

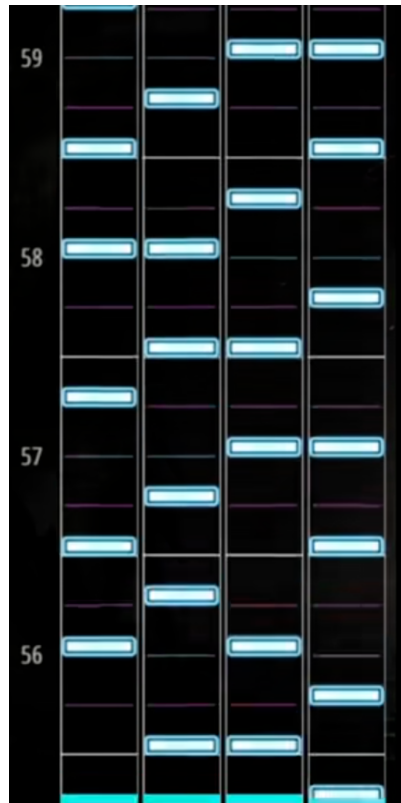
AC算法社算法个人赛暨ICPC网络赛选拔赛

【选手须知】

- 本场比赛采用 **ACM 赛制**。每道题目仅有“通过”与“未通过”两种状态，不设部分分。
- 比赛排名首先按通过题目总数降序排列；若通过题数相同，则按总罚时升序排列。每道 AC 题目的罚时 = 解题用时 + 该题 AC 前的错误提交次数 × 20 分钟（未 AC 的题目不计入总罚时）。
- 比赛在 PTA 平台进行，并全程使用 **OMS 监考系统**，选手须提前配置好监考环境。
- 本场比赛允许携带纸质材料（如纸质代码模板、图书、笔记等）。全程严禁查阅任何电子文档、检索互联网、使用 AI 辅助工具或与他人交流。
- 本场比赛共 11 道题目，**难度并不按顺序排列**，请选手合理规划做题顺序。
- 比赛开始后，选手可通过 PTA 在线评测系统提交代码。请合理安排时间，注意保存代码，并在截止前完成提交。
- 选手提交的程序必须能够对给定输入输出正确结果。评测使用的数据与题面样例不一定相同，程序必须是通用解法，不能针对样例特判。
- 选手提交的代码不得依赖图形界面、Win32 API、硬件操作或其他与操作系统强相关的接口。
- C++ 选手：允许使用 STL，所有依赖的头文件都必须在源文件中显式包含。
- Java 选手：主类名必须为 Main，且不得包含 package 包名声明；如使用类库，请将相应的 import 语句与程序正文一并提交。
- Python 选手：所有依赖的标准库模块都必须在源文件中显式 import，请不要依赖本地额外安装的第三方模块。
- 所有源码应提交为单文件形式。提交前请再次确认代码版本、语言与编译器配置是否正确。
- 违规处理：**比赛结束后将统一进行代码查重检测**，任何形式的抄袭、代考或违反监考规定的行为，均取消比赛成绩。

A. 切换

题目描述：



如图所示，在 4K 音游（共四个轨道，一个轨道为一列）中，切换（Stream）是一种基础键型。其判定规则非常简单：

同一轨道上不能出现连续的按键。即对于任意一条轨道，不能有相邻两行都出现按键。

现在给定 T 段铺面，请你分别判断每一段是不是一个合法的 Stream。

输入格式：

第一行一个整数 T ($1 \leq T \leq 100$)，表示测试数据组数。

接下来依次给出每组数据。每组数据的格式如下：

- 第一行一个整数 n ($2 \leq n \leq 10^6$)，表示该段铺面的行数。
- 接下来 n 行，每行 4 个整数（0 或 1），空格分隔，依次表示轨道 1, 2, 3, 4 是否有键。1 表示有键，0 表示无键。

保证所有测试数据的 n 之和不超过 10^6 。

输出格式：

对于每组数据，输出一行。如果该段铺面是合法的 Stream 键型，输出 YES，否则输出 NO（区分大小写）。

样例输入：

```
4
8
0 1 0 1
1 0 1 0
0 1 0 0
0 0 1 1
1 0 0 0
0 1 0 1
0 0 1 0
1 1 0 0
4
1 0 0 0
1 0 0 0
1 1 0 0
0 0 1 1
3
0 1 0 0
0 0 1 0
0 1 0 0
2
1 1 1 1
1 1 1 1
```

