

2026 山东省大学生程序设计竞赛

正式赛

2026 年 5 月 24 日



试题列表

A	华容道
B	字典 2
C	会议日程
D	最大数码 2
E	简单的构造题
F	礼物归位
G	吸血鬼爬行者
H	拼图
I	版本号
J	插松枝
K	最小生成树
L	分数迭代
M	博物馆奇妙夜 2

本试题册共 13 题，18 页。
如果您的试题册缺少页面，立即通知志愿者。

由 SUA 程序设计竞赛命题组命题。
<https://sua.ac/>

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. 华容道

有一个 n 行 m 列的网格。每个格子中有一个箭头，指向上 (U)，下 (D)，左 (L) 或右 (R)。

每次操作您可以移除恰好一个箭头。然而，要移除一个箭头，必须确保其指向的方向上没有其他箭头。更具体地说，设 $s_{i,j}$ 是位于第 i 行第 j 列的箭头。要移除 $s_{i,j}$ ：

- 若 $s_{i,j} = \text{U}$ ，则之前必须移除所有满足 $1 \leq k < i$ 的 $s_{k,j}$ 。
- 若 $s_{i,j} = \text{D}$ ，则之前必须移除所有满足 $i < k \leq n$ 的 $s_{k,j}$ 。
- 若 $s_{i,j} = \text{L}$ ，则之前必须移除所有满足 $1 \leq k < j$ 的 $s_{i,k}$ 。
- 若 $s_{i,j} = \text{R}$ ，则之前必须移除所有满足 $j < k \leq m$ 的 $s_{i,k}$ 。

对于每个箭头，分别回答以下问题：移除这个箭头所需的最少操作次数是多少？

Input

有多组测试数据。第一行输入一个整数 T ($1 \leq T \leq 10^3$)，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 m ($1 \leq n, m \leq 5 \times 10^4$, $n \times m \leq 5 \times 10^4$)，表示网格的行数和列数。

对于接下来的 n 行，第 i 行输入一个长度为 m 的字符串 $s_{i,1}s_{i,2}\dots s_{i,m}$ ($s_{i,j} \in \{\text{U}, \text{D}, \text{L}, \text{R}\}$)，其中 $s_{i,j}$ 表示第 i 行第 j 列的箭头。

保证所有测试数据中 $n \times m$ 的总和不超过 5×10^4 。

Output

每组测试数据输出 n 行。第 i 行包含 m 个由单个空格分隔的整数 $a_{i,1}, a_{i,2}, \dots, a_{i,m}$ ，其中 $a_{i,j}$ 是移除箭头 $s_{i,j}$ 所需的最少操作次数。如果无法移除箭头 $s_{i,j}$ ，则 $a_{i,j}$ 应为 -1 。

Example

standard input	standard output
2	-1 -1 -1
4 3	7 -1 -1
RRD	5 3 2
DUL	3 2 1
RDD	1 1
RRR	
1 2	
LR	

Problem B. 字典 2

给定一棵有 n 个节点的有根树，根节点为 1。每个节点 u 关联一个小写英文字母 c_u 。对于任意节点 u ，定义 $S(u)$ 为从 u 开始到根的路径上经过的字符依次拼接形成的字符串。更正式地：

- 若 $u = 1$ ，则 $S(u) = c_u$ 。
- 若 v 是 u 的子节点，则 $S(v) = c_v + S(u)$ ，其中 $+$ 表示字符串连接。

令 \mathcal{V} 为树中所有节点形成的集合。令 \mathcal{D} 为集合 $\{S(u) \mid u \in \mathcal{V}\}$ 中所有字符串的所有非空子串形成的集合。也就是说， $w \in \mathcal{D}$ 当且仅当存在某个节点 u ，使得 w 是 $S(u)$ 的非空子串。

我们定义字典序 \leq_{lex} 如下：对于两个字符串 A 和 B ， $A \leq_{lex} B$ 当且仅当 A 是 B 的前缀，或者在它们第一个不同的字符位置上， A 的字符小于 B 的对应字符。

