

Problem A. 能量采集

Input file: standard input
Output file: standard output
Time limit: 2 seconds
Memory limit: 1024 megabytes

你正在玩一个游戏。游戏中一共有两种能量，分别表示为 A 和 B。

游戏地图可以看成是一张 n 行 m 列的地图，第 i 行第 j 列的格子所含有的能量是 A 或 B 的其中一种。你从 $(1, 1)$ 出发，走到 (n, m) 。你只能往下或往右走。

你有一个容量为 k 的能量容器。这个容器是一个队列，即当收集到的能量总量大于 k 时，越早收集到的能量会越先从队列弹出（即消失），直到能量总量等于 k 为止。

你所属的阵营是 A，即每当该容器内的能量全是种类 A 并且容器装满了的时候，你所属的阵营的分数会加一（之后容器内的能量不会消失）。当你站在任意格子上时，不论该格子内的能量是哪一种，你都会收集一单位的能量到容器内。

你想知道：对于所有 $\binom{n+m-2}{n-1}$ 条可能的路径，你期望能够给自己阵营加多少分？

Input

第一行三个正整数 n, m, k ($1 \leq n, m \leq 400, 1 \leq k < n + m$)，含义见题面。

接下来输入游戏地图：一共输入 n 行，每行输入一个长度为 m 的只含有 A, B 的字符串。

Output

输出一个整数，表示期望加的分数对 998244353 取模后的结果。

Example

standard input	standard output
2 3 2 AAA BBA	332748119

Note

一共有三条路径，分别收集到的能量顺序为：AAAA, AABA, ABBA。这三条路径给阵营加的分数分别为 3, 1, 0，故答案为 $\frac{3+1}{3} = \frac{4}{3}$ 。

Problem B. A Boring Game

Input file: **standard input**
Output file: **standard output**
Time limit: 4 seconds
Memory limit: 256 megabytes

GuapiStarOier hates junk advertisements.

One day GuapiStarOier sees a junk advertisement about a boring game. However, he comes up with a simple idea.

The game has a tower with n floors and each floor with an enemy. The strength of the enemy in i -th floor is a_i and the experience the player can get is b_i , and the player's goal is to defeat all enemies. If the player is on the y -th floor, he can move to $(y + 1)$ -th or $(y - 1)$ -th floor (if it exists).

Every time he moves to a new floor that he never visited before, he will fight against the enemy on this floor. Assume the strength of the player is x , the strength of the enemy is y and the experience is z , if $x \geq y$, the player defeats the enemy and his strength becomes $x + z$, otherwise the player loses the game.

There are q rounds, and in each round the player will initially be on a certain floor p with strength v (also he has to fight against the enemy on that floor initially). GuapiStarOier wonders about the maximal strength he can get during the game in each round. Since this problem is too easy for him, GuapiStarOier gives it to you and asks you to solve it.

Input

The first line contains a single integer T ($1 \leq T \leq 100$), denoting the number of test cases.

For each test case, the first line contains two integers n, q ($1 \leq n \leq 10^5$, $1 \leq q \leq 10^5$), indicating the number of floors and the number of queries.

Each of the following n lines contains two integers a_i, b_i ($0 \leq a_i, b_i \leq 10^{12}$), denoting the strength of the enemy and the experience the player can get.

Each of the following q lines contains two integers p, v ($1 \leq p \leq n, 0 \leq v \leq 10^{12}$), denoting the query in which the player is on the floor p with strength v .

It is guaranteed that the sum of n and q over all test cases won't exceed 10^6 .

Output

For each query, output an integer denoting the maximal strength the player can get.

Example

standard input	standard output
2	24
6 5	1
3 6	32
1 2	28
9 10	6
2 2	9
7 1	1
4 2	9
2 1	11
3 1	10
3 9	
4 5	
6 4	
6 5	
3 6	
1 2	
13 10	
2 2	
7 1	
4 3	
2 1	
3 1	
3 9	
4 5	
6 4	

Problem C. 往日重现

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

平面上有 n 个圆，保证这 n 个圆之间没有任何公共点。

reknilB 有 m 个问题，每个问题形如：「从 (x, y) 到 (p, q) 最少要穿过多少个圆的边界。」

你需要回答他的这些问题。

Input

第一行两个整数 n, m ($1 \leq n, m \leq 10^5$)。

接下来 n 行，每行三个整数 a_i, b_i, r_i ($-10^7 \leq a_i, b_i \leq 10^7, 1 \leq r_i \leq 10^7$)，表示一个圆心在 (a_i, b_i) ，半径为 r_i 的圆。保证这 n 个圆之间没有任何公共点。