样例输出：

```
YES
NO
YES
NO
```

说明/提示：

- **第 1 组** ($n = 8$)：每条轨道上均未出现连续两行有键的情况，合法，输出 YES。

样例 1

1				
2				
3				
4				
5				
6				
7				
8				
	1	2	3	4

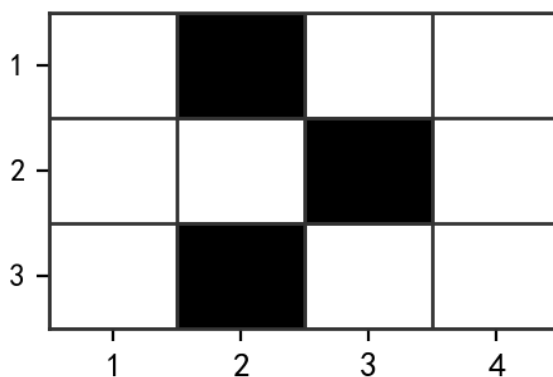
- **第 2 组** ($n = 4$)：轨道 1 在第 1、2、3 行连续有键，不合法，输出 NO。

样例 2

1				
2				
3				
4				
	1	2	3	4

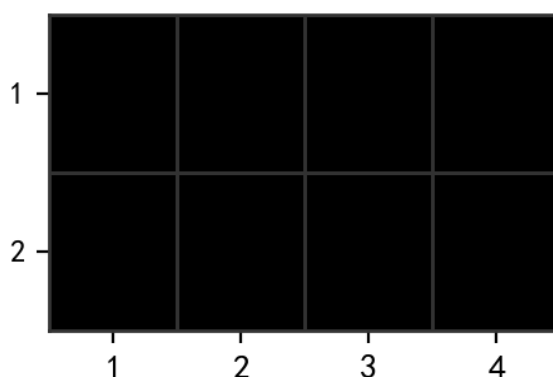
- **第 3 组** ($n = 3$)：轨道 2 在第 1 和第 3 行有键，但并非相邻行，合法，输出 YES。

样例 3



- **第 4 组** ($n = 2$)：所有 4 条轨道在第 1 行和第 2 行均有键，即每条轨道都出现了连续按键，不合法，输出 NO。

样例 4

**提示：**

本题输入数据量较大，请注意输入输出效率，避免因 I/O 耗时过长导致 TLE：

- **C++ 选手**：建议在 main 函数开头解除 I/O 同步，并使用 '\n' 替代 std::endl：

```
ios::sync_with_stdio(false);
cin.tie(nullptr);
#define endl '\n'
```

- **Java 选手**：请勿使用 Scanner 和 System.out.println，建议使用 BufferedReader 与 StringTokenizer 提取输入，并在结束时使用 PrintWriter 统一输出。
- **Python 选手**：提交时**务必选择 PyPy3 解释器**，并使用 sys.stdin.readline 替代默认的 input()，或使用 sys.stdin.read().split() 一次性读取。

B. 因子重排

题目描述：

小陶给了你一个长度为 n 的正整数数组 a 。

定义 $f(a)$ 表示 a 的所有子数组中，元素乘积能被 6 整除的子数组的数量。

更具体地说，对于每对下标 l 和 r ，满足 $1 \leq l \leq r \leq n$ ，考虑子数组 $a_l, a_{l+1}, \dots, a_r$ 。如果该子数组的元素乘积能被 6 整除，则计入 $f(a)$ 。

例如，若 $a = [1, 6, 2]$ ，则乘积能被 6 整除的子数组为 $[6]$ 、 $[1, 6]$ 、 $[6, 2]$ 以及 $[1, 6, 2]$ ，因此 $f(a) = 4$ 。