您需要依次处理 q 个操作。对于每个操作，给定一个加密整数 u' 和一个字符 c 。令 N 为该操作执行前的节点总数， l 为上一个操作的输出结果（对于第一个操作， $l = 0$ ）。请执行以下步骤：

1. 计算真实的父节点编号 $u = ((u' + l) \bmod N) + 1$ 。创建一个编号为 $(N + 1)$ 的新节点作为 u 的子节点，其字符为 c ，从而形成新的字符串 $S(N + 1) = c + S(u)$ 。更新集合 \mathcal{D} ，将 $S(N + 1)$ 的所有非空子串中尚未存在于 \mathcal{D} 的部分加入该集合。
2. 输出更新后的集合 \mathcal{D} 中字典序小于或等于 $S(N + 1)$ 的字符串数量。更正式地，输出以下值：

$$|\{w \in \mathcal{D} \mid w \leq_{lex} S(N + 1)\}|$$

Input

每个测试文件中只有一组测试数据。

第一行输入两个整数 n 和 q ($1 \leq n, q \leq 10^5$)。

第二行输入一个由小写英文字母构成的字符串 $c_1 c_2 \dots c_n$ ，其中 c_i 是节点 i 的字符。

对于接下来 $(n - 1)$ 行，第 i 行输入两个整数 u_i 和 v_i ($1 \leq u_i, v_i \leq n$)，表示一条连接节点 u_i 和节点 v_i 之间的边。

对于接下来 q 行，第 i 行输入一个整数 u'_i ($1 \leq u'_i \leq N$ ，其中 N 是该操作执行前的节点总数) 和一个小写英文字母 c_i ，表示第 i 次操作。

Output

每次操作输出一行，包含一个整数，表示答案。

Example

standard input	standard output
2 3	5
ab	2
1 2	8
1 b	
1 a	
2 c	

Note

以下对样例数据进行解释。

初始时, $S(1) = a$, $S(2) = ba$, 集合 $\mathcal{D} = \{a, b, ba\}$ 。

1. 输入 $u' = 1, c = b$ 。解密得到父节点 $u = 2$ 。新建节点 3, $S(3) = bba$ 。新增子串 bb, bba 。 \mathcal{D} 中 $\leq_{\text{lex}} bba$ 的字符串共 5 个。
2. 输入 $u' = 1, c = a$ 。解密得到父节点 $u = 1$ 。新建节点 4, $S(4) = aa$ 。新增子串 aa 。 \mathcal{D} 中 $\leq_{\text{lex}} aa$ 的字符串共 2 个。
3. 输入 $u' = 2, c = c$ 。解密得到父节点 $u = 1$ 。新建节点 5, $S(5) = ca$ 。新增子串 c, ca 。 \mathcal{D} 中 $\leq_{\text{lex}} ca$ 的字符串共 8 个。

Problem C. 会议日程

一家公司有 n 名员工，每名员工 i 有一个空闲时间窗口 $[l_i, r_i]$ （两个端点均为整数），在此期间他们可以参加会议。

公司正在组织团队建设活动，每对员工都需要进行一对一的交谈。为了让两名员工见面，他们的空闲时间窗口必须相交（至少有一个公共点）。

员工可以通过改变空闲时间窗口的端点来调整他们的日程。然而，重新安排日程很麻烦，将一个端点从时间 a 改为时间 b 会导致 $(a - b)^2$ 单位的烦恼值。较大的调整需要移动更多的日程，因此烦恼值呈平方增长。调整后，每个时间窗口仍需合法（左端点小于等于右端点，两个端点仍然均为整数）。

您需要求出最小总烦恼值，使得每对员工都能找到见面时间。

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：

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

对于接下来的 n 行，第 i 行输入两个整数 l_i 和 r_i ($0 \leq l_i \leq r_i \leq 10^6$)，表示第 i 名员工的空闲时间窗口。