接下来 m 行，每行四个整数 x, y, p, q ($-10^7 \leq x, y, p, q \leq 10^7$)，表示一次询问。保证点 (x, y) 和点 (p, q) 不在任何圆上。

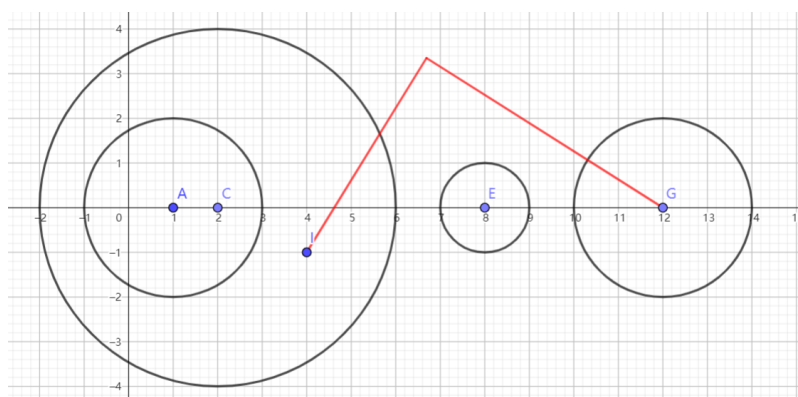
Output

输出 m 行，每行输出对应问题的回答。

Example

standard input	standard output
4 4	0
1 0 2	1
2 0 4	2
8 0 1	2
12 0 2	
1 0 2 0	
4 -1 2 0	
4 -1 8 0	
4 -1 12 0	

Note



对于第一个询问，如图中从 A 点到 C 点，不需要穿过任何圆的边界，因此输出 0；

对于第二个询问，如图中从 I 点到 C 点，需要穿过圆 A 的边界，因此输出 1；

对于第三个询问，如图中从 I 点到 E 点，需要穿过圆 C 和圆 E 的边界，因此输出 2；

对于第四个询问，如图中从 I 点到 G 点，需要穿过圆 C 和圆 G 的边界，因此输出 2。请注意，为了穿过最少的圆边界，移动路线可以不是一条直线。因此对于这组询问，我们只需要在移动中绕过圆 E 围成区域即可，一种可能的移动路径用红色线标出。

Problem D. 小美爱画鱼

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

小美十分热爱画画。

这天小美拿到了一张方格纸，方格纸可以看作在第一象限内无限大的一个格点图。小美发现，如果将一个格子的左上角和右下角相连起对角线，就可以得到一条美丽的鱼。

于是小美开始动手画起这样的线，她现在想画的线有 n 条，每条线都会给出起始的左上角的坐标和终止的右下角坐标，但小美也不想重复劳动，因此她想知道是否不存在一个格子，使得这个格子所拥有的对角线被经过多次，以及最后画完后所有被画上的线的长度，你只要输出这个数除以 $\sqrt{2}$ 的结果即可（可以证明这个数一定是整数）。

Input

第一行输入 T ($T \leq 10$) 表示数据组数。

对于每组数据，第一行输入一个整数 n ($1 \leq n \leq 10^5$)，表示需要画的线段数。

接下来 n 行每行 4 个整数 x_1, y_1, x_2, y_2 ($0 \leq x_1, y_1, x_2, y_2 \leq 10^5$) 表示每条线段的起点和终点，保证 $x_1 + y_1 = x_2 + y_2, x_1 < x_2$ 。

Output

输出共 $2T$ 行。

如果没有一个格子的对角线经过多次，则第一行输出 YES，否则输出 NO。

第二行输出一个整数，表示所有被画上的线的长度为这个数的 $\sqrt{2}$ 倍。

Example

standard input	standard output
3	YES
2	2
0 1 1 0	YES
3 2 4 1	3
3	NO
0 1 1 0	6
2 3 3 2	
3 2 4 1	
4	
0 1 1 0	
2 3 3 2	
3 2 4 1	
0 5 5 0	

Problem E. 小团来打字

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

小团每天都要将很多文件输入到电脑里面去，由于键盘长期的使用，导致键盘出现了一些 bug，如果连续敲击了某个键 k 次后，再次敲击该键位的话，却不会有任何的输入，这个 bug 可以通过两种方式解决，一种是在这个时候敲击另一个键位，或者是再敲击 k 次这个键位，从而使得这个 bug 被修复。