你的任务是重新排列数组 a 的元素，使得 $f(a)$ 最小。如果有多种方式满足条件，你可以输出任意一种。

注意：当数组 b 可以通过从数组 a 的开头和末尾各删除零个或若干个元素得到时， b 被称为 a 的子数组。

输入格式：

输入第一行为一个整数 T ($1 \leq T \leq 10^4$) ——表示测试用例的数量。

每个测试用例的第一行为一个整数 n ($1 \leq n \leq 2 \cdot 10^5$) ——数组的长度。

接下来一行输入 n 个整数 $a_1, a_2, \dots, a_n$ ($1 \leq a_i \leq 10^9$) ——数组的元素。

保证所有测试用例中 n 的总和不超过 $2 \cdot 10^5$ 。

输出格式：

对于每组测试数据，输出重新排列后的数组 a ，使 $f(a)$ 最小。如果有多种答案，可以输出任意一种。

样例输入：

```
5
6
12 7 9 4 18 5
4
3 6 2 8
7
1 10 15 20 3 6 9
5
11 14 21 2 5
3
6 6 6
```

样例输出：

```
12 18 4 7 5 9
2 8 3 6
6 10 20 1 15 3 9
21 5 11 2 14
6 6 6
```

说明/提示：

在第一个测试用例中，一种最优的排列是 $a = [12, 18, 4, 7, 5, 9]$ 。其中乘积能被 6 整除的子数组为：

- $[12]$
- $[18]$
- $[12, 18]$
- $[18, 4]$
- $[12, 18, 4]$
- $[18, 4, 7]$
- $[12, 18, 4, 7]$
- $[18, 4, 7, 5]$
- $[4, 7, 5, 9]$
- $[12, 18, 4, 7, 5]$
- $[18, 4, 7, 5, 9]$
- $[12, 18, 4, 7, 5, 9]$

因此 $f(a) = 12$ 。可以证明不存在更优的排列使得 $f(a)$ 更小。

C. 绝对排序

题目描述：

给定一个包含 n 个整数的数组 a 。如果 $a_1 \leq a_2 \leq \dots \leq a_n$ ，则称数组 a 是有序的。

你可以进行如下操作恰好一次，使数组 a 变为有序：

- 选择一个整数 x ，然后对每个 $i \in [1, n]$ ，将 a_i 替换为 $|a_i - x|$ 。

请你找出任意一个能使数组有序的 x 的值，或者报告不存在这样的 x 。

输入格式：

第一行包含一个整数 T ($1 \leq T \leq 2 \cdot 10^4$)，表示测试用例的数量。

每个测试用例包含两行。第一行包含一个整数 n ($2 \leq n \leq 2 \cdot 10^5$)。第二行包含 n 个整数 $a_1, a_2, \dots, a_n$ ($1 \leq a_i \leq 10^8$)。

输入的额外约束：所有测试用例中 n 的总和不超过 $2 \cdot 10^5$ 。

输出格式：

对于每个测试用例，输出一个整数 x ($0 \leq x \leq 10^9$)，使得经过操作后数组有序。如果不存在这样的 x ，输出 -1 。如果有多个满足条件的 x ，输出其中任意一个即可。

样例输入：

```
8
5
5 3 3 3 5
4
5 3 4 5
8
1 2 3 4 5 6 7 8
6
10 5 4 3 2 1
3
3 3 1
3
42 43 42
2
100000000 99999999
6
29613295 52036613 75100585 78027446 81409090 73215
```

样例输出：

```
4
-1
0
42
2
-1
100000000
40741153
```

说明/提示：

在第一个测试用例中，选择 $x = 4$ 后，数组变为 $[1, 1, 1, 1, 1]$ 。

在第三个测试用例中，选择 $x = 0$ 后，数组变为 $[1, 2, 3, 4, 5, 6, 7, 8]$ 。

在第四个测试用例中，选择 $x = 42$ 后，数组变为 $[32, 37, 38, 39, 40, 41]$ 。

D. 石子游戏

题目描述：

Mike 和 Joe 正在玩石子，Mike 先手。他们有 n 堆大小为 $a_1, a_2, \dots, a_n$ 的石子，堆呈圆形排列。

玩家从第一堆开始，顺时针依次从一堆中取出一些正数的石头。如果一个玩家在回合中从第 i 堆取石头，另一个玩家在下一轮从 $((i \bmod n) + 1)$ 堆取石头。

