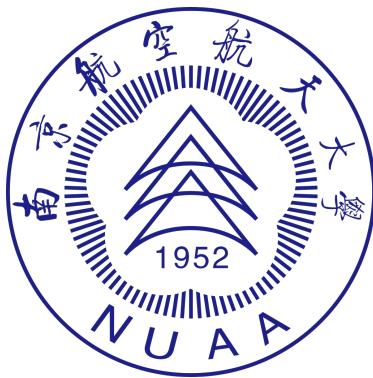


2024  icpc 国际大学生程序设计竞赛
亚洲区域赛（南京站）

正式赛

2024 年 11 月 3 日



试题列表

A	嘿，有看到我的袋鼠吗？
B	生日礼物
C	拓扑
D	棋字井
E	左移 3
F	地铁
G	二叉树
H	博德之跃 2
I	Bingo
J	社交媒体
K	纸条
L	$P \text{ xor } Q = R$
M	我将如潮水般归来

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。

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. 嘿，有看到我的袋鼠吗？

请注意本题不同寻常的空间限制。

继 2018, 2019, 2020, 2021, 2022 和 2023 年成功承办赛事之后，南京航空航天大学（NUAA）将连续第七年承办国际大学生程序设计竞赛（ICPC）。

在 2018 与 2019 年，“中二之力”队与“三个顶俩”队为清华大学赢得了冠军。在 2020, 2021 与 2022 年，北京大学的“逆十字”队赢得三连冠。在 2023 年，来自北京大学的另一支队伍“重生之我是菜狗”赢得了冠军。他们还赢得了第 46 届 ICPC 世界总决赛的冠军，在 13 年后为 EC 赛区重新赢回了奖杯。

今年，将会有约 335 支队伍参与南京站的竞赛。本次竞赛将会颁发至多 33 项金奖，66 项银奖与 99 项铜奖（数字仅供参考）。让我们期待选手们出色的表现！我们还想要感谢竞赛组委会与志愿者们的努力付出。感谢你们为本次竞赛做出的贡献！



在 2023 ICPC 国际大学生程序设计竞赛亚洲区域赛（南京站）中拍摄的照片

在 2018 年的竞赛中，K 题《袋鼠谜题》要求选手为以下游戏构造一个操作序列：

谜题由一个 n 行 m 列的网格 ($1 \leq n, m \leq 20$) 组成，且有一些（至少 2 只）袋鼠位于网格中。玩家的目标是控制袋鼠并把它们聚集在同一个格子中。一些格子里有墙，袋鼠无法进入这些有墙的格子，而其它格子是空的。袋鼠可以从一个空格子移动到上，下，左，右相邻的另一个空格子中。

游戏开始时，每个空格子里都有一只袋鼠。玩家可以通过键盘上 U, D, L, R 四个按键控制袋鼠的移动。所有袋鼠会同时根据您按下的按键移动。

选手需要构造一个长度至多为 5×10^4 且由 U, D, L, R 组成的操作序列以达成目标。

在 2020 年的竞赛中，A 题《啊，昨日重现》要求选手构造一张输入地图，以证明以下代码并不是上述问题的解：

```
#include <bits/stdc++.h>
using namespace std;
string s = "UDLR";
int main()
{
    srand(time(NULL));
    for (int i = 1; i <= 50000; i++) putchar(s[rand() % 4]);
    return 0;
}
```

此外，在 2021 年的竞赛（A 题，《呀，昨日再次重现》），2022 年的竞赛（A 题，《停停，昨日请不要再重现》）和 2023 年的竞赛（A 题，《酷，昨日四次重现》）中，每年都有一道与袋鼠相关的问题！我们很想向您介绍所有这些问题，但如果每年都这样做，在 2024 年的竞赛中将会有一道题拥有长达 500 页的题面。因此，这次我们省略它们。另外，您可能已经在热身赛中见过它们了。

在 2024 年的竞赛中，如大家期待的那样，袋鼠题又回来啦！我们不知道为什么命题组的成员们那么喜欢袋鼠，但题目如下：

给定一张 n 行 m 列的网格。一些格子里有墙，袋鼠无法进入这些有墙的格子，而其它格子是空的，每个空格子中都有一只袋鼠。袋鼠可以从一个空格子移动到上，下，左，右相邻的另一个空格子中。

