“规律未来杯” 2025 年广东工业大学 ACM 新生程序设计竞赛（决赛）

广东工业大学 ACM 集训队

2025 年 11 月 30 日

试题列表

A	极差最大的区间
B	魔法棋盘
C	区间乘
D	环球旅行商
E	质数变化
F	宇宙射线风暴
G	万能矩阵
H	能量汇聚
I	比较大小
J	协会的实验
K	线段覆盖
L	气球采购
M	敌人的敌人
N	最大化仿射变换

共 14 题

Problem A. 极差最大的区间

输入文件: standard input
输出文件: standard output

给定一个长度为 n 的整数序列 A，下标从 1 到 n。

你需要找到一个**连续的子区间** $[l, r]$（其中 $1 \le l \le r \le n$），使得该子区间内的元素的**极差最大**。

一个区间的极差定义为该区间内所有元素中的最大值与最小值之差。

即：对于子区间 $A[l..r]$，其极差为 $\max(A[l..r]) - \min(A[l..r])$。

你需要输出**任意一个**满足最大极差的连续子区间 $[l, r]$ 的起始和结束下标 l 和 r，以及这个最大极差值。

输入格式

第一行包含一个整数 n $(1 \le n \le 100)$，表示序列的长度。

第二行包含 n 个整数 $A_1, A_2, \ldots, A_n$ $(0 \le A_i \le 100)$，表示序列的元素。

输出格式

输出一行三个整数：

- 第一个整数表示拥有最大极差的连续子区间的起始下标 l；
- 第二个整数表示该连续子区间的结束下标 r；
- 第三个整数表示这个最大极差值。

如果存在多个子区间拥有相同的最大极差，你可以输出其中任意一个。

样例

standard input	standard output
4 2 3 4 1	2 4 3
5 1 5 2 9 3	1 4 8

Problem B. 魔法棋盘

输入文件: standard input
输出文件: standard output

小 P 得到了一块神秘的魔法棋盘。棋盘是一个大小为 $\mathbf{1000} \times \mathbf{1000}$的网格图，格子黑白相间，其中左上角 $(1,1)$ 为**白色**。换句话说，格子行与列的奇偶性相同时为白色，否则为黑色。

每次启动魔法棋盘，都需要完成一个挑战：在棋盘上选出一个**连通块**，其中包含恰好 a 个白格子和 b 个黑格子。

当两个格子共享一条边时，视为它们相邻；如果一个格子集合中任意两个格子都可以通过若干次相邻移动互达，则该集合为一个**连通块**。

小 P 发现有些情况下根本无法找到满足条件的连通块，因此他希望你判断：

- 给定的 a 和 b 是否可行；
- 若可行，请给出任意一组满足条件的构造方案。

输入格式

第一行输入一个整数 t $(1 \le t \le 10^4)$，表示测试数据的组数。

接下来 t 行，每行输入两个整数 a, b $(0 \le a, b \le 2 \times 10^5$ 且 $1 \le a + b \le 2 \times 10^5)$，分别表示连通块的白格子数量和黑格子数量。

保证所有测试用例中 $a + b$ 的总和不超过 10^6。

输出格式

对于每组数据：

- 若存在合法构造，输出一行 `YES`，随后输出 $a + b$ 行，每行两个整数 x, y $(1 \le x, y \le 1000)$，表示选择的格子坐标。
- 若不存在合法构造，输出一行 `NO`。

若存在多种可行方案，以任意顺序输出任意一种可行方案即可。

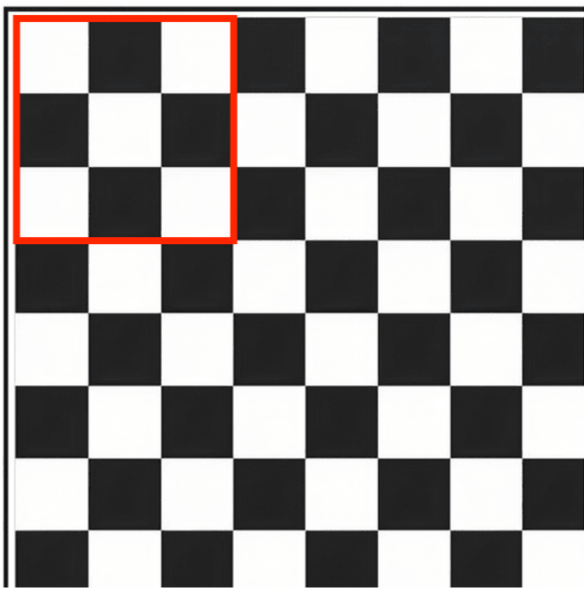
你可以以任意大小写输出 `YES` 和 `NO`（例如，字符串 `yEs`、`yes`、`Yes` 和 `YES` 将被识别为肯定的回答）。