如果玩家在回合中无法取走任何石头（因为堆是空的），他就输了。

假设 Mike 和 Joe 都采取最优策略，那么谁会赢？

输入格式：

每个测试点包含多组数据。

第一行包含数据组数 T ($1 \leq T \leq 1000$)。

每组数据的第一行包含一个整数 n ($1 \leq 50$) 表示堆的数量。

第二行包含 $a_1, a_2, \dots, a_n$ ($1 \leq a_i \leq 10^9$) 表示每个堆的大小。

输出格式：

对于每组数据，输出获胜者，Mike 或 Joe。

样例输入：

```
2
1
37
2
100 100
```

样例输出：

```
Mike
Joe
```

E. 山间采茶

题目背景：

晚秋时节，云山雾绕。小陶准备入山采摘一些野生茶株，赶在初雪到来前煮一壶清茶。

山间不同位置生长着不同的茶树，越靠近山顶，路途越崎岖。

题目描述：

山上共有 N ($1 \leq N \leq 100$) 种野生茶树，第 i 种茶树产出的茶叶属性如下：

- 重量为 w_i ($1 \leq w_i \leq W$)；
- 浸泡后能提供的香气值为 v_i ($1 \leq v_i \leq 100$)；
- 生长在海拔 L_i ($1 \leq L_i \leq 10^9$) 的山腰处（需要小陶攀登的高度至少达到 L_i 才能采摘到）。

小陶随身背着的竹筐最大承重为 W ($1 \leq W \leq 100$)。他希望采摘到的茶叶总香气值至少达到 V ($1 \leq V \leq 1000$)。

秋日山路陡峭，为了安全与省力，小陶希望尽量降低自己需要攀登的最大海拔高度。换句话说，若选中的茶叶组合中，最高采摘海拔为 K ，小陶希望这个 K 尽可能小。

请你帮小陶计算：在竹筐承重不超过 W 且总香气值至少为 V 的前提下，需要攀登的最大海拔 K 的最小值是多少？如果无论攀登多高都无法满足要求，输出 -1 。

输入格式：

第一行包含三个正整数 N, W, V ，依次表示茶树种类数、竹筐最大承重、目标总香气值。

接下来的 N 行，每行包含三个正整数 w_i, v_i, L_i ，依次表示第 i 种茶叶的重量、香气值和生长海拔。

输出格式：

输出一个整数，表示满足条件的最小最大海拔 K 。若无解，输出 -1 。

样例输入：

```
3 10 15
5 10 300
4 8 200
6 12 500
```

样例输出：

300

说明/提示：

- 若小陶最多攀登到海拔 200，只能采到第 2 种茶叶（ $w = 4, v = 8$ ），香气值只有 8，未达到 15。
- 若小陶攀登到海拔 300，可以采到第 1 和第 2 种茶叶（总重量 $5 + 4 = 9 \leq 10$ ，总香气值 $10 + 8 = 18 \geq 15$ ），满足要求。
- 因此所需的最少攀登海拔为 300。

F. 特殊字符串

题目描述：

小陶正在玩一个关于字符串 s 的游戏，字符串长度为 n 。初始时， s 只包含小写拉丁字母。

每一回合，小陶可以选择一对整数 (i, j) ，满足：

- $1 \leq i < j \leq n$;
- $s_i = s_j \neq *$;
- 对于所有 $i < k < j$ ，有 $s_k = *$ 。

如果不存在这样的 i, j ，则游戏结束。否则，小陶令 $s_i := *$ 且 $s_j := *$ 。当游戏结束后，只有当 s 的每一个字符都等于 $*$ 时，小陶才获胜。请判断是否存在一种操作方式，使得小陶能够获胜。

注意： $*$ 表示 ASCII 字符 42。

输入格式：

第一行为一个整数 T ($1 \leq T \leq 100$)，表示测试用例的数量。

每个测试用例的第一行为一个整数 n ($1 \leq n \leq 5000$)，表示字符串的长度。

每个测试用例的第二行为一个仅由小写拉丁字母组成的字符串 s 。

所有测试用例中 n 的总和不超过 5000。

输出格式：

对于每个测试用例，输出一行答案。如果小陶能够获胜，输出 "YES"（不带引号）；否则，输出 "NO"。

样例输入：

```
6
1
a
6
llmllm
6
uwuuwu
6
byebye
6
oooioi
12
siixxsevvnn
```

样例输出：

```
NO
YES
YES
NO
NO
YES
```

说明/提示：

在第一个测试用例中，可以证明小陶无法获胜。

在第二个测试用例中，小陶可以按以下方式获胜： $llmllm \rightarrow llm**m \rightarrow **m**m \rightarrow ****$ 。