袋鼠可以被键盘上的 U, D, L, R 键控制。所有袋鼠会同时根据按下的按键移动。具体来说，对于一只位于第 i 行第 j 列的格子（用 (i, j) 表示）上的袋鼠：

1. 按键 U：若 $i > 1$ 且 $(i - 1, j)$ 不是墙，它会移动到 $(i - 1, j)$ ，否则它会待在原地。
2. 按键 D：若 $i < n$ 且 $(i + 1, j)$ 不是墙，它会移动到 $(i + 1, j)$ ，否则它会待在原地。
3. 按键 L：若 $j > 1$ 且 $(i, j - 1)$ 不是墙，它会移动到 $(i, j - 1)$ ，否则它会待在原地。
4. 按键 R：若 $j < m$ 且 $(i, j + 1)$ 不是墙，它会移动到 $(i, j + 1)$ ，否则它会待在原地。

给定一个仅由字符 ‘U’，‘D’，‘L’，‘R’ 组成的操作序列 $s_1 s_2 \dots s_k$ ，我们将根据序列进行无限次操作。具体来说，若 $1 \leq t \leq k$ ，则第 t 次操作就是 s_t ；否则若 $t > k$ ，则第 t 次操作和第 $(t - k)$ 次操作相同。对于每个 $1 \leq i \leq n \times m$ ，求最小的整数 v_i ，使得执行 v_i 次操作后，最多有 i 个格子里含有袋鼠。

Input

每个测试文件仅有一组测试数据。

第一行输入三个整数 n , m 和 k ($1 \leq n, m \leq 2 \times 10^5$, $1 \leq n \times m \leq 2 \times 10^5$, $1 \leq k \leq 200$)，表示网格的行数和列数，以及操作序列的长度。

第二行输入一个字符串 $s_1 s_2 \dots s_k$ ($s_i \in \{‘U’, ‘D’, ‘L’, ‘R’\}$)，表示操作序列。

对于接下来 n 行，第 i 行输入一个二进制字符串 $a_{i,1} a_{i,2} \dots a_{i,m}$ ($a_{i,j} \in \{‘0’, ‘1’\}$)。若 $a_{i,j} = ‘1’$ 则格子 (i, j) 是空的；否则若 $a_{i,j} = ‘0’$ 则格子 (i, j) 被阻塞，无法进入。保证网格中至少有一个空格子。

Output

输出 $n \times m$ 行，其中第 i 行输出一个整数 v_i ，表示最少需要几次操作，才能使最多有 i 个格子里含有袋鼠。如果不可能做到，则在这一行输出 -1。

Examples

standard input	standard output
3 3 6 ULDDRR 010 111 010	-1 4 2 1 0 0 0 0
3 3 6 ULDDRR 010 111 011	7 4 2 1 1 0 0 0
1 5 1 R 11111	4 3 2 1 0

Problem B. 生日礼物

Grammy 的生日快要来了，她从她的朋友那里获得了一个序列 A 作为礼物。序列由 0, 1 和 2 构成。Grammy 觉得这个序列太长了，所以她打算把 A 修改得短一些。

更正式地，Grammy 可以执行任意次操作。每次她可以执行以下三种操作之一：

- 将任意一个 2 改为 0 或 1。
- 选择两个相邻的 0，删除它们，并将剩下的部分连接起来。
- 选择两个相邻的 1，删除它们，并将剩下的部分连接起来。

求 Grammy 能得到的最短序列的长度。

Input

有多组测试数据。第一行输入一个整数 T 表示测试数据组数。对于每组测试数据：

第一行输入一个长度为 n 的字符串 ($1 \leq n \leq 2 \times 10^5$)。字符串由数字 0, 1 和 2 构成，表示初始序列 A 。

保证所有数据 n 之和不超过 5×10^5 。

Output

每组数据输出一行一个整数，表示 Grammy 能得到的最短序列的长度。

Example

standard input	standard output
5	3
0110101	4
01020102	0
0000021111	6
1012121010	0
0100202010	

Problem C. 拓扑

给定一棵由 n 个节点组成的树，其中节点 1 是根。保证每个节点的编号都比它所有子节点小。树的拓扑序是一个满足以下限制的 n 的排列 p_1, p_2, \dots, p_n : 对于所有 $1 \leq i < j \leq n$, 节点 p_j 都不是节点 p_i 的父节点。