保证所有测试数据中 n 的总和不超过 2×10^5 。

Output

每组测试数据输出一行，包含一个整数，表示最小总烦恼值。

Example

standard input	standard output
3	8
3	0
6 8	4901
0 2	
1 7	
2	
1 3	
2 4	
2	
1 1	
100 100	

Note

对于第一组样例数据，员工 1 可以将窗口从 $[6, 8]$ 调整到 $[4, 8]$ ，烦恼值为 $(6 - 4)^2 = 4$ ，员工 2 可以将窗口从 $[0, 2]$ 调整到 $[0, 4]$ ，烦恼值为 $(4 - 2)^2 = 4$ 。调整后，三个窗口 $[4, 8]$ 、 $[0, 4]$ 、 $[1, 7]$ 两两相交，因此每对员工都可以见面。总烦恼值为 $4 + 4 = 8$ 。

对于第二组样例数据，两名员工的窗口已经相交，因此不需要调整。

Problem D. 最大数码 2

令 $f(x)$ 为正整数 x 在十进制表示下的最大数码。例如, $f(4523) = 5$ 以及 $f(1001) = 1$ 。

给定四个正整数 l_a, r_a, l_b 和 r_b 满足 $l_a \leq r_a$ 且 $l_b \leq r_b$, 您需要计算

$$\sum_{a=l_a}^{r_a} \sum_{b=l_b}^{r_b} f(a+b)$$

也就是说, 对所有满足 $l_a \leq a \leq r_a$ 且 $l_b \leq b \leq r_b$ 的 (a, b) 求 $f(a+b)$ 的总和。

Input

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

第一行输入四个整数 l_a, r_a, l_b 和 r_b ($1 \leq l_a \leq r_a \leq 10^9, 1 \leq l_b \leq r_b \leq 10^9$)。

Output

每组测试数据输出一行, 包含一个整数, 表示答案。

Examples

standard input	standard output
2 178 182 83 85 2 5 3 6	91 100
1 1 1000000000 1 1000000000	8425695016000000001

Problem E. 简单的构造题

给定一个 n 行 m 列的网格。您需要在每个格子中填入从 0 到 $(nm - 1)$ 的整数（每个整数恰好使用一次），使得对于所有 $0 \leq i < nm$ ，填有 i 的格子和填有 $(i + 1) \bmod nm$ 的格子不相邻（即不共享一条边）。

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：第一行输入两个整数 n 和 m ($1 \leq n, m \leq 2 \times 10^5$, $3 \leq n \times m \leq 2 \times 10^5$)，表示网格的行数和列数。保证所有测试数据中 $n \times m$ 的总和不超过 2×10^5 。

Output

对于每组测试数据：

- 如果存在合法方案，首先输出一行 **Yes**。接下来输出 n 行。第 i 行包含 m 个由单个空格分隔的整数，其中第 j 个整数表示填在第 i 行第 j 列格子中的数。如果有多种合法方案，输出任意一种。
- 否则，如果不存在合法方案，只要输出一行 **No**。

Example

standard input	standard output
2	Yes
3 4	4 9 7 1
1 3	10 0 5 3
	6 2 11 8
	No

Problem F. 礼物归位

n 个人围坐在一张圆桌旁，按顺时针方向依次编号为 1 到 n 。对每个 $i = 1, 2, \dots, n$ ，第 i 个人与第 $((i \bmod n) + 1)$ 个人相邻。

现有 n 份礼物，依次编号为 1 到 n 。初始时，对每个 $i = 1, 2, \dots, n$ ，第 i 个人拿到第 p_i 份礼物，但忙中出错，导致至少有一人拿错了礼物。在正确的分发方案下，对每个 $i = 1, 2, \dots, n$ ，第 i 个人应该拿到第 q_i 份礼物。