在第三个测试用例中，小陶可以按以下方式获胜： $uwuuwu \rightarrow uw**wu \rightarrow u****u \rightarrow ****$ 。

G. 传送门

题目描述：

给你一个长度为 n 的排列 p 。在位置 x 和 y ($x < y$) 各有一个传送门。

位置 i 的传送门最初位于数组第 i 个和第 $(i + 1)$ 个元素之间。特别地，如果 $i = 0$ ，传送门位于第一个元素之前；如果 $i = n$ ，传送门位于最后一个元素之后。

你可以无限次执行以下两种操作中的一种：

1. 将某个传送门左侧紧邻的元素移除，并将其插入到另一个传送门右侧紧邻处。
2. 将某个传送门右侧紧邻的元素移除，并将其插入到另一个传送门左侧紧邻处。

令 $\circ$ 表示传送门。例如，若 $p = [3, \circ, 2, 4, \circ, 1]$ ：

- 分别在左、右传送门使用操作 1，得到数组 $[\circ, 2, 4, \circ, 3, 1]$ 和 $[3, \circ, 4, 2, \circ, 1]$ 。
- 分别在左、右传送门使用操作 2，得到数组 $[3, \circ, 4, 2, \circ, 1]$ 和 $[3, 1, \circ, 2, 4, \circ]$ 。

请找出你能够通过这些操作得到的排列中字典序最小的一个。注意，传送门在字典序比较中没有影响。

- 长度为 n 的排列是一个长度为 n 的数组，包含 1 到 n 的每个整数各一次。
- 如果存在下标 i 使得对于所有 $1 \leq j < i$ 都有 $a_j = b_j$ ，且 $a_i < b_i$ ，则排列 a 字典序小于排列 b 。

输入格式：

第一行包含一个整数 t ($1 \leq t \leq 2 \cdot 10^4$)，表示测试用例个数。

每个测试用例的第一行包含三个整数 n 、 x 、 y ($1 \leq n \leq 2 \cdot 10^5$ ， $0 \leq x < y \leq n$)。

每个测试用例的第二行包含 n 个整数 $p_1, p_2, \dots, p_n$ ，表示一个长度为 n 的排列。

所有测试用例中的 n 之和不超过 $2 \cdot 10^5$ 。

输出格式：

对于每个测试用例，输出一行 n 个整数，表示你可以得到的字典序最小的排列。

样例输入：

```
4
4 0 4
3 1 4 2
3 1 2
3 2 1
5 1 3
1 3 5 2 4
2 0 1
1 2
```

样例输出：

```
1 4 2 3
2 3 1
1 2 3 5 4
1 2
```

说明/提示：

令 $\emptyset$ 表示传送门。

在第一个测试用例中，数组为 $[\emptyset, 3, 1, 4, 2, \emptyset]$ 。在左侧传送门使用操作 2 得到 $[\emptyset, 1, 4, 2, 3, \emptyset]$ ，这是能得到的字典序最小的排列。



The diagram shows the sequence of numbers 3, 1, 4, 2. A thick vertical red bar is positioned to the left of the number 3, and another thick vertical red bar is positioned to the right of the number 2. This represents the left and right portals in the problem context.

上述为描述中的操作动画。

在第二个测试用例中，数组为 $[3, \emptyset, 2, \emptyset, 1]$ ，在左侧传送门使用操作 1 得到 $[\emptyset, 2, \emptyset, 3, 1]$ ，这是能得到的字典序最小的排列。

在第四个测试用例中，最优方案是不进行任何操作。

H. 雷达安装

题目描述：

假设海岸线是一条无限延伸的直线。它的一侧是陆地，另一侧是海洋。每一座小岛是在海面上的一个点。雷达必须安装在陆地上（包括海岸线），并且每个雷达都有相同的扫描范围 d 。你的任务是建立尽量少的雷达站，使所有小岛都在扫描范围之内。

数据使用笛卡尔坐标系，定义海岸线为 x 轴。在 x 轴上方为海洋，下方为陆地。

输入格式：

第一行包括 2 个整数 n ($n \leq 1000$) 和 d ($d \leq 2 \times 10^4$)， n 是岛屿数目， d 是雷达扫描范围。

接下来 n 行，每行两个整数，为岛屿坐标 x 和 y ($|x_i| \leq 2 \times 10^6$, $0 \leq y_i \leq 2 \times 10^4$)。

