
10:        end if
11:    end while
12:    return l
13: end function
```

二分法的使用有一个前提条件，即查找的序列必须有序。不过在本题中，我们将暂时无视这一限制，研究二分法在任意序列上的表现。

给定两个整数 n 与 k（$1 \le k \le n$），称一个 n 的排列 $P = p_1, p_2, \cdots, p_n$ 是好的，若在排列 P 上使用二分法可以正确地找到整数 k 出现的下标。也就是说，令 $i =$ `BINARYSEARCH(P, k)`，称 P 是好的，若 $p_i = k$。

现从所有 n 的排列里等概率地抽取一个排列，求抽到好的排列的概率。

Input

有多组测试数据。第一行输入一个整数 T（$1 \le T \le 10^4$）表示测试数据组数，对于每组测试数据：

第一行输入两个整数 n 和 k（$1 \le k \le n \le 10^9$）。

Output

每组数据输出一行一个整数，表示抽到好的排列的概率。答案对 $(10^9 + 7)$ 取模后输出。

可以证明答案是一个有理数 $\frac{P}{Q}$。您需要输出 $PQ^{-1} \bmod (10^9 + 7)$ 的值，其中 Q^{-1} 是满足 $QQ^{-1} \bmod (10^9 + 7) = 1$ 的整数。

Example

standard input	standard output
4 3 2 3 1 3 3 1 1	500000004 333333336 666666672 1

Note

对于第一组样例数据，排列 $\{1,2,3\}$，$\{2,3,1\}$ 和 $\{3,1,2\}$ 都是好的。因此答案是 $\frac{3}{3!} = \frac{1}{2}$。因为 $2 \times 500000004 \bmod (10^9+7) = 1$，所以输出 $1 \times 500000004 \bmod (10^9+7) = 500000004$。

Problem K. 路径规划 2

有一个 n 行 m 列的网格。网格里的每个格子都写着一个整数，其中第 i 行第 j 列的格子里写着整数 $a_{i,j}$。

令 (i,j) 表示位于第 i 行第 j 列的格子。您现在需要从 $(1,1)$ 出发并前往 (n,m)。当您位于格子 (i,j) 时，您可以选择走到右方的格子 $(i,j+1)$（若 $j<m$），也可以选择走到下方的格子 $(i+1,j)$（若 $i<n$）。

令 $\mathbb{S}$ 表示路径上每个格子里的整数形成的集合，包括 $a_{1,1}$ 和 $a_{n,m}$。令 $\operatorname{mex}(\mathbb{S})$ 表示不属于 $\mathbb{S}$ 的最小非负整数。请找出一条路径以最小化 $\operatorname{mex}(\mathbb{S})$，并求出这个最小的值。

Input

有多组测试数据。第一行输入一个整数 T（$1\le T\le 10^4$）表示测试数据组数。对于每组测试数据：

第一行输入两个整数 n 和 m（$1\le n,m\le 10^6$，$1\le n\times m\le 10^6$）表示网格的行数和列数。

对于接下来 n 行，第 i 行输入 m 个整数 $a_{i,1},a_{i,2},\cdots,a_{i,m}$（$0\le a_{i,j}\le 10^9$），其中 $a_{i,j}$ 表示格子 (i,j) 里的整数。

保证所有数据 $n\times m$ 之和不超过 10^6。

Output

每组数据输出一行一个整数，表示最小的 $\operatorname{mex}(\mathbb{S})$。

Example

standard input	standard output
2 2 3 2 0 1 0 3 4 1 5 100 0 2 0 1	1 3

Note

对于第一组样例数据，共有 3 条可能的路径。

- 第一条路径为 $(1,1)\to(1,2)\to(1,3)\to(2,3)$。$\mathbb{S}=\{2,0,1,4\}$ 因此 $\operatorname{mex}(\mathbb{S})=3$。
- 第二条路径为 $(1,1)\to(1,2)\to(2,2)\to(2,3)$。$\mathbb{S}=\{2,0,3,4\}$ 因此 $\operatorname{mex}(\mathbb{S})=1$。
- 第三条路径为 $(1,1)\to(2,1)\to(2,2)\to(2,3)$。$\mathbb{S}=\{2,0,3,4\}$ 因此 $\operatorname{mex}(\mathbb{S})=1$。