现在您可以进行若干次操作，每次操作可以交换任意两位相邻者手中的礼物，请用最少的操作次数使得所有人都拿到正确的礼物。

Input

有多组测试数据。第一行输入一个整数 T ($1 \leq T \leq 10^3$)，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 ($3 \leq n \leq 3 \times 10^3$)。

第二行输入 n 个互不相同的整数 p_1, p_2, \dots, p_n ($1 \leq p_i \leq n$)。

第三行输入 n 个互不相同的整数 q_1, q_2, \dots, q_n ($1 \leq q_i \leq n$)。

保证所有测试数据中 n 的总和不超过 3×10^3 。

Output

对于每组测试数据，输出两行：

第一行包含一个整数 k ，表示最少的操作次数。

第二行包含 k 个整数 x_1, x_2, \dots, x_k ，其中第 i 个整数 x_i 表示在第 i 次操作中，交换第 x_i 个人与第 $((x_i \bmod n) + 1)$ 个人手中的礼物。

如果有多种合法方案，您可以输出任意一种。

Example

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

Problem G. 吸血鬼爬行者

最近，由 poncle 开发的 roguelike 卡组构筑游戏《吸血鬼爬行者》已经正式发布。在游戏中，玩家会构筑牌组，并在战斗中打出卡牌来造成伤害。游戏的核心机制是连击系统：当您按照法力值消耗递增的顺序打出卡牌时，伤害倍率会不断叠加。



您当前有 n 张卡牌，第 i 张卡牌有一个消耗值和一个基础伤害。您需要选择一种顺序，将这 n 张卡牌全部打出。打出一张卡牌时：

- 如果这不是打出的第一张卡牌，且其消耗值等于上一张卡牌的消耗值加 1，那么伤害倍率增加 1。
- 否则，伤害倍率重置为 1。

在更新伤害倍率后，这张卡牌造成的伤害等于其基础伤害乘以倍率。

您要求出能够造成的最大总伤害。

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：

第一行输入一个整数 n ($1 \leq n \leq 2 \times 10^5$)，表示卡牌的数量。

对于接下来的 n 行，第 i 行输入两个整数 a_i 和 b_i ($0 \leq a_i \leq 10^9$, $1 \leq b_i \leq 10^6$)，表示第 i 张卡牌的消耗值和伤害。

保证所有测试数据中 n 的总和不超过 2×10^5 。

Output

每组测试数据输出一行，包含一个整数，表示最大总伤害。

Example

standard input	standard output
2	105
9	1000000
1 3	
3 10	
6 4	
1 2	
5 8	
0 5	
2 7	
6 1	
2 7	
1	
1000000000 1000000	

Note

对于第一组样例数据，按 6, 1, 9, 2, 8, 4, 7, 5, 3 的顺序打出卡牌，

总伤害为 $5 \times 1 + 3 \times 2 + 7 \times 3 + 10 \times 4 + 1 \times 1 + 2 \times 1 + 7 \times 2 + 8 \times 1 + 4 \times 2 = 105$ 。

Problem H. 拼图

有一个 n 行 m 列的网格。每个格子要么是空的（用 \cdot 表示），要么有障碍（用 $\#$ 表示）。

另外给定 k 块拼图，编号为 $1, 2, \dots, k$ 。每块拼图都是一个四连通的多格骨牌，不能旋转也不能翻转。每块拼图用它的最小外接矩形来描述，其中 $\#$ 表示属于拼图的格子， \cdot 表示不属于拼图的格子。

您需要从这 k 块拼图中选出若干块，将它们放置在网格中，使得：

- 每块被选中的拼图的每个格子，必须恰好覆盖网格中的一个空格。
- 不同拼图覆盖的格子互不相同。
- 网格中的每个空格恰好被一块拼图覆盖。

求满足条件的方案数。称两种方案是不同的，当且仅当两种方案使用的拼图集合不同，或存在某块拼图在两种方案中覆盖的格子不同。

Input

有多组测试数据。第一行输入一个整数 T ($1 \leq T \leq 10^3$)，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 ($1 \leq n, m \leq 8, 1 \leq k \leq 8$)，表示网格的行数，列数和拼图的数量。