样例

standard input	standard output
3 1 1 0 1000 5 4	YES 1 1 1 2 NO YES 1 1 1 2 1 3 2 1 2 2 2 3 3 1 3 2 3 3

提示说明

对于第三个测试用例，下图中红色边界内的格子构成了满足条件的连通块：



红色边界内的连通块共 5 个白色格子，4 个黑色格子，符合要求。

Problem C. 区间乘

输入文件: standard input
输出文件: standard output

给定一个长度为 n 的正整数序列 a，其中元素编号从 1 到 n。

你需要处理 q 次查询。每一次查询会给定一个目标正整数 x，你需要判断是否存在一个**连续的子区间** $[l, r]$（其中 $1 \le l \le r \le n$），使得该子区间内所有元素的乘积等于 x。即：

$$\prod_{i=l}^{r} a_i = a_l \times a_{l+1} \times \cdots \times a_r = x$$

如果存在满足条件的连续子区间，输出 YES，否则输出 NO。

输入格式

第一行输入一个整数 T（$1 \le T \le 10^4$），表示测试数据的组数。

对于每组测试数据：

- 第一行输入两个整数 n 和 q（$1 \le n, q \le 10^5$），分别表示序列的长度和查询次数。
- 第二行输入 n 个正整数 $a_1, a_2, \ldots, a_n$（$1 \le a_i \le 10^9$），表示序列 a 中的元素。
- 接下来 q 行，每行输入一个正整数 x（$1 \le x \le 10^9$），表示一次查询的目标乘积。

保证所有测试数据的 n 之和不超过 2×10^5，所有测试数据的 q 之和不超过 2×10^5。

输出格式

对于每组测试数据中的每个查询，输出一行：

- 如果存在满足条件的连续子区间，输出 YES。
- 如果不存在满足条件的连续子区间，输出 NO。

你可以以任意大小写输出 YES 和 NO（例如，字符串 yEs、yes、Yes 和 YES 将被识别为肯定的回答）。

样例

standard input	standard output
2 5 3 2 3 6 1 4 6 12 1 7 1 2 1 1 1 1 1 3 6	YES NO YES YES

提示说明

对于第一组测试数据，序列为 $[2, 3, 6, 1, 4]$ ：

1. 查询 $x = 6$：区间 $[1, 2]$ 乘积为 $2 \times 3 = 6$，存在，输出 YES。
2. 查询 $x = 12$：不存在连续子区间的乘积等于 12，输出 NO。
3. 查询 $x = 1$：区间 $[4, 4]$ 乘积为 1，存在，输出 YES。

Problem D. 环球旅行商

输入文件: standard input
输出文件: standard output

小 P 是一位环球旅行商，计划在世界各地采购物资。

世界地图可以看作一个 $n+2$ 行 m 列的网格，行编号从 0 到 $n+1$ ，其中第 0 行代表北极，第 $n+1$ 行代表南极，而列编号从 1 到 m 。

由于地球是圆的，网格在**列方向是循环的**：从第 1 列向左移动会到达第 m 列，从第 m 列向右移动会到达第 1 列。

小 P 需要访问 k 个物资点。他的旅行规则如下：

- 从北极点（第 0 行的任意位置）出发，最终到达南极点（第 $n+1$ 行的任意位置）。
- 旅途中**不能向北走**（即行号不能减少）。从位置 (x, y) 出发，他只能移动一步到：
  - $(x+1, y)$（向南）
  - $(x, y-1)$（向西，如果 $y=1$ 则到达 (x, m)）
  - $(x, y+1)$（向东，如果 $y=m$ 则到达 $(x, 1)$）