输出格式：

一个整数表示最少需要的雷达数目，若不可能覆盖所有岛屿，输出 -1。

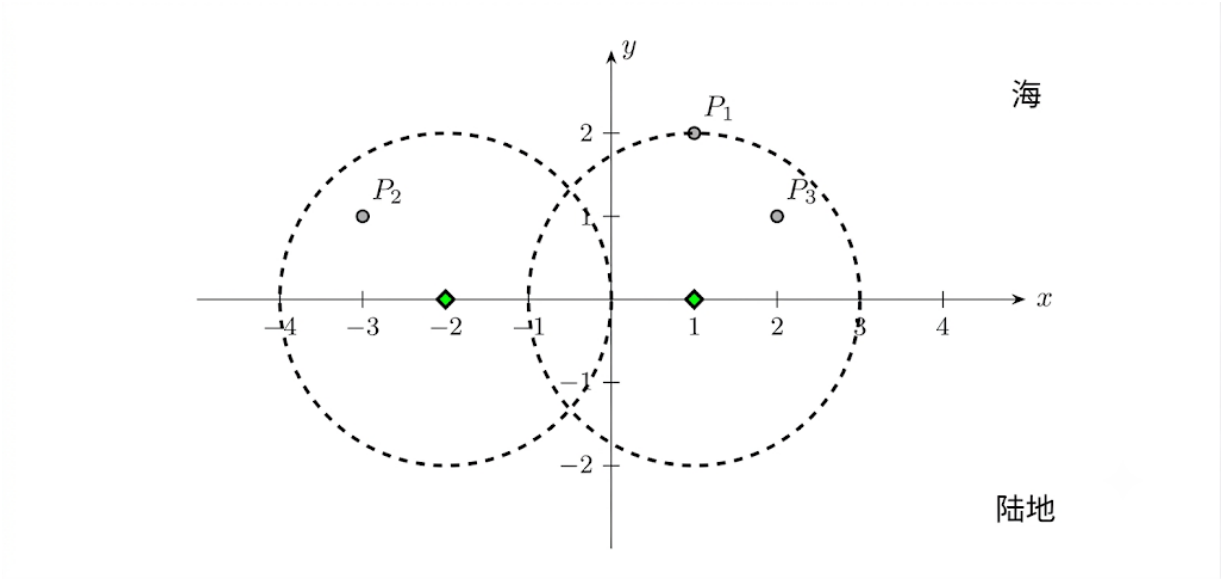
样例输入：

```
3 2
1 2
-3 1
2 1
```

样例输出：

```
2
```

说明/提示:



I. 胜负关系

题目描述：

我们考虑由 n 个实体和 m 个关系组成的集合 G 。为简单起见，这些 n 实体的标记从 1 到 n 。

在 G 上的排序 R 被定义为一个 n 阶排列 $R = r_1, r_2, \dots, r_n$ ($1 \leq r_i \leq n$)。

在 G 上的胜关系被定义为有序对 $\langle u, v \rangle$ ($1 \leq u, v \leq n, u \neq v$)，表示实体 u 在与实体 v 的竞争中胜。

我们说排名 R 满足获胜关系 $\langle u, v \rangle$ ，当且仅当存在 $1 \leq i < j \leq n$ ，使得 $r_i = u$ 和 $r_j = v$ 。

给定实体数 n 和胜出关系数 m ，要求构建最少数量的排序，使得每个胜出关系在所构建的排序中至少有一个排序得到满足。

输入描述：

第一行包含两个正整数 n, m ($2 \leq n \leq 10^6, 1 \leq m \leq 10^6$)，分别代表实体数和获胜关系数。

接下来的 m 行，每行包含两个正整数 u, v ($1 \leq u, v \leq n, u \neq v$)，代表一个获胜关系 $\langle u, v \rangle$ 。

输出描述：

第一行包含一个正整数 k ，代表所需的最小排序数。

接下来的 k 行中每一行都包含 n 个整数 $r_{i1}, r_{i2}, \dots, r_{in}$ ($1 \leq r_{ij} \leq n$)，代表第 i 次排名 $R_i = r_{i1}, r_{i2}, \dots, r_{in}$ 。