接下来 n 行，每行输入一个长度为 m 的字符串，描述网格。其中 \cdot 表示空格子， $\#$ 表示障碍。保证网格中至少有一个空格。

接下来描述 k 块拼图。对于每块拼图，第一行输入两个整数 r 和 c ($1 \leq r, c \leq 8$)，表示该拼图最小外接矩形的行数和列数。接下来 r 行，每行输入一个长度为 c 的字符串，描述该拼图。保证每块拼图是一个四连通的多格骨牌，且外接矩形的每一行和每一列都至少包含一个 $\#$ 。

保证所有测试数据中至多 10 组满足 k 大于 5。

Output

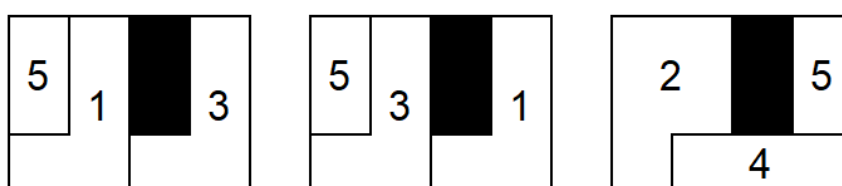
每组测试数据输出一行，包含一个整数，表示满足条件的方案数。

Example

standard input	standard output
3	3
3 4 5	0
..#.	4
..#.	
....	
3 2	
.#	
.#	
##	
3 2	
##	
##	
#.	
3 2	
.#	
.#	
##	
1 3	
###	
2 1	
#	
#	
2 2 2	
..	
..	
1 1	
#	
1 2	
##	
2 2 3	
..	
..	
1 1	
#	
1 1	
#	
1 2	
##	

Note

第一组样例数据的 3 种方案如下图所示，图中的数字表示拼图的编号。



Problem I. 版本号

在大语言模型发展的早期，研究者们发现了一个有趣的现象：当被问到“5.11 和 5.9 哪个更大”时，许多模型会自信地回答 5.11。人们猜测，这是因为语言模型混淆了小数的比较规则和版本号的比较规则——在版本号的比较中，5.11 的确大于 5.9，因为次版本号 11 大于 9。

每个版本号由主版本号 x 和次版本号 y 两个非负整数组成，记作 $x.y$ 。

版本号的比较规则如下：先比较主版本号，主版本号较大的版本更大；若主版本号相同，则比较次版本号，次版本号较大的版本更大。形式化地，版本 $x_1.y_1$ 大于版本 $x_2.y_2$ ，当且仅当 $x_1 > x_2$ ，或 $x_1 = x_2$ 且 $y_1 > y_2$ 。

给定 n 个版本号，您需要找出其中最大的一个。

Input

有多组测试数据。第一行输入一个整数 T ($1 \leq T \leq 100$)，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 ($2 \leq n \leq 100$)，表示版本号的个数。

对于接下来 n 行，第 i 行输入两个整数 x_i 和 y_i ($0 \leq x_i, y_i \leq 10^3$)，表示版本号 $x_i.y_i$ 。

Output

每组测试数据输出一行，包含两个由单个空格分隔的整数 x' 和 y' ，表示最大版本号的主版本号和次版本号。

Example

standard input	standard output
3	5 11
3	1 0
5 11	0 0
5 9	
3 12	
2	
1 0	
0 1	
2	
0 0	
0 0	

Problem J. 插松枝

人造松枝加工场的工人需要将各种尺寸的塑料松针插到枝干上，做成大大小小的装饰品。

桌子上有一根空的枝干以及两只盒子。其中一只盒子是空的，另一只盒子装有堆成一叠的 n 片松针，从盒顶到盒底第 i 片松针的大小为 a_i 。每次工人可以执行以下两种操作中的一种：