现在，小 P 想知道：从北极点出发，经过 **所有 k 个物资点**（每个点至少访问一次），最后到达南极点，最少需要移动多少步？

输入格式

第一行一个整数 t $(1 \le t \le 10^4)$，表示测试数据组数。

对于每组测试数据：

- 第一行三个整数 n, m, k $(1 \le n, m \le 10^9, 1 \le k \le 10^6)$，分别表示地图的行数、列数和物资点的数量。
- 接下来 k 行，每行两个整数 x_i, y_i $(1 \le x_i \le n, 1 \le y_i \le m)$，表示物资点坐标，保证坐标不重复。

保证所有测试数据的 k 之和不超过 10^6。

输出格式

对于每组数据，输出一行一个整数，表示最少步数。

样例

standard input	standard output
2 1 1 1 1 1 10 10 6 1 1 1 10 5 10 5 9 5 2 10 1	2 17

Problem E. 质数变化

输入文件: `standard input`
输出文件: `standard output`

小 P 非常喜欢质数（Prime Number），不仅因为质数在数论中地位重要，还因为它的英文首字母恰好是 P！

这天，小 P 手中有两个**四位以内的质数** A 和 B。他突发奇想，想通过若干次"数字变换"把 A 变成 B。

每次变换操作定义如下：

- 将当前数字视为一个**四位数**（不足四位时，在前面补前导零，例如 13 写作 0013）；
- 选择其中**任意一位数字**，将其修改为 0 到 9 中的任意一个数字；
- 变换后的数**必须仍然是一个质数**，且**可以有前导零**（即变换后的数必须是 $[2, 9999]$ 范围内的质数）。

例如：1031 可以变为 0031 ，但不能变成 1032（因为 1032 不是质数），也不能变成 21031（因为 21031 不是四位以内的数）。

现在，小 P 想知道：**最少需要多少次变换操作，才能把 A 变成 B**？如果无法做到，则输出 -1。

输入格式

第一行包含一个整数 T（$1 \le T \le 100$），表示测试数据组数。

接下来 T 行，每行包含两个整数 A 和 B（$2 \le A, B < 10000$），保证 A 和 B 均为质数。

输出格式

对于每组测试数据，输出一行：

- 如果能在有限次数内从 A 变换到 B，输出所需的最少变换次数；
- 否则，输出 `-1`。

样例

standard input	standard output
3 2 3 1031 37 2 8849	1 2 7

提示说明

对于第二个样例，一种可行的变化方案是：$1031 \to 0031 \to 0037$

Problem F. 宇宙射线风暴

输入文件: standard input
输出文件: standard output

在 X-99 星系的防御体系中，核心是一个覆盖了巨大连续区域的能量矩阵。我们可以将这个区域视为一个二维平面上的正方形闭区间，其横纵坐标范围均为 $[0, N]$。

监测系统侦测到 Q 条宇宙射线正在穿越该区域，第 i 条射线携带特定的能量 E_i 。

这些宇宙射线表现为贯穿整个坐标系的无限长直线，且轨迹极其规律，只分为两种类型：

1. **45 度射线**：轨迹满足 $y = x + b$
2. **135 度射线**：轨迹满足 $y = -x + c$

射线能量具有叠加性，当多条射线经过同一个点时，该点处的能量强度为这些射线能量的**总和**。

防御指挥官需要寻找能量最强的点，以确定护盾应该如何部署。你需要计算出在区域 $[0, N] \times [0, N]$ 范围内（含边界），任意单点所能承受的最大能量叠加值是多少。

请注意，由于能量场是连续的，**最高能量点不一定位于整数坐标上，也不一定被多条宇宙射线穿过，还可能落在区域的边界线上**，只要这个点 (x, y) 满足 $0 \le x, y \le N$，即被视为有效目标点。

输入格式

第一行包含一个整数 T $(1 \le T \le 1000)$，表示测试数据的组数。

对于每组测试数据：

- 第一行包含两个整数 N 和 Q $(1 \le N \le 10^5, 1 \le Q \le 10^5)$，分别表示区域的边长和射线的数量。
- 接下来 Q 行，第 i 行包含四个整数 t_i, x_i, y_i, E_i，用于描述第 i 条射线：
  - t_i $(1 \le t_i \le 2)$：第 i 条射线的类型：$t_i = 1$ 表示 45 度射线，$t_i = 2$ 表示 135 度射线。
  - x_i, y_i $(0 \le x_i, y_i \le N)$：第 i 条射线经过的区域内任意一点的坐标 (x, y)，用于确定这条射线。
  - E_i $(1 \le E_i \le 10^9)$：第 i 条射线的能量值。