对于每个 $1 \leq i \leq n$, 计算给定的树有多少拓扑序满足 $p_i = i$ 。答案对 998 244 353 取模。

Input

每个测试文件仅有一组测试数据。

第一行输入一个整数 n ($2 \leq n \leq 5\,000$)，表示树的节点数量。

第二行输入 $(n - 1)$ 个整数 f_2, f_3, \dots, f_n ($1 \leq f_i < i$)，其中 f_i 是节点 i 的父节点。

Output

输出一行 n 个由单个空格分隔的整数 a_1, a_2, \dots, a_n ，其中 a_i 表示给定的树有多少拓扑序满足 $p_i = i$ 。答案对 998 244 353 取模。

Examples

standard input	standard output
4 1 1 2	3 2 1 2
9 1 1 2 2 3 3 4 5	672 420 180 160 152 108 120 170 210

Note

对于第一组样例数据，树的拓扑序有: $\{1, 2, 3, 4\}$, $\{1, 3, 2, 4\}$ 和 $\{1, 2, 4, 3\}$ 。其中有 3 个序列满足 $p_1 = 1$, 2 个序列满足 $p_2 = 2$, 1 个序列满足 $p_3 = 3$, 以及 2 个序列满足 $p_4 = 4$ 。

Problem D. 棋字井

Alice 和 Bob 正在 n 块 3×3 列的棋盘上玩“棋字井”游戏。一些棋盘的一些格子一开始是空的，而其它格子里都有一些记号。Alice 先行动，他们轮流选择一个棋盘，并在该棋盘的一个空格中画上自己的记号。Alice 的记号是 ‘x’，Bob 的记号是 ‘o’。

每位玩家必须确保在他/她行动之后，任何棋盘的任何行、列或对角线上都没有三个相同的标记。无法在自己回合进行有效行动的玩家将输掉游戏，同时宣告另一位玩家获胜。

给定 n 块棋盘的初始状态，假设两位玩家都采用最优策略，您需要确定谁会获胜。

Input

有多组测试数据。第一行输入一个整数 T 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 ($1 \leq n \leq 10^5$)，表示游戏中的棋盘数量。

接下来输入 n 块大小为 3×3 的棋盘。对于每块棋盘：

- 如果不是第一块棋盘，首先会有一个空行。
- 对于接下来三行，第 i 行输入一个长度为 3 的字符串 $s_{i,1}s_{i,2}, s_{i,3}$ ，字符串由字符 ‘x’，‘o’ 和 ‘.’ 组成，描述一块大小为 3×3 的棋盘。令 (i, j) 表示位于第 i 行第 j 列的格子。若 $s_{i,j} = 'x'$ 则格子 (i, j) 里有记号 ‘x’；若 $s_{i,j} = 'o'$ 则格子 (i, j) 里有记号 ‘o’；若 $s_{i,j} = '.'$ 则格子 (i, j) 是空的。

保证任何棋盘的任何行、列或对角线上都没有三个相同的标记。另外保证所有数据 n 之和不超过 10^5 。

Output

每组数据输出一行。如果 Alice 获胜，则输出 Alice；如果 Bob 获胜，则输出 Bob。

Example

standard input	standard output
4	Alice
1	Alice
...	Bob
...	Bob
...	
1	
...	
oo.	
oo.	
2	
...	
oo.	
oo.	
...	
xx.	
xx.	
2	
..x	
xo.	
...	
xo.	
o..	
.x.	

Problem E. 左移 3

给定一个长度为 n 的字符串 $S = s_0s_1 \cdots s_{n-1}$ ，您可以将 S 左移至多 k 次（包括零次）。求操作之后，字符串中最多含有几个“nanjing”子串。

更正式地，令 $f(S, d)$ 表示将 S 左移 d 次后获得的字符串。也就是说 $f(S, d) = s_{(d+0) \bmod n}s_{(d+1) \bmod n} \cdots s_{(d+n-1) \bmod n}$ 。令 $g(f(S, d), l, r) = s_{(d+l) \bmod n}s_{(d+l+1) \bmod n} \cdots s_{(d+r) \bmod n}$ 。令 $h(d)$ 表示整数对 (l, r) 的数量，满足 $0 \leq l \leq r < n$ 且 $g(f(S, d), l, r) = \text{nanjing}$ 。找到一个整数 d 满足 $0 \leq d \leq k$ 并最大化 $h(d)$ 。输出这个最大化的值。