因此答案为 $\min(3,1,1)=1$。

对于第二组样例数据，只有 1 条可能的路径，即 $(1,1)\to(1,2)\to(1,3)\to(1,4)\to(1,5)$。$\mathbb{S}=\{100,0,2,1\}$ 因此 $\operatorname{mex}(\mathbb{S})=3$。

Problem L. 恒星

在天文学中，恒星分类是根据恒星的光谱特征对其进行的分类。大多数恒星目前根据摩根-肯纳（MK）系统进行分类，使用字母 O、B、A、F、G、K 和 M，从最热的（O 型）到最冷的（M 型）依次排列。每个字母类别下再使用数码进行细分，0 表示最热，9 表示最冷。例如，A8、A9、F0 和 F1 形成了从热到冷的序列。

显然，7 个字母和 10 个数码给我们提供了总共 70 种不同的类别，其中 O0 是最热的，M9 是最冷的。给定两个类别，判断第一个类别是比第二个类别更热，更冷，还是相同。

Input

有多组测试数据。第一行输入一个整数 T（$1 \le T \le 10^3$）表示测试数据组数，对于每组测试数据：

第一行输入两个长度为 2 的字符串，表示两个类别。

Output

每组数据输出一行。

- 若第一个类别比第二个类别更热，输出 `hotter`。
- 若第一个类别比第二个类别更冷，输出 `cooler`。
- 若第一个类别和第二个类别相同，输出 `same`。

Example

standard input	standard output
5 O2 O7 M9 M2 G2 G2 A0 B9 F8 K1	hotter cooler same cooler hotter

Problem M. 三角剖分

圆上等距离分布着 n 个点，从某个点开始按顺时针方向编号从 1 到 n。令 d 表示相邻两点之间的弧长。

堡堡在圆上画了 $(2n-3)$ 条弦，每条弦都连接了 n 个点中的两个。保证不存在两条弦连接同一对点，且不存在两条弦相交于圆的内部（不包括边界）。容易看出，这些弦构成了 $(n-2)$ 个三角形。这些三角形称为圆的一个三角剖分。

堡堡很喜欢这个三角剖分，所以他把这 $(n-2)$ 个三角形记在了一张纸上。第 i 个三角形用三个整数 $k_{i,1}, k_{i,2}, k_{i,3}$ 记录，其中 $k_{i,j} \times d$ 是第 i 个三角形的第 j 个顶点到下一个顶点之间的最短弧长。对于 $j=1$ 和 $j=2$，它的下一个顶点是第 $(j+1)$ 个顶点。对于 $j=3$，它的下一个顶点是第 1 个顶点。

几天后，当堡堡想要再次欣赏三角剖分时，他失望地发现，所有的弦都被他的室友梦格擦掉了，所以他只能根据纸上的信息还原三角剖分。您的任务是帮助堡堡确定 $v_{i,j}$（对于所有 $1 \le i \le n$ 以及 $1 \le j \le 3$），意思是第 i 个三角形的第 j 个顶点是圆上的第 $v_{i,j}$ 个点。堡堡也有可能记错了信息，这种情况下，您需要告诉堡堡这样的三角剖分并不存在。

Input

有多组测试数据。第一行输入一个整数 T（$1 \le T \le 10^4$）表示测试数据组数，对于每组测试数据：

第一行输入一个整数 n（$3 \le n \le 2 \times 10^5$），表示圆上点的数量。

对于接下来 $(n-2)$ 行，第 i 行输入三个整数 $k_{i,1}, k_{i,2}, k_{i,3}$（$1 \le k_{i,j} \le \lfloor \frac{n}{2} \rfloor$），其中 $k_{i,j} \times d$ 是第 i 个三角形的第 j 个顶点到下一个顶点之间的最短弧长。

保证所有数据 n 之和不超过 2×10^5。

Output

对于每组数据：

- 如果存在这样的三角剖分，首先输出一行 `Yes`。然后输出 $(n-2)$ 行，其中第 i 行包含三个由单个空格分隔的整数 $v_{i,1}, v_{i,2}, v_{i,3}$，意思是第 i 个三角形的第 j 个顶点是圆上的第 $v_{i,j}$ 个点。如果有多种合法答案，您可以输出任意一种。
- 如果不存在这样的三角剖分，只要输出一行 `No`。

Example

standard input	standard output
3 3 1 1 1 4 1 1 1 1 2 1 6 3 1 2 1 2 1 1 3 2 1 2 1	Yes 3 2 1 No Yes 2 5 4 1 2 6 6 5 2 3 2 4

Note

第一和第三组样例数据如下图所示。