现在小团又收到了一份文件，文件里面有着一堆这样的连续的需要被键入的文字，但是由于小团使用的键盘是特制的字符集巨大的键盘，所以她不能使用删除键来进行操作，所以小团想知道，如果她需要输入接下来这个文件，那么她需要以怎样的方式来敲击键盘？但如果你把小团需要输出的字符丢在她面前，那也过于冗长了，因此小团只想知道对于每个键小团需要敲击多少次。

Input

第一行两个整数 n, k ($1 \leq n \leq 10^5, 1 \leq k \leq 10^9$)，表示文件的连续段数目和产生 bug 的临界值。

接下来 n 行每行两个正整数 a_i, b_i ($1 \leq a_i, b_i \leq 10^9$)，表示 a_i 这个字符将会紧接着输入 b_i 次。

Output

第一行输出一个整数 m ，表示小团需要敲击的键位数。

接下来 m 行每行两个正整数 c_i, d_i ，表示 c_i 这个字符键被敲击了 d_i 次。请按 c_i 从小到大的顺序输出。

Example

standard input	standard output
5 4	2
1 6	1 27
2 1	2 21
1 9	
2 2	
2 10	

Problem F. 炸弹鸭

Input file: standard input
Output file: standard output
Time limit: 4 seconds
Memory limit: 256 megabytes

众所周知，炸弹鸭是鹅鸭杀中最为有游戏气氛的角色之一。

假设现在有一局鹅鸭杀游戏，游戏可以抽象的看成在一个 n 个点 m 条边的无向图上进行。每条边有边权 w 代表两点之间的距离。每个人恰好占据一个节点。

但与原游戏不同，每个人在接到炸弹后只能丢给与自己直接有边相连的并且距离小于等于给定的限制 k 的人，但炸弹鸭一次可能会塞给你很多个炸弹，且每次你只能把一个炸弹给一个满足条件且此次你还未传递炸弹的人，因此你需要把自己所有的炸弹按照上述条件传递出去，我们定义爆炸被避免当且仅当你手上的炸弹全被传递出去并且存在一个与你有关边相连并且距离小于等于给定限制 k 的人没有被你传递炸弹。

每次游戏炸弹鸭给人炸弹的概率是等可能的，因此你想知道，如果知道炸弹鸭每次给人的炸弹数以及给定限制 k ，有多少位置的人不能够完成传递来避免爆炸。

Input

第一行三个整数 n, m, q ($1 \leq n \leq 2 \times 10^5, 1 \leq m, q \leq 10^6$)，表示人数，边数和询问次数。

接下来 m 行，每行三个整数 u, v, w ($1 \leq u, v \leq n, u \neq v, 1 \leq w \leq 10^9$) 分别表示在 u 和 v 之间有一条边，这条边长度为 w ，保证数据无重边以及自环。

接下来 q 行，每行两个整数 d, k ($1 \leq d \leq n, 1 \leq k \leq 10^9$)，表示炸弹数和限制长度。

Output

输出共 q 行。每行输出一个数字表示答案。

Example

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

Note

对于第一组询问，如果炸弹鸭给 1, 3, 4 号人炸弹均无法避免爆炸，如果给 2 号人炸弹，他可以传给 1 号人，此时有 3 个人不能避免爆炸。

对于第二组询问，有且仅有 4 号人无法避免爆炸。

Problem G. Guard the Kingdom

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

Kanade becomes the king of the Kingdom Whatever. There're n cities in the kingdom, which are connected by $n - 1$ biconnected roads. In other words, the cities and roads construct a tree structure. There're k important cities that need to be protected by arms. It is known that each troop protects all important cities that are closest to it, and only them. If there are multiple important cities with the minimum distance, all of them are protected. The distance between a city and a troop is equal to the number of roads on the unique path between the city and the city containing the troop. Additionally, the troop can be placed in an important city. Of course, the troop protects only that important city in that case.

There're m troops that are placed in some cities now. In order to make the plan of kingdom defense for next year, Kanade would like to know how many important cities that each troop protects, and how many troops protect each important city.

Input

The first line contains three integers n, m, k ($2 \leq n \leq 2 \cdot 10^5, 1 \leq m, k \leq n$), indicating the number of cities, the number of troops, and the number of important cities, respectively. Cities are numbered from 1 to n .

The second line contains $n - 1$ integers p_2, p_3, \dots, p_n ($1 \leq p_i \leq n, p_i \neq i$). The i -th integer indicates there is a biconnected road between city $i + 1$ and p_{i+1} .