Input

有多组测试数据。第一行输入一个整数 T 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 ($1 \leq n \leq 2 \times 10^5$, $0 \leq k \leq 10^9$)，表示字符串的长度和最多能进行几次左移操作。

第二行输入一个长度为 n 的字符串 $s_0s_1 \cdots s_{n-1}$ 。字符串由小写英文字母组成。

保证所有数据 n 之和不超过 5×10^5 。

Output

每组数据输出一行一个整数，表示字符串中最多含有几个“nanjing”子串。

Example

standard input	standard output
4	2
21 10	1
jingicpcnanjingsuanan	3
21 0	0
jingicpcnanjingsuanan	
21 3	
nanjingnanjingnanjing	
4 100	
icpc	

Note

对于第一组样例数据，我们可以将字符串左移 6 次，得到字符串“pcnanjingsuananjic”。其中有两个“nanjing”子串。

对于第二组样例数据，因为 $k = 0$ ，我们无法进行任何左移操作。原字符串中有一个“nanjing”子串。

Problem F. 地铁

Pigeland 的地铁系统非常先进。地铁系统由 n 座车站构成，编号从 1 到 n ，还有 k 条有向地铁线，编号从 1 到 k 。线路 i 按顺序经过车站 $x_{i,1}, x_{i,2}, \dots, x_{i,p_i}$ ，其中 $x_{i,j}$ 是线路 i 经过的第 j 座车站。搭乘线路 i 从车站 $x_{i,j}$ 到车站 $x_{i,j+1}$ 需要花 $w_{i,j}$ 单位时间。

当多条线路经过同一车站时，乘客可以在线路之间换乘。若乘客目前位于线路 x 上的一座车站，而线路 y 也经过该车站，他/她就能花 $a_y \times b_x$ 单位时间从线路 x 换乘到线路 y ，其中 a_y 和 b_x 是线路 y 和 x 给定的参数。换乘后，乘客位于相同车站的线路 y 中。

您将从车站 1 出发。对所有 $2 \leq s \leq n$ ，求到达车站 s 需要的最短时间。更具体地，您可以选择从车站 1 的任意线路出发，出发时不消耗换乘时间。保证所有车站都能从车站 1 到达。

Input

每个测试文件仅有一组测试数据。

第一行输入两个整数 n 和 k ($2 \leq n \leq 2 \times 10^5$, $1 \leq k \leq 2 \times 10^5$)，表示车站的数量和地铁线的数量。

第二行输入 k 个整数 a_1, a_2, \dots, a_k ($1 \leq a_i \leq 10^6$)。

第三行输入 k 个整数 b_1, b_2, \dots, b_k ($1 \leq b_i \leq 10^6$)。

对于接下来 k 行，第 i 行首先输入一个整数 p_i ($2 \leq p_i \leq n$)，表示线路 i 经过的车站数。接下来输入 $(2p_i - 1)$ 个整数 $x_{i,1}, w_{i,1}, x_{i,2}, \dots, x_{i,p_i-1}, w_{i,p_i-1}, x_{i,p_i}$ ($1 \leq x_{i,j} \leq n$, $1 \leq w_{i,j} \leq 10^9$)，其中 $x_{i,j}$ 是线路 i 经过的第 j 座车站， $w_{i,j}$ 是搭乘线路 i 从车站 $x_{i,j}$ 到车站 $x_{i,j+1}$ 的耗时。一条地铁线经过的车站互不相同。

保证 $\sum_{i=1}^k (p_i - 1) \leq 2 \times 10^5$ 。

Output

输出一行 $(n - 1)$ 个由单个空格分隔的整数 d_2, d_3, \dots, d_n ，其中 d_i 是从车站 1 到车站 i 的最短时间。

Examples

standard input	standard output
6 3 1 5 1 5 5 1 3 1 2 2 3 3 3 5 1 2 1 4 3 3 4 5 4 6	2 5 21 14 18
6 3 1 5 1 5 5 1 5 1 2 2 100 3 100 6 1 4 5 1 100 2 4 3 100 5 1 4 2 3 1 5	2 31 43 37 136