保证所有测试数据 N 的总和，Q 的总和均不超过 10^5。

输出格式

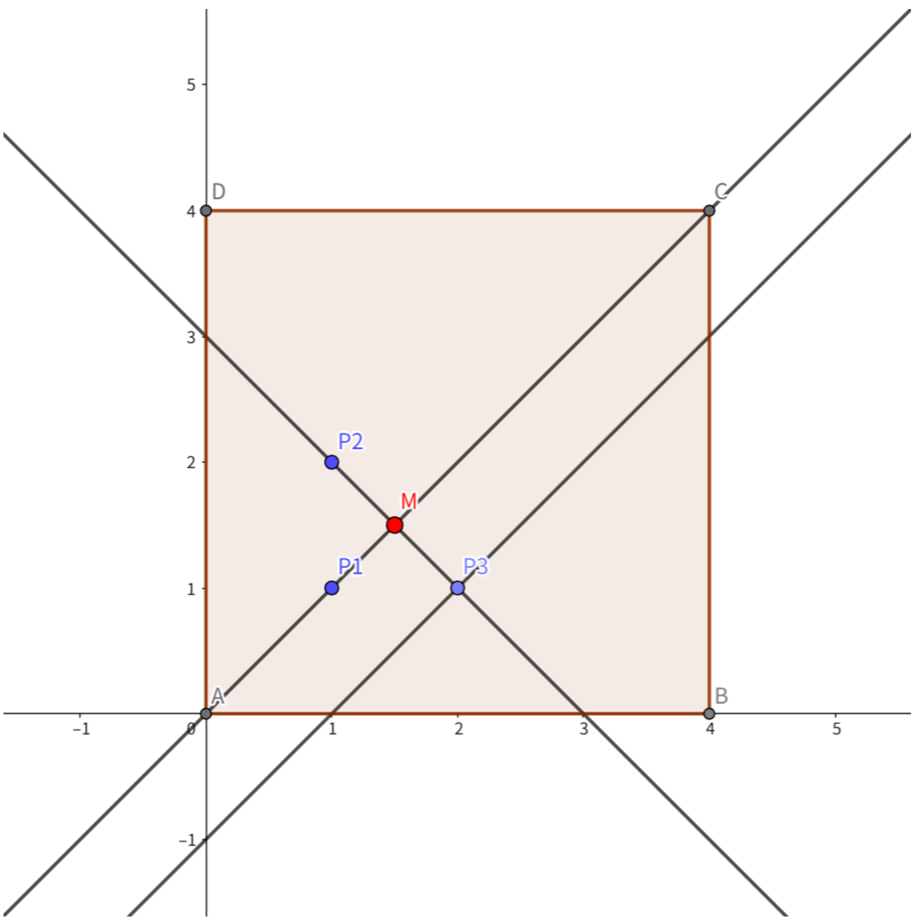
对于每组测试数据，输出一行一个整数，表示区域内（包括非整数坐标和边界）单点受到的最大能量总和。

样例

standard input	standard output
2 4 3 1 1 1 10 2 1 2 20 1 2 1 5 3 2 1 0 0 5 2 3 3 8	30 13

提示说明

对于第一组测试样例，所有射线情况如下图所示：



其中 M 点能量总和最大，为 $10 + 20 = 30$ 。

Problem G. 万能矩阵

输入文件: `standard input`
输出文件: `standard output`

给定一个正整数 n，你需要构造一个 $2n \times 2n$ 的整数矩阵 M，其元素 $M_{i,j}$ 满足 $0 \le M_{i,j} \le n^4$（$1 \le i, j \le 2n$）。

构造的目标是使得区间 $[1, n^4]$ 内的每一个整数 K 都可以表示为该矩阵 M 的某个**连续子矩阵**的所有元素之和。

一个连续子矩阵由其左上角坐标 (r_1, c_1) 和右下角坐标 (r_2, c_2) 确定，其中 $1 \le r_1 \le r_2 \le 2n$ 且 $1 \le c_1 \le c_2 \le 2n$。