- 选择一只装有松针的盒子，将盒子顶部的一片松针移动到另一只盒子的顶部。
- 选择一只装有松针的盒子，将盒子顶部的一片松针插到枝干上。

工人需要把所有 n 片松针都插到枝干上，且每次将松针插到枝干上时，不能比之前插上的松针更大。也就是说，令 b_i 表示被插到枝干上的第 i 片松针的大小，则有 $b_1 \geq b_2 \geq \dots \geq b_n$ 。

您要求出工人至少执行几次操作，才能将所有松针都插到枝干上。

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：

第一行输入一个整数 n ($1 \leq n \leq 4 \times 10^5$)，表示盒子里有几片松针。

第二行输入 n 个整数 a_1, a_2, \dots, a_n ($1 \leq a_i \leq 10^9$)，其中 a_i 表示从盒顶到盒底第 i 片松针的大小。

保证所有测试数据中 n 的总和不超过 4×10^5 。

Output

每组测试数据输出一行，包含一个整数，表示最少可能的操作次数。

Example

standard input	standard output
2	8
5	3
7 3 10 7 2	
3	
1 1 1	

Note

以下展示第一组样例数据的一种最优策略。盒子里的松针大小按从顶到底的顺序列出，枝干上的松针大小按插入顺序列出。

#	盒子 A	盒子 B	枝干	#	盒子 A	盒子 B	枝干
1	{3, 10, 7, 2}	{7}	{}	5	{3, 2}	{7}	{10, 7}
2	{10, 7, 2}	{3, 7}	{}	6	{3, 2}	{}	{10, 7, 7}
3	{7, 2}	{3, 7}	{10}	7	{2}	{}	{10, 7, 7, 3}
4	{2}	{3, 7}	{10, 7}	8	{}	{}	{10, 7, 7, 3, 2}

Problem K. 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，其中有 k 个节点是特殊节点，称一棵生成树是好的，若所有特殊节点都是生成树的叶子。

您需要求出所有好生成树中，边权之和的最小值。

请回忆：生成树是原图的连通子图，该连通子图含有 n 个点以及 $(n - 1)$ 条边。另外，叶子指的是树上恰有一条边与其相连的点。

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 n ， m 和 k ($2 \leq n \leq 2 \times 10^5$ ， $n - 1 \leq m \leq 2 \times 10^5$ ， $1 \leq k \leq n$)，表示图的点数，边数，和特殊节点数。

第二行输入 k 个互不相同的整数 a_1, a_2, \dots, a_k ($1 \leq a_i \leq n$)，表示特殊节点。

对于接下来的 m 行，第 i 行输入三个整数 u_i ， v_i 和 w_i ($1 \leq u_i, v_i \leq n$ ， $1 \leq w_i \leq 10^9$)，表示有一条边权为 w_i 的边连接点 u_i 和 v_i 。保证输入的图是连通的，但可能有自环或重边。

保证所有测试数据中 n 的总和与 m 的总和均不超过 2×10^5 。

Output

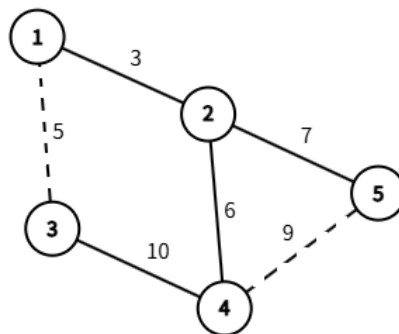
每组测试数据输出一行，包含一个整数，表示好生成树中边权之和的最小值。若不存在好生成树，则输出 -1 。

Example

standard input	standard output
3	26
5 6 2	-1
1 5	1010
1 2 3	
2 5 7	
4 2 6	
5 4 9	
3 4 10	
1 3 5	
4 4 4	
1 2 3 4	
1 2 1	
2 3 1	
3 4 1	
4 1 1	
3 4 1	
1	
1 2 10	
1 2 100	
2 3 1000	
3 3 100000	