The third line contains m integers t_i ($1 \leq t_i \leq n$). The i -th integer indicates the i -th troop is placed in city t_i . It is guaranteed that for all $1 \leq i, j \leq n$ and $i \neq j$, $t_i \neq t_j$ meets.

The fourth line contains k integers c_i ($1 \leq c_i \leq n$). The i -th integer indicates c_i is an important city. It is guaranteed that for all $1 \leq i, j \leq n$ and $i \neq j$, $c_i \neq c_j$ meets.

Output

Output two lines. The first line contains m integers. The i -th integer indicates the number of important cities protected by the troop t_i . The second line contains k integers. The i -th integer indicates the number of troops protecting the important city c_i .

Example

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

Note

City 5 is protected by troop 2, 3, and City 6 is only protected by troop 3.

Problem H. 约瑟夫问题

Input file: standard input
Output file: standard output
Time limit: 2.5 seconds
Memory limit: 256 megabytes

传统的约瑟夫问题为: n 个人围成一圈, 顺时针依次编号为 $1, 2, \dots, n$ 。从编号为 1 的人开始顺时针从 1 开始报数, 每报到 k 时这个人就出局, 之后他的下一个人又从 1 开始报数, 最后询问剩下的那个人的编号。

而在本题中, 每一轮的 k 都不完全相同。游戏一共进行 q 轮, 你需要输出每一轮出局的人的编号。

Input

第一行两个正整数 n, q ($1 \leq q < n \leq 5 \times 10^5$);

接下来 q 行, 每行输入一个正整数 k ($2 \leq k \leq 10^9$), 表示这一轮报到 k 的人出局。下一轮由出局的顺时针方向的下一个人开始报数。

Output

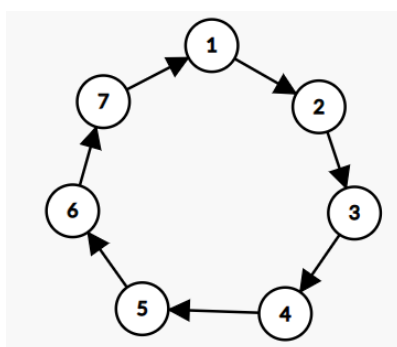
输出 q 行, 第 i 行一个正整数, 表示该轮出局的人的编号。

Example

standard input	standard output
7 3	5
5	1
9	4
8	

Note

样例如图所示, 箭头表示报数顺序:



第一轮, 从编号为 1 的人开始报数, 报到 5 时编号为 5 的人出局, 下一轮由编号为 6 的人开始报数;
第二轮, 从编号为 6 的人开始报数, 报到 9 时编号为 1 的人出局, 下一轮由编号为 2 的人开始报数;
第三轮, 从编号为 2 的人开始报数, 报到 8 时编号为 4 的人出局。

Problem I. 公主连结! Re:Dive

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

Kanade 最近在打公主连结的活动。众所周知，活动中 Normal 难度的关卡有 15 个，1 – 4 关卡通关一次要耗费 8 体力，5 – 10 关卡通关一次要耗费 9 体力，11 – 15 关卡通关一次要耗费 10 体力。每个玩家一天可以通关无限次关卡。

现在 Kanade 有 n 体力，她想要把这些体力全部用在刷 Normal 关卡上，不剩余任何体力。请输出有多少种满足条件的方案。两种方案不同，当且仅当通关消耗某相同体力关卡的次数不同。

Input

第一行一个整数 T ($1 \leq T \leq 10^4$)，表示数据组数。

接下来 T 行，每行一个整数 n ($1 \leq n \leq 10^{18}$)，表示当前体力值。

Output

对于每组数据输出一行，表示满足条件的方案数。由于这个方案数可能会很大，请输出它对 $10^9 + 7$ 取模后的值。

Example

standard input	standard output
2	1
10	2
18	

Note

对于第二组数据，可以通关两次消耗 9 体力的关卡，也可以通关一次消耗 8 体力的关卡再通关一次消耗 10 体力的关卡。

Problem J. 数矩形

Input file: **standard input**
Output file: **standard output**
Time limit: 1 second
Memory limit: 512 megabytes

平面上有 n 个点，这 n 个点两两不重合。问这 n 个点可以组成多少个矩形。

请注意：矩形的边不必平行于坐标轴。

Input

第一行一个整数 n ($4 \leq n \leq 1000$)，表示点个数。

接下来 n 行，每行两个整数 x, y ($0 \leq x, y \leq 10^9$)，表示一个点。保证这些点两两不重合。

Output

输出一行一个整数，表示组成的矩形个数。

Example

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

Problem K. 打地鼠

Input file: standard input
Output file: standard output
Time limit: 0.5 seconds
Memory limit: 256 megabytes