即，对于任意 $K \in [1, n^4]$，存在 $1 \le r_1 \le r_2 \le 2n$ 和 $1 \le c_1 \le c_2 \le 2n$ 使得：

$$K = \sum_{i=r_1}^{r_2} \sum_{j=c_1}^{c_2} M_{i,j}$$

如果存在任何满足条件的 $2n \times 2n$ 的矩阵 M ，输出这个矩阵 M ，否则输出 “IMPOSSIBLE” 。

输入格式

第一行输入一个整数 t（$1 \le t \le 50$），表示测试用例数量。

每个测试用例包括一行一个正整数 n $(1 \le n \le 100)$。

保证所有测试用例的 n^4 之和不超过 10^8 。

输出格式

对每个测试用例，如果存在任意一个满足条件的矩阵 M ，输出 $2n$ 行，每行 $2n$ 个整数，以空格分隔。这些整数构成了你构造的 $2n \times 2n$ 矩阵 M ，其元素 $M_{i,j}$ 满足 $0 \le M_{i,j} \le n^4$（$1 \le i, j \le 2n$）；否则输出“IMPOSSIBLE”。

样例

standard input	standard output
2 1 2	1 0 0 1 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16

Problem H. 能量汇聚

输入文件: standard input
输出文件: standard output

在一个一维数轴上，有 $n+1$ 个位置，编号从 0 到 n。而 1 到 n 的每个位置 i 都包含一定的能量值 E_i。位置 0 作为起点，不含能量。

你的目标是从起点 0 跳跃到终点 n，并在此过程中汇聚最大的总能量。

当你的位置小于 n 时，可以进行跳跃，每次跳跃可以向前移动 1 步或 2 步。即，如果你当前在位置 i，你可以跳到位置 $i+1$ 或 $i+2$。当然，不能超过终点 n，也就是你位于位置 $n-1$ 时无法进行 2 步的跳跃。

在跳跃过程中，存在一个**限制条件**：你不能连续两次跳 1 步。

- 如果你从 $i-1$ 跳到 i（1 步），那么下一次跳跃必须从 i 跳到 $i+2$（2 步）。
- 如果你从 $i-2$ 跳到 i（2 步），那么下一次跳跃可以从 i 跳到 $i+1$（1 步）或 $i+2$（2 步）。

每次跳到位置 i（$1 \le i \le n$）时，你将获得该位置的能量 E_i。你需要求出从位置 0 到位置 n 能够汇聚到的 **最大总能量**。

输入格式

第一行包含一个整数 n $(1 \le n \le 10^5)$，表示终点的位置编号。

第二行包含 n 个整数 $E_1, E_2, \ldots, E_n$ $(1 \le E_i \le 1000)$，依次表示位置 1 到位置 n 的能量值。

输出格式

输出一行一个整数，表示从 0 跳到 n 能够汇聚到的最大总能量。

样例

standard input	standard output
5 5 10 10 10 1	21
2 10 1	1

提示说明

在样例1中，目标是跳到位置 5，能量值 $E = [1, 10, 10, 10, 1]$。

最优跳跃路径为 $0 \to 2 \to 4 \to 5$ ，获得的能量如下：

- $0 \to 2$（2 步）。获得能量 $E_2 = 10$。
- $2 \to 4$（2 步）。获得能量 $E_4 = 10$。
- $4 \to 5$（1 步）。获得能量 $E_5 = 1$。

总能量：$10 + 10 + 1 = 21$。

注意： 路径 $0 \to 1 \to 2 \to \dots$ 中，不能连续跳 1 步，例如 $0 \to 1$（1 步）后，不能接 $1 \to 2$（1 步）。

可以证明，在所有可行的路径中，最大总能量为 21。

Problem I. 比较大小

输入文件: `standard input`
输出文件: `standard output`

本题是一道交互题，选手的程序可以向交互器进行若干询问，最后进行回答。具体交互方式请参考下方交互协议部分。

现在有 n 个**未知**的非负整数 $X_1, X_2, \ldots, X_n$ （$1 \le X_i \le 10^{18}$），同时，给定一个质数 p。

你需要通过不超过 $10n$ 次询问，确定这 n 个数中的最大值 $M = \max(X_1, X_2, \ldots, X_n)$，并**输出** M **对** p **取模后的结果，注意** M **不是取模后的最大值**。