Problem G. 二叉树

这是一道交互题。

给定一棵有 n 个节点的二叉树，您需要用至多 $p = \lfloor \log_2 n \rfloor$ 次询问找到树中的一个特殊节点 s 。也就是说， p 是满足 $2^p \leq n$ 的最大整数。

每次询问包含两个不同的节点 u 和 v 。裁判程序会输出一个整数 t ($0 \leq t \leq 2$) 表示询问的答案。令 $d(a, b)$ 表示从节点 a 到节点 b 的简单路径上有几条边。

- 若 $t = 0$ ，则节点 u 离特殊节点更近。也就是说， $d(u, s) < d(v, s)$ 。
- 若 $t = 1$ ，则节点 u 和节点 v 到特殊节点的距离相同。也就是说， $d(u, s) = d(v, s)$ 。
- 若 $t = 2$ ，则节点 v 离特殊节点更近。也就是说， $d(u, s) > d(v, s)$ 。

请注意：裁判程序是适应性的。也就是说，每组测试数据的答案不是事先确定的。裁判程序可以根据您的询问决定特殊节点，只要它的答案与之前的询问和答案不冲突即可。

Input

有多组测试数据。第一行输入一个整数 T 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 ($2 \leq n \leq 10^5$) 表示二叉树中节点的数量。

对于接下来 n 行，第 i 行输入两个整数 x_i 和 y_i ($0 \leq x_i, y_i \leq n$)，表示第 i 个节点的左子节点和右子节点。若 $x_i = 0$ ，则第 i 个节点没有左子节点；若 $y_i = 0$ ，则第 i 个节点没有右子节点。

保证所有数据 n 之和不超过 2×10^5 。

Interaction Protocol

要提出询问，请输出一行。首先输出 `?`，之后跟一个空格，然后输出两个不同的由单个空格分隔的整数 u 和 v ($1 \leq u, v \leq n$)。在清空输出缓冲区之后，您的程序需要读入一个整数 t ，表示对您的询问的回答。

要猜测特殊节点，请输出一行。首先输出 `!`，之后跟一个空格，然后输出一个整数 s ($1 \leq s \leq n$) 表示特殊节点。在清空输出缓冲区之后，您的程序应该马上开始处理下一组测试数据。如果没有更多测试数据，您的程序应该立即退出。还请注意，猜测特殊节点不算一次询问。

清空输出缓冲区可以使用以下方式：

- C 和 C++ 使用 `fflush(stdout)`（如果您使用 `printf`）或 `cout.flush()`（如果您使用 `cout`）。
- Java 使用 `System.out.flush()`。
- Python 使用 `stdout.flush()`。

Example

standard input	standard output
2	
5	
0 0	
1 5	? 5 1
2 4	? 1 4
0 0	! 2
0 0	
1	
0	
2	
0 2	
0 0	
2	
2	? 2 1
	! 1

Problem H. 博德之跃 2

您有一个由小写英文字母组成的字符串 S 。您需要对 S 执行若干次操作，直到它变为空字符串。每次您可以执行以下三种操作中的一种：

1. 删除 S 的第一个字符。
2. 删除 S 的最后一个字符。
3. 选择 S 的一个好子串 S' ，并将 S 替换为 S' 。

一个非空字符串 S' 被称为 S 的好子串，当且仅当 $S' \neq S$, S' 是 S 的前缀，且 S' 的反串是 S 的后缀。长度为 k 的字符串 $p_1p_2 \cdots p_k$ 的反串是另一个长度为 k 的字符串 $p_kp_{k-1} \cdots p_1$ 。

求最多能执行多少次第 3 种操作。

Input

有多组测试数据。第一行输入一个整数 T 表示测试数据组数。对于每组测试数据：

第一行输入一个由小写字母组成的字符串 S ($1 \leq |S| \leq 10^5$)。

保证所有测试数据 $|S|$ 之和不超过 2×10^5 。

Output

每组测试数据输出一行一个整数，表示最多能执行多少次第 3 种操作。

Example

standard input	standard output
3	3
aaaa	4
abbaabba	0
xy	

Note

对于第一组样例数据： $\text{aaaa} \xrightarrow{\text{op. 3}} \text{aaa} \xrightarrow{\text{op. 3}} \text{aa} \xrightarrow{\text{op. 3}} \text{a} \xrightarrow{\text{op. 2}} \emptyset$ 。