Kanade 正在玩打地鼠游戏。地鼠生活在一个由 n 个洞和 m 条连接洞的双向通道构成的地下。这 m 条通道可以让任意两个洞口互相可达。

第一轮时，地鼠将从这 n 个洞中的某一个探出头。如果 Kanade 能打中地鼠，那么她就会在这次游戏中获胜。否则，地鼠会通过某条与这个洞相连的通道，爬到与这个洞相邻的另一个洞中。注意，这时地鼠必须移动。接着重复这个过程。

Kanade 反应速度很慢，因此很难反应过来地鼠在哪儿出现。但是她手速很快，所以她确定了一个击打序列 h_1, h_2, \dots, h_k ，在第 i ($1 \leq i \leq k$) 轮中，她只会击打 h_i 洞口，因为她手速很快，所以如果这一轮地鼠在 h_i 处出现，她一定会打到。然而地鼠不知道这个击打序列。

Kanade 同时知道了地下这 n 个洞口和 m 条通道的构造，Kanade 想知道她设计的击打序列是否能让她在游戏中获胜。也就是说，无论地鼠如何在地下移动，Kanade 都至少可以打中一次地鼠。如果一定可以获胜的话，Kanade 还想知道在最晚第几轮时获胜。

Input

第一行三个整数 n, m, k ($1 \leq n \leq 10^3, n-1 \leq m \leq \frac{n(n-1)}{2}, 1 \leq k \leq 5 \times 10^3$)，分别表示洞口数量，通道数量和击打序列的长度。洞口按 1 到 n 编号。

接下来 m 行，每行两个整数 u_i, v_i ($1 \leq u_i, v_i \leq n, u_i \neq v_i$)，表示洞口 u_i 与 v_i 间有一条通道。保证同一条通道在数据中只出现一次，保证这 m 条通道可以让任意两个洞口互相可达。

最后一行 k 个整数 h_1, h_2, \dots, h_k ，表示击打序列。

Output

输出一行，如果 Kanade 设计的击打序列一定能让她在游戏中获胜，输出最晚一定能获胜的轮次，否则输出 Lose。

Examples

standard input	standard output
4 3 4 1 2 1 3 1 4 1 1 2 3	2
4 3 4 1 2 1 3 1 4 3 4 1 2	Lose

Note

对于第一组样例，如果地鼠第一轮在洞口 1 出现，那么 Kanade 会打中，否则，地鼠在第二轮一定在 1 出现，Kanade 也会打中。所以 Kanade 一定会获胜，并且最晚在第二轮中获胜。

对于第二组样例，一种 Kanade 不会获胜的地鼠出现顺序为 $\{2, 1, 2, 1\}$ 。

Problem L. 树边重排

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

给一棵树，要求用一个序列 p_1, p_2, \dots, p_{n-1} 表示这棵树，其中 p_i 表示 i 与 p_i 有一条边。

Input

第一行一个整数 n ($2 \leq n \leq 10^5$)。

接下来 $n - 1$ 行，每行两个整数 u, v ($1 \leq u, v \leq n, u \neq v$)，表示树上的一条边。保证输入描述了一棵合法的树。

Output

输出一行 $n - 1$ 个整数 p_1, p_2, \dots, p_{n-1} ，其中 p_i 表示 i 与 p_i 有一条边。

Example

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

Problem M. 操作系统计算题

Input file: standard input
Output file: standard output
Time limit: 3 seconds
Memory limit: 256 megabytes

这天，rockdu 正在复习操作系统，但是被一个叫做 HRRN 的算法难住了。

HRRN（最高响应比优先调度）算法是操作系统中的一个经典算法。其背景如下：给出许多进程，第 i 个进程有一个出现时刻 x_i 和一个持续时间 s_i ，表示该进程从 x_i 时刻才出现（进程出现之后才可以被选择），并且需要分配 s_i 个单位的执行时间。若当前时刻为 t ，定义一个进程在当前时刻的响应比为 $1 + \frac{t-x_i}{s_i}$ ，HRRN 算法的关键步骤就是要选择已经出现的进程（即 $x_i \leq t$ 的进程）中响应比最大的进程。