每次询问的方式如下：

- 你可以选择一个整数 c（$1 \le c \le n$），并指定 c 个**互不相同的**下标 $i_1, i_2, \ldots, i_c$（$1 \le i_j \le n$）。
- 交互器会计算这 c 个数中的最小值 $\min(X_{i_1}, X_{i_2}, \ldots, X_{i_c})$，并返回这个**最小值对** p **取模后的结果，注意不是取模后的最小值**。

请注意，交互器是非自适应的。换句话说，所有的 X_i 和 p 都是在任何查询之前就固定好的。

输入格式

每个测试包含多个测试用例。

第一行输入一个整数 t（$1 \le t \le 50$），表示测试用例数量。

每个测试用例包括一行两个整数 n 和 p：

- n（$1 \le n \le 100$），表示未知数的数量。
- p（$2 \le p \le 10^9 + 7$，且 p 为质数）。

保证所有测试用例的 n 之和不超过 100 。

交互协议

每个测试用例的程序交互应从读取整数 n 和 p 开始。

要进行查询，请按以下格式输出一行：

- $?\ c\ i_1\ i_2\ \ldots\ i_c$

其中 c（$1 \le c \le n$）是你选择的数字个数，$i_1, i_2, \ldots, i_c$ 是这 c 个数的下标，你需要保证查询的所有下标互不相同。

然后你会收到一个整数 R（$0 \le R < p$），表示 $\min(X_{i_1}, X_{i_2}, \ldots, X_{i_c}) \pmod p$ 的结果。

要报告最终答案（即最大值 M 对 p 取模后的结果），请按以下格式输出一行：

- $!\ M$

请注意，总查询次数不能超过 $10n$ 次，但报告最终答案不计入查询次数限制。

在输出答案后，您的程序应继续处理下一个测试用例，若无更多测试用例则终止程序。

需要注意的是，每次输出完毕后，你都需要打印换行符并**刷新缓冲区**！你可以使用如下语句来刷新缓冲区：

- 对于 C/C++：`fflush(stdout)`
- 对于 C++：`std::cout << std::flush`
- 对于 Java：`System.out.flush()`
- 对于 Python：`stdout.flush()`

对于 C++ 语言，在输出换行时如果你使用 `std::endl` 而不是 ‘`\n`’，也可以自动刷新缓冲区。

样例

standard input	standard output
2	
1 3	
	? 1 1
2	
	! 2
3 2	
	? 3 1 2 3
0	
	? 1 1
1	
	? 1 3
1	
	! 1

Problem J. 协会的实验

输入文件: `standard input`
输出文件: `standard output`

ACM — *Association for Counting Massively*（大规模计数协会）是一个致力于研究各类场景下高效大规模计数方法的学术组织。

近期，协会启动了一项关于动态字符串演化过程中的子序列计数的实验。作为研究员，你将模拟一个动态字符串的构建过程：**初始时字符串为空**；随后，系统会按照给定的操作序列，在字符串的**前端**或**后端**逐个插入字符。你的任务是：在每次插入操作完成后，实时输出当前字符串中子序列 “acm” 的出现次数，结果对 10^9+7 取模。

一个字符串的子序列，是指通过删除若干（可以为零个）字符而得到的序列，且剩余字符的相对顺序不变。例如，字符串 “aaccmm” 中存在多个 “acm” 子序列（如取第 1、3、5 位字符）。

输入格式

第一行包含一个整数 t（$1 \le t \le 1000$），表示测试用例数量。

对于每个测试用例：

输入一行字符串 s（$1 \le |s| \le 10^6$），表示**操作序列**。字符串 s 仅由字符 a, c, m, A, C, M 组成，其含义为：

- 小写字母（a, c, m）：在字符串**前端**插入对应的小写字符；
- 大写字母（A, C, M）：在字符串**后端**插入对应的小写字符（即 A 插入 a，C 插入 c，M 插入 m）。

保证所有测试用例的 $|s|$ 之和不超过 10^6

输出格式

对每个测试用例，输出一行 $|s|$ 个整数，用空格分隔：第 i 个整数为执行完前 i 次操作后，当前字符串中子序列 “acm” 的数量对 10^9+7 取模的结果。