对于第二组样例数据： $\text{abbaabba} \xrightarrow{\text{op. 3}} \text{abbaabb} \xrightarrow{\text{op. 1}} \text{bbaabb} \xrightarrow{\text{op. 3}} \text{bbaab} \xrightarrow{\text{op. 1}} \text{baab} \xrightarrow{\text{op. 3}} \text{baa} \xrightarrow{\text{op. 1}} \text{aa}$
 $\xrightarrow{\text{op. 3}} \text{a} \xrightarrow{\text{op. 1}} \emptyset$ 。

Problem I. Bingo

给定两个整数 n 和 m , 以及一个长度为 $n \times m$ 的整数序列 a_1, a_2, \dots, a_{nm} , 我们将用序列里的数字填入一个 n 行 m 列的网格。具体来说, 令 (i, j) 表示位于第 i 行第 j 列的格子, 我们会将序列里的第 $((i-1) \times m + j)$ 个数 (也就是 $a_{(i-1) \times m + j}$) 填入那个格子中。

称整数 k 是序列的“bingo 整数”, 若将所有数字填入格子后, 以下两个条件至少满足一个。

- 至少存在一行, 使得那一行所有格子里的整数都小于等于 k 。
- 至少存在一列, 使得那一列所有格子里的整数都小于等于 k 。

容易发现, 一个序列可以有很多 bingo 整数。不过本题中, 我们只对最小的 bingo 整数感兴趣。

对于给定序列的所有 $(nm)!$ 个排列, 求每个排列的最小 bingo 整数之和。由于答案可能很大, 请将答案对 998 244 353 取模后输出。

Input

有多组测试数据。第一行输入一个整数 T 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 ($1 \leq n, m \leq 2 \times 10^5$, $1 \leq n \times m \leq 2 \times 10^5$) , 表示网格的行数和列数。

第二行输入 $n \times m$ 个整数 a_1, a_2, \dots, a_{nm} ($0 \leq a_i < 998 244 353$) 表示给定序列。

保证所有数据 $n \times m$ 之和不超过 4×10^5 。

Output

每组数据输出一行一个整数表示答案。

Example

standard input	standard output
4	56
2 2	60
1 3 2 4	60
3 1	855346687
10 10 10	
1 3	
20 10 30	
3 4	
1 1 4 5 1 4 1 9 1 9 8 10	

Note

对于第一组样例数据, 如果 1 和 2 不在同一行或同一列, 那么最小 bingo 整数就是 3, 否则最小 bingo 整数就是 2。在 8 个排列中, 1 和 2 不在同一行或同一列, 所以答案是 $8 \times 3 + (4! - 8) \times 2 = 56$ 。

对于第二组样例数据, 最小 bingo 整数总是 10, 所以答案是 $3! \times 10 = 60$ 。

Problem J. 社交媒体

在一个社交媒体平台上，用户可以在别人的帖子下方留下评论以发表自己的感想。不过这些评论并非对所有人可见。具体来说，如果用户 C 想要看到用户 A 对用户 B 的帖子的评论，他/她必须同时与 A 和 B 是好友关系。如果用户在自己的帖子下方留下评论，那么他/她的所有好友都能看到这条评论。

作为该平台的活跃用户，您想要看到尽可能多的评论。平台上目前有 k 名用户（除您以外），编号从 1 到 k 。平台上还有 m 条评论，然而您可能无法看到所有评论，因为您只有 n 位好友。由于您需要参加 2024 ICPC 国际大学生程序设计竞赛亚洲区域赛南京站，您没有时间结交太多新朋友。问：如果您至多新增两位好友，最多可以看到几条评论。

Input

有多组测试数据。第一行输入一个整数 T 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 ($1 \leq n \leq k \leq 2 \times 10^5$, $1 \leq m \leq 2 \times 10^5$)，表示您的好友数量，评论的数量，以及平台上的用户数（除您以外）。

第二行输入 n 个不同的整数 f_1, f_2, \dots, f_n ($1 \leq f_i \leq k$) 表示您在平台上的好友。

对于接下来 m 行，第 i 行输入两个整数 a_i 和 b_i ($1 \leq a_i, b_i \leq k$) 表示用户 a_i 在用户 b_i 的帖子下留下的一条评论。