现在给定 m 个询问 t_j ，为了使问题变得更简单，只需要对每个询问都求出该时刻能选择出的进程的最大响应比。如果直到 t_j 时刻还没有进程出现，需要回答 -1 。rockdu 的操作系统学的很差，所以他只好找到你来出主意，聪明的你可以解决这个问题吗？

简化题意：给你 n 个二元组 (x_i, s_i) ，并给定 m 次查询，每次查询一个数 t_j ，询问所有满足 $x_i \leq t_j$ 的元组中 $1 + \frac{t-x_i}{s_i}$ 的最大值。

Input

第一行一个正整数 n ($1 \leq n \leq 10^6$)，表示共有 n 个进程。

接下来 n 行每行两个整数 x_i, s_i ($0 \leq x_i \leq 10^6, 1 \leq s_i \leq 10^6$)，两数用空格隔开，依次表示第 i 个进程的出现时间和持续时间。

接下来一行一个正整数 m ($1 \leq m \leq 10^6$)，表示共有 m 个询问。

接下来 m 行每行一个整数 t_j ($0 \leq t_j \leq 10^6$)，表示第 j 个询问为 t_j 。

Output

输出共 m 行。

若 t_j 时刻及以前已有进程出现，则第 j 行一个实数，表示在 t_j 时刻，已经出现的进程中响应比最大的进程的响应比；

若 t_j 时刻前没有进程出现，则该行输出 -1 。

你的输出与答案的绝对误差或相对误差在 10^{-6} 以内都被视为正确。

Example

standard input	standard output
5	1.222222
2 9	1.000000
6 2	3.000000
8 2	1.000000
4 9	1.000000
3 9	
5	
4	
2	
10	
2	
2	

Problem N. 本质不同的 01 环计数

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

求有多少个本质不同长度为 n 的由 0 和 1 构成的环，满足环上任意一段长度为 k 的子段和均相等。

Input

第一行一个整数 T ($1 \leq T \leq 10^5$)，为数据组数。

接下来 T 行每行两个整数 n, k ($2 \leq n \leq 10^6, 1 \leq k < n$)，意义如题目描述。

Output

输出 T 行，对于每组数据输出一行一个整数，表示本质不同且满足条件的 01 环个数，因为答案过大，只要求输出答案模 998244353 后的结果。

Example

standard input	standard output
4	2
2 1	3
4 2	3
10 6	108
100 10	

Note

循环同构的环在这里被认为是本质相同的，如 01 与 10，00110 与 01100 都被认为是同构的，只能计算一次贡献。

Problem O. 爬塔

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

LL 最近在尝试各种各样的游戏。

某天 LL 在广告里看到了这样一个简单的游戏：有一个高度为 n 层的塔，其中第 1 到 n 层分别有一个敌人，等级分别为 a_1, \dots, a_n ，玩家出生在第 0 层，初始等级为 1，从低往高走，每到一层会与当层敌人决斗，如果玩家等级 x 大于等于敌人的等级 k ，那么将打败敌人，自己的等级将增加至 $x + k$ ，否则游戏失败。当玩家打败第 n 层的敌人后，游戏通关。

此外，玩家随时可以选择付出一单位代价，等级增加一级，并返回到第 0 层，复活所有被击败的敌人（敌人等级不变），因为这是一个很简单的游戏，所以 LL 不希望自己失败，并且希望在付出的代价尽可能少的情况下通关游戏，现在她想知道自己最少需要付出多少代价。