样例

standard input	standard output
2 ACMca AACCMM	0 0 1 1 3 0 0 0 0 4 8

Problem K. 线段覆盖

输入文件: `standard input`
输出文件: `standard output`

按递增顺序给出数轴上的 n 个互不相同的点，它们的坐标分别为 $x_1, x_2, \ldots, x_n$。

你需要使用不超过 k 个线段（线段可以退化为点，长度为 0）来覆盖所有给定的 n 个点。每个线段 $[a, b]$ 的长度定义为 $|b - a|$。

请注意，线段 $[a, b]$ 覆盖点 x 意味着 $a \le x \le b$ 或 $b \le x \le a$。

你需要找到一种覆盖方案，使得所有使用的线段的**总长度之和最短**。

对于每一个 k $(1 \le k \le n)$，请输出满足条件的最短总线段长度。

输入格式

第一行包含一个整数 n $(1 \le n \le 2 \times 10^5)$，表示点的数量。

第二行包含 n 个整数 $x_1, x_2, \ldots, x_n$ $(1 \le x_i \le 10^9)$，表示这 n 个点的坐标。

保证给定的 n 个坐标互不相同，且单调递增，即 $x_1 < x_2 < \cdots < x_n$。

输出格式

输出一行 n 个整数，第 i 个整数表示 $k = i$ 时的答案。

样例

standard input	standard output
3 1 2 3	2 1 0
1 114514	0
5 1 2 3 9 10	9 3 2 1 0

Problem L. 气球采购

输入文件: standard input
输出文件: standard output

新生赛正在火热筹备中，好心的出题组为每种题目都准备了颜色独一无二的气球！

在比赛的前几天，出题组发现，准备的题目气球可能不够了！

比赛共有 n 道题目，参赛总人数为 m ，对于第 i 道题目，出题组的预期过题率为 $\frac{p_i}{q_i}$ $(p_i, q_i \neq 0)$。

因此，预计需要的气球数为：

$$\text{need}_i = \left\lceil m \cdot \frac{p_i}{q_i} \right\rceil$$

目前第 i 题已有 w_i 个气球。若 $w_i < \text{need}_i$，需额外购买 $\text{need}_i - w_i$ 个；否则无需购买。

出题组想知道，总共需额外购买多少个气球。

输入格式

第一行两个整数 n（$1 \leq n \leq 14$）和 m（$1 \leq m \leq 200$）。

接下来 n 行：每行三个整数 p_i, q_i, w_i（$1 \leq p_i \leq q_i \leq 100$，$0 \leq w_i \leq m$）。

输出格式

一个整数，表示总共需额外购买的气球数量。

样例

standard input	standard output
3 10 1 1 0 50 100 10 1 11 0	11
10 141 1 47 0 73 97 10 2 71 20 1 71 30 18 47 40 6 77 50 7 38 60 15 92 70 1 3 80 1 71 90	114

提示说明

在第一个样例中：

- 第一题需要 $\lceil 10 \cdot \frac{1}{1} \rceil = 10$ 只气球，需要额外购买 10 个气球；
- 第二题需要 $\lceil 10 \cdot \frac{50}{100} \rceil = 5$ 只气球，无需额外购买气球；
- 第三题需要 $\lceil 10 \cdot \frac{1}{11} \rceil = 1$ 只气球，需要额外购买 1 个气球。

总共需要额外购买 $10 + 1 = 11$ 个气球。

向上取整可通过以下方式计算，以避免浮点误差：

$$\left\lceil \frac{a}{b} \right\rceil = \left\lfloor \frac{a+b-1}{b} \right\rfloor \quad (a, b > 0)$$

Problem M. 敌人的敌人

输入文件: `standard input`
输出文件: `standard output`

在这个世界中，有 n 个国家，编号从 1 到 n。

国家之间的敌对关系可以用一个图来表示：每个国家是图中的一个**节点**，每条**边**表示两个国家之间存在敌对关系，即 u 和 v 之间存在边意味着国家 u 和国家 v **互为敌人**。