Note

第一组样例数据如下图所示。实线表示生成树中的边，虚线表示不在生成树中的边。



Problem L. 分数迭代

给定一个最简分数 $\frac{a}{b}$ (即 $\gcd(a, b) = 1$)。每一秒, 分子和分母同时加 1, 然后立即化简为最简形式。

具体地, 若当前分数为 $\frac{a}{b}$, 一秒后变为 $\frac{a+1}{b+1}$, 再令 $g = \gcd(a+1, b+1)$, 更新为 $\frac{(a+1)/g}{(b+1)/g}$ 。

现在给定初始分数 $\frac{a}{b}$ 以及 q 次询问。每次询问给出一个整数 k_i , 您需要求出从初始状态开始经过恰好 k_i 秒后的分数。

Input

有多组测试数据。第一行输入一个整数 T ($1 \leq T \leq 10^3$), 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 a, b 和 q ($1 \leq a, b \leq 10^{12}, 1 \leq q \leq 100$)。保证初始时 $\gcd(a, b) = 1$ 。

对于接下来的 q 行, 第 i 行输入一个整数 k_i ($1 \leq k_i \leq 10^{18}$), 表示询问的时间。

Output

每个询问输出一行, 包含两个由单个空格分隔的整数 a' 和 b' , 表示从初始状态开始经过恰好 k_i 秒后的分数的分子和分母。

Example

standard input	standard output
3	1 4
1 7 3	2 5
1	1 2
2	1 1
3	5 4
1 1 1	8 3
1	8 7
13 3 3	
7	
2	
10	

Problem M. 博物馆奇妙夜 2

一名保安在博物馆内巡逻以监控展品。巡逻路线是一条由 n 个点组成的闭合折线，这些点从 1 到 n 标号，顺次连接（最后一个点连接回第一个点）形成 n 条首尾相接的线段。保安沿 $1 \rightarrow 2 \rightarrow \dots \rightarrow n \rightarrow 1 \rightarrow \dots$ 的顺序巡逻。当保安不在线段端点时，其视野是半径为 $10^{10^{10}}$ 、圆心角为 $2a$ 的扇形，且视野中线方向与当前巡逻线段的前进方向相同。

称从巡逻线段上的一个点 P （不含线段端点）可以看到位于点 Q 的展品，当且仅当 Q 位于以 P 为顶点的扇形区域内。给定由 n 个点组成的闭合折线巡逻路线和 m 件展品的位置，求能够同时看到所有展品的巡逻路线的总长度。

Input

每个测试文件中只有一组测试数据。

第一行输入三个整数 n ， m 和 a ($3 \leq n \leq 2 \times 10^5$, $1 \leq m \leq 2 \times 10^5$, $0 < a < 90$)，分别表示闭合折线的点数，展品的数量和视野圆心角的一半（以度为单位）。

对于接下来的 n 行，第 i 行输入两个整数 x_i 和 y_i ($-10^9 \leq x_i, y_i \leq 10^9$)，表示折线上第 i 个点的坐标。连续的点形成巡逻线段，最后一个点连接回第一个点形成闭环。保证没有两个连续的点（包括最后一个和第一个点）是相同的。

对于接下来的 m 行，第 i 行输入两个整数 x'_i 和 y'_i ($-10^9 \leq x'_i, y'_i \leq 10^9$)，表示第 i 件展品的坐标。

Output

输出一行，包含一个实数，表示能够同时看到所有展品的巡逻路线的总长度。

如果您的答案的绝对误差或相对误差不超过 10^{-6} ，则将被视为正确。更正式地，假设您的输出为 a ，标准答案为 b ，当且仅当 $\frac{|a-b|}{\max(1,|b|)} \leq 10^{-6}$ 时，您的输出才会被接受。

Examples

standard input	standard output
3 2 45 1 1 5 5 8 1 2 3 4 2	4.4142135624
4 1 60 0 0 2 0 2 2 0 2 1 1	1.6905989232

Note

第一组样例数据如下图所示。A 和 B 是两件展品，G 是保安。