Input

第一行一个正整数 n ($1 \leq n \leq 10^5$)，表示塔的高度。

接下来一行每行 n 个正整数，分别表示 a_1, a_2, \dots, a_n ($0 \leq a_i \leq 10^9$)。

Output

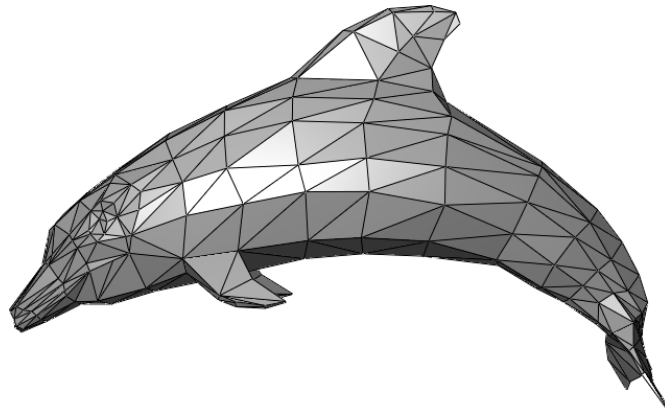
输出一个整数表示 LL 所需要付出的最小代价。

Example

standard input	standard output
6 2 0 5 9 22 30	3

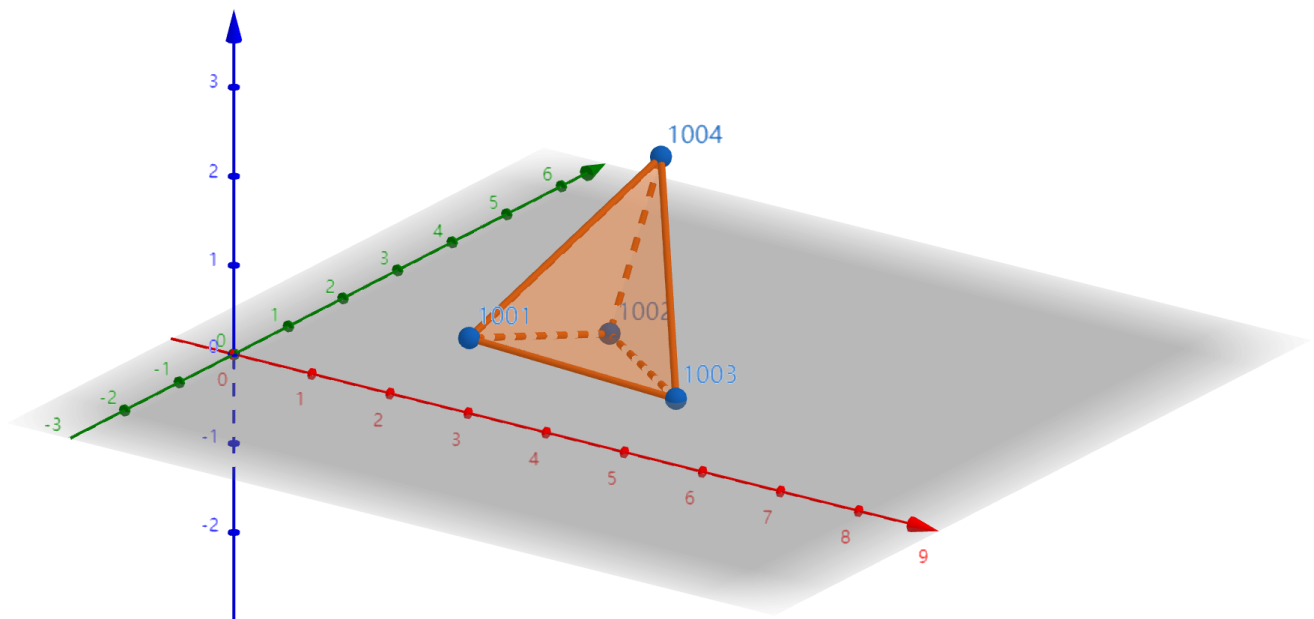
Problem P. 三维模型

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



四汪是一名游戏开发者，他在处理三维模型时遇到了一些问题，于是来找作为算法大佬的你求助。

在游戏引擎中，三维模型通常是由三角形组成的，这种表示方法被称为三角形网格（triangle mesh）。三角形网格之所以被广泛使用，是因为它具有简单性、稳定性、易于渲染、易于建模、易于处理等特点。三角形网格通常通过顶点数据和索引数据两部分来存储，其中顶点数据包含了构成三维模型的所有顶点的信息，包括顶点的索引值以及空间坐标；而索引数据用于描述三角形，每个三角形由三个整数表示其三个顶点在顶点数据中的索引。例如，对于下图中的三维模型，其顶点数据和索引数据分别如下。



顶点数据:

索引	坐标
1001	(2.5, 0.74, 0.5)
1002	(2.88, 2.76, 0.0)
1003	(4.55, 1.6, 0.0)
1004	(2.6, 4.1, 1.5)

索引数据:

- [1001, 1002, 1003]
- [1001, 1002, 1004]
- [1001, 1003, 1004]
- [1002, 1003, 1004]

我们认为两个三角形是相连的，当且仅当这两个三角形有一条公共边，即这两个三角形的索引数据中有两个点是相同的。只有当两个三角形相连时，我们认为这两个三角形属于同一个三维模型。如果不满足上述条件，即使这两个三角形在空间中是相交的，我们也不认为他们属于同一个三维模型。注意，同一条边可以被不止两个三角形共用，并且三维模型不一定必须是一个闭合的多面体。