样例输入：

```
6 4
1 4
2 3
5 6
5 3
```

样例输出：

```
1
2 5 1 6 3 4
```

说明/提示:

- 对于 C++ 选手，本题请使用 g++ 编译器。

J. 公因对

题目描述：

小陶拿到了一个长度为 n 的数组 $a_1, a_2, \dots, a_n$ 。他想从中选出两个不同位置的元素，使得它们的最大公约数大于 1。

请你帮他找出这样一对元素，或者告诉他不存在这样的数对。

测试数据均由以下伪代码生成：

Algorithm 1 函数 f 用于生成测试数据

```

1: function  $f(l, r)$ 
2:   if  $l = r$  then
3:      $a_l \leftarrow \text{RandomInt}(1, 10^9)$ 
4:     return
5:   end if
6:    $mid \leftarrow \lfloor (l + r) / 2 \rfloor$ 
7:    $f(l, mid)$ 
8:    $f(mid + 1, r)$ 
9: end function

```

初始调用为 $n \leftarrow \text{RandomInt}(1, 2 \times 10^5)$ ，随后调用 $f(1, n)$ 。

其中 $\text{RandomInt}(L, R)$ 表示从闭区间 $[L, R]$ 中等概率随机选取一个整数，每次调用相互独立。

输入格式：

第一行包含一个整数 T ($1 \leq T \leq 10^4$)，表示测试数据的组数。

每组测试数据的第一行包含一个整数 n ($1 \leq n \leq 2 \times 10^5$)，表示数组长度。

第二行包含 n 个整数 $a_1, a_2, \dots, a_n$ ($1 \leq a_i \leq 10^9$)，表示数组元素。

保证单个测试文件中所有 n 之和不超过 2×10^5 。

输出格式：

对于每组测试数据，如果不存在满足条件的两个元素，输出一行 **-1**；否则，输出一行两个整数，表示选中的两个元素的值。如果存在多解，输出任意一解即可。

样例输入：

```

2
5
101770993 7791101 39239295 480041099 460611962
5
981262894 280102212 13579679 112334424 10169419

```

样例输出：

-1

981262894 280102212

K. 月落星沉

题目描述：

夜空中有 n 颗星辰，编号为 $1 \sim n$ 。部分星辰之间存在连接的星光，每条星光连接两颗星辰，双向可达。

随着寅时到来，将有 k 颗星辰按照既定顺序接连隐没。当一颗星辰隐没时，与该星辰相连的所有星光都会随之消散。

如果若干颗仍在闪耀的星辰能够通过尚未消散的星光彼此到达，则它们属于同一片星域。一片星域的大小定义为其中包含的闪耀星辰数量。

你需要计算在每次星辰隐没之后，夜空中最大星域的大小（即包含星辰数最多的星域所含有的星辰数量）。若此时没有仍在闪耀的星辰，则输出 0。

输入格式：

第一行包含三个整数 n, m, k ($1 \leq n \leq 10^5, 1 \leq m \leq 2 \cdot 10^5, 1 \leq k \leq n$)，分别表示星辰数量、星光数量以及隐没的星辰数量。

接下来 m 行，每行包含两个整数 u, v ($1 \leq u, v \leq n$)，表示星辰 u 与星辰 v 之间存在一条星光。输入中可能存在重边或自环。

接下来 k 行，每行包含一个整数 x_i ($1 \leq x_i \leq n$)，表示第 i 颗隐没的星辰编号。保证 $x_1, x_2, \dots, x_k$ 互不相同。

输出格式：

输出共 k 行。第 i 行输出一个整数，表示第 i 颗星辰隐没后，夜空中最大星域的大小。

样例输入：

```
4 3 2
1 2
2 3
3 4
2
3
```

样例输出：

```
2
1
```

说明/提示：

初始星光连接情况为链状：1 — 2 — 3 — 4。

- **第 1 次隐没（星辰 2）**：剩余闪耀星辰为 {1, 3, 4}。星辰 3 与 4 仍有星光相连，组成大小为 2 的星域；星辰 1 孤立，大小为 1。最大星域大小为 2。
- **第 2 次隐没（星辰 3）**：剩余闪耀星辰为 {1, 4}，彼此无星光相连，每个星辰自成星域，最大星域大小为 1。