保证所有数据 k 之和与 m 之和均不超过 2×10^5 。

Output

每组数据输出一行一个整数，表示如果您在平台上新增至多两位好友，最多能看到几条评论。

Example

standard input	standard output
5	9
4 12 7	5
5 7 3 6	1
3 6	1
2 2	1
1 4	
2 4	
1 3	
7 6	
4 1	
5 4	
1 1	
1 1	
2 1	
3 7	
2 7 6	
2 4	
1 2	
3 2	
2 5	
5 4	
2 6	
4 6	
2 6	
1 1 2	
1	
1 2	
2 1 2	
1 2	
1 2	
2 1 100	
24 11	
11 24	

Note

对于第一组样例数据，您可以和用户 1 与 4 成为好友。

对于第二组样例数据，您可以和用户 5 与 6 成为好友。

对于第三组样例数据，您可以和用户 2 成为好友。

对于第四和第五组样例数据，您不需要新增好友，因为您已经可以看到所有评论。

Problem K. 纸条

有 w 个格子排成一行，从左到右编号从 1 到 w 。这些格子中，有 n 个是红色的， m 个是黑色的，剩下的 $(w - n - m)$ 个是白色的。

您需要用一些纸条覆盖所有红色格子。每张纸条必须覆盖 k 个连续的格子。找到覆盖所有红色格子的方式，同时还要满足以下所有限制：

- 每个红色格子都被纸条覆盖。
- 没有黑色格子被纸条覆盖。
- 没有两张纸条覆盖了同一个格子。也就是说，每个格子最多被一张纸条覆盖。
- 使用的纸条数尽可能小。

Input

有多组测试数据。第一行输入一个整数 T 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 n , m , k 和 w ($1 \leq n, m \leq 10^5$, $1 \leq k \leq w \leq 10^9$, $n + m \leq w$)，表示红色格子的数量，黑色格子的数量，每张纸条的长度和格子的总数。

第二行输入 n 个整数 a_1, a_2, \dots, a_n ($1 \leq a_i \leq w$)，表示格子 a_i 是红色的。

第三行输入 m 个整数 b_1, b_2, \dots, b_m ($1 \leq b_i \leq w$)，表示格子 b_i 是黑色的。

保证所有给定的 $(n + m)$ 个格子互不相同。同时保证所有数据 n 之和与 m 之和均不超过 2×10^5 。

Output

对于每组数据：

如果可以覆盖所有红色格子，同时满足所有限制，首先输出一行一个整数 c 表示最少使用几张纸条。接下来输出一行 c 个由单个空格分隔的整数 l_1, l_2, \dots, l_c ($1 \leq l_i \leq w - k + 1$)，其中 l_i 表示第 i 张纸条覆盖的最左边的格子。如果有多种合法答案，您可以输出任意一种。

如果无法完成要求，只要输出一行 -1 。

Example

standard input	standard output
4	4
5 2 3 16	6 2 14 9
7 11 2 9 14	-1
13 5	2
3 2 4 11	1 4
6 10 2	-1
1 11	
2 1 2 6	
1 5	
3	
2 1 2 6	
1 5	
2	

Problem L. $P \oplus Q = R$

Alice 想要训练自己解决构造题的能力。所以她的朋友，超级人工智能 Kei，为 Alice 生成了以下问题。

给定一个整数 n ，构造两个 $0, 1, \dots, (n - 1)$ 的排列 $P = p_1, p_2, \dots, p_n$ 和 $Q = q_1, q_2, \dots, q_n$ ，使得序列 $R = r_1, r_2, \dots, r_n$ 仍然是一个 $0, 1, \dots, (n - 1)$ 的排列，其中 $r_i = p_i \oplus q_i$ 。这里 $x \oplus y$ 表示 x 和 y 按位异或的结果。

Alice 利用她强大的计算能力解决了这个问题，现在她决定和您分享这个问题。您能解决它吗？

Input

有多组测试数据。第一行输入一个整数 T 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 ($1 \leq n \leq 2 \times 10^5$) 表示排列的长度。

保证所有数据 n 之和不超过 2×10^6 。

Output

对于每组数据：

如果存在符合要求的两个排列，首先输出一行 **Yes**。接下来输出第二行，包含 n 个由单个空格分隔的整数 p_1, p_2, \dots, p_n 。最后输出第三行，包含 n 个由单个空格分隔的整数 q_1, q_2, \dots, q_n 。如果有多种合法答案，您可以输出任意一种。