巧合的是，敌对关系形成的图恰好是一棵树（恰好有 $n-1$ 条边的无环的连通图）。

根据"敌人的敌人就是朋友"的规则：

- 如果存在三个不同的国家 x, y, z，满足 y 是 x 的敌人，z 是 y 的敌人，那么 z 就是 x 的朋友。

注意：朋友关系和敌对关系均不能传递（例如，如果 A 是 B 的朋友，B 是 C 的朋友，不能推断 A 是 C 的朋友）。

请你找出朋友数量最多的国家（输出任意一个即可），以及它的朋友数量。

输入格式

第一行输入一个整数 t（$1 \le t \le 10^4$），表示测试用例数量。

对于每个测试用例：

- 第一行一个整数 n $(2 \le n \le 2 \times 10^5)$，表示国家的数量。
- 接下来 $n-1$ 行，每行两个整数 u, v $(1 \le u, v \le n)$，表示国家 u 和国家 v 之间存在敌对关系。

保证输入的图构成一棵树，所有测试数据的 n 之和不超过 2×10^5 。

输出格式

对于每个测试用例，输出一行两个整数：

- 第一个整数表示朋友最多的国家编号；
- 第二个整数表示该国家的朋友数量。

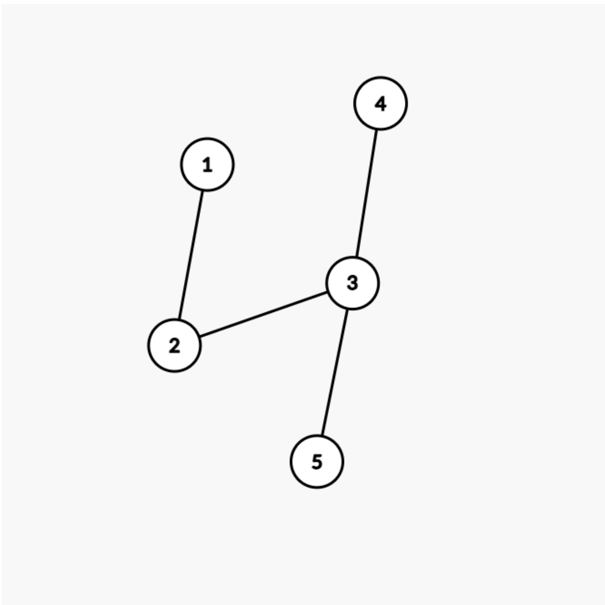
如果存在多个国家拥有相同的朋友，你可以输出其中任意一个朋友最多的国家编号。

样例

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

提示说明

对于第二个测试用例，敌对关系形成的图如下图所示：



- 1 号国家的敌人有 2，朋友有 3 ，共 1 个朋友。
- 2 号国家的敌人有 1, 3，朋友有 4, 5 ，共 2 个朋友。
- 3 号国家的敌人有 2, 4, 5，朋友有 1 ，共 1 个朋友。
- 4 号国家的敌人有 3，朋友有 2, 5 ，共 2 个朋友。
- 5 号国家的敌人有 3，朋友有 2, 4 ，共 2 个朋友。

综上，2, 4, 5 号国家都有相同数量的朋友，朋友数量为 2。

Problem N. 最大化仿射变换

输入文件: `standard input`
输出文件: `standard output`

有一个变量 x ，初始时 $x = 0$ 。

给定 n 个操作，第 i 个操作定义了一个仿射变换，形式为：

$$x := a_i x + b_i$$

其中 := 为赋值号，a_i 和 b_i 均为非负整数。

你需要将这 n 个操作安排一个执行顺序，并依次执行。目标是使得所有操作执行完毕后，最终 x 的值达到**最大**。

由于最终 x 的值可能会很大，请输出这个最大值对 $10^9 + 7$ 取模后的结果，**注意不是取模后的最大值，而是对最大值取模**。

输入格式

第一行输入一个整数 t（$1 \le t \le 10^4$），表示测试用例数量。

每个测试用例格式如下：

- 第一行包含一个正整数 n（$1 \le n \le 10^6$），表示操作的总数。
- 接下来的 n 行，第 i 行包含两个非负整数 a_i, b_i （$0 \le a_i, b_i \le 10^9$），表示第 i 个操作的参数。

保证所有测试用例的 n 之和不超过 10^6 。

输出格式

输出一个整数，表示在所有可能的执行顺序中，最终 x 的最大值对 $10^9 + 7$ 取模后的结果。

样例

standard input	standard output
5 3 2 1 1 5 3 0 4 1 1 2 1 1 2 2 2 3 0 1 0 100 1 10 4 0 0 1 0 0 1 1 1 5 4 3 2 2 1 0 3 1 5 4	33 17 110 2 178