从上述定义中不难看出，判断两个三角形是否属于同一个三维模型只与其索引数据有关，与其顶点数据并没有关系。现给出 n 个三角形的索引数据，请求出一共有多少个不同的三维模型，并给出每个模型包含的三角形个数。

Input

第一行一个整数 T ($1 \leq T \leq 10^5$)，表示测试数据组数。

每组测试数据第一行一个整数 n ($1 \leq n \leq 10^5$)，表示三角形的个数。

接下来 n 行，每行三个空格隔开的不同的整数，表示三角形三个顶点的索引值，每个数在 $[0, 10^9]$ 的范围内。数据保证不会出现完全相同的三角形。

所有测试数据的 n 总和 $\sum n \leq 10^5$ 。

Output

对于每组测试数据，第一行输出一共有多少个不同的三维模型，第二行按从小到大的顺序输出每个模型包含的三角形个数。

Example

standard input	standard output
1 7 1001 1002 1003 1001 1002 1004 1001 1003 1004 1002 1003 1004 1006 1005 1001 1006 1005 1002 1006 1005 1003	2 3 4

Problem Q. Du Cuo Ti Le

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

You are given an array of $2n$ numbers a_1, a_2, \dots, a_{2n} , which values are 0 initially.

You are given n operations. The i -th operation has two parameters $l_i, r_i (l_i < r_i)$, which means it sets $a_{l_i}, a_{l_i+1}, \dots, a_{r_i-1}$ to i . And you can perform the operations in any order. It is guranteed that $l_1, \dots, l_n, r_1, \dots, r_n$ forms a permutation of $[1, 2n]$.

You are asked to find the minimal and maximal value of i such that $a_i \neq a_{i+1}$.

Note that the maximal and minimal values can appear in different performance orders.

Input

The first line contains one integer n ($2 \leq n \leq 10^5$).

The following n lines each contains two integers l_i, r_i ($1 \leq l_i < r_i \leq 2n$).

Output

Output two integers in one line denoting the minimal and the maximal value of i .

Example

standard input	standard output
2 1 3 2 4	1 3

Problem R. 明信片

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

小 F、小 R、小 I、小 S、小 K 和小 Y 共 6 人在旅游季过后互相寄了 6 种不同系列的明信片。一共有 6 个信箱编号为 1, 2, ..., 6, 每个人拥有一个信箱。每个人只拥有一个信箱且只会寄一种系列的明信片, 明信片不能被寄给自己。

现已知:

1. 六人中有且仅有三人实现了三人间的两两互寄, 他们分别是 6 号信箱的主人、寄出「柯南」系列明信片的人和 Y;
2. K 一共收到三张明信片, 也寄出去三张明信片。这三个收信人同时也收到了「新年」系列明信片;
3. R 一共寄出三张明信片, 分别给了 1 号信箱的主人、寄出「巴黎」系列明信片的人和 S;
4. 寄出「治愈」系列明信片的人收到了所有其他系列的明信片;
5. 3 号信箱共收到三张明信片, 分别是「柯南」、「校园」、「巴黎」;
6. 收到「新年」系列明信片的共三人, 分别是 4 号信箱的主人、寄出「校园」系列明信片的人和 Y;
7. S 和 5 号信箱的主人都收到了 F 的明信片和「校园」系列明信片;
8. R 一共收到四张明信片, 其中包括「节气」系列明信片和 5 号信箱主人发来的明信片;
9. I 一共收到三张明信片, 其中包括 2 号信箱主人发来的「校园」系列明信片。I 寄出的明信片中, 有两张是给 3 号和 6 号信箱的; 但是 3 号信箱主人和 6 号信箱主人没寄明信片给 I, 且他们之间也没有互寄;
10. 如果把 S 和 4 号信箱主人收到的明信片放在一起, 则刚好集齐 6 种不同主题的明信片。

并且符合上述关系的寄送情况只有一种。请根据这些信息推断出谁拥有几号信箱, 寄出什么系列的明信片。

Input

无输入。

Output

输出 6 行, 每行包括一个大写字母、两个 1 到 6 之间的数字, 分别代表人名、明信片的系列与信箱编号。其中人名请以 F、R、I、S、K、Y 的顺序输出, 明信片的系列与数字的对应关系为「柯南」对应 1、「新年」对应 2、「治愈」对应 3、「巴黎」对应 4、「校园」对应 5、「节气」对应 6。

如「F12」表示 F 发的明信片系列是「柯南」, 使用 2 号信箱。

Example

standard input	standard output
(no input)	F11 R22 I33 S44 K55 Y66

Note

样例的输出是错误答案，仅仅表示输出的格式。