如果不存在符合要求的两个排列，只要输出一行 **No**。

Example

standard input	standard output
2	No
3	Yes
4	0 2 1 3 3 2 0 1

Note

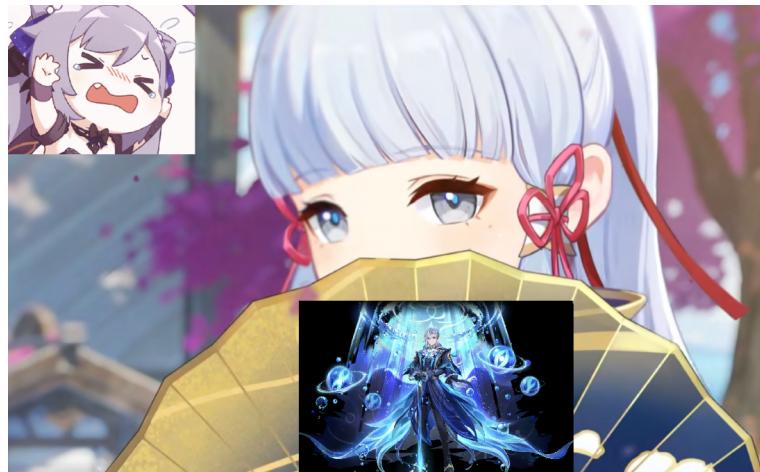
对于第二组样例数据， $R = \{3, 0, 1, 2\}$ 仍然是 $0, 1, 2, 3$ 的排列。



Problem M. 我将如潮水般归来

那维莱特是枫丹的最高审判官，因其无懈可击的「秉公无私」而闻名。作为世界著名游戏《原神》中的可玩角色，他以其强大的蓄力攻击而闻名，该类攻击可以一次性击中特定范围内的敌人。

由于他非常强大，许多玩家在挑战几乎每个任务时都会使用他。然而，提瓦特中并非所有人都对此感到高兴，尤其是其他 ADC（主要输出角色），比如神里绫华、刻晴等。于是，他们决定说服米哈游在游戏中削弱那维莱特。为此，他们必须提交一份关于那维莱特在一些场景下的伤害报告。



基于《原神》官方素材制作

每个战斗场景都发生在一个二维平面上。那维莱特站在 $(0, 0)$ ，最初面朝 (x_0, y_0) ，进行持续 t 单位时间的蓄力攻击，并以每单位时间 1 弧度的速度逆时针旋转。也就是说，那维莱特会在 2π 单位时间内逆时针转一圈。

考虑从 $(0, 0)$ 指向那维莱特面朝方向的射线，攻击范围是距离射线最多为 d 的点的集合。如果目标（一个凸多边形）与攻击范围有公共点，它将每单位时间受到 1 点持续伤害。

作为一名经验丰富的程序员，您被绫华召唤。这次，您的任务是计算目标在前 t 单位时间内所遭受的伤害。

Input

每个测试文件仅有一组测试数据。

第一行输入五个整数 n , x_0 , y_0 , d 和 t ($3 \leq n \leq 100$, $-10^4 \leq x_0, y_0 \leq 10^4$, $x_0^2 + y_0^2 > 0$, $1 \leq d, t \leq 10^4$)。

对于接下来 n 行，第 i 行输入两个整数 x_i 和 y_i ($-10^4 \leq x_i, y_i \leq 10^4$)，表示凸多边形第 i 个顶点的坐标。

所有 n 个顶点按逆时针顺序给出，并且任意三个顶点不共线。另外保证该形状与以 $(0, 0)$ 为中心、半径为 d 的圆没有公共点。也就是说，不存在一个点既在凸多边形的内部或边界上，同时又在圆的内部或边界上。

Output

输出一行一个实数，表示目标在前 t 单位时间内所遭受的伤害。

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 1 0 1 1 1 2 2 1 2 2	1.000000000000
3 1 0 1 2 1 2 2 1 2 2	1.570796326795
3 1 0 1 10000 1 2 2 1 2 2	2500.707752257475

Note

下图同时展示了各个样例数据的初始状态。

