2024 icpc 国际大学生程序设计竞赛
全国邀请赛（昆明）

正式赛

2024 年 5 月 26 日

试题列表

A	两星级竞赛
B	金牌
C	阻止城堡 2
D	生成字符串
E	学而时习之
F	收集硬币
G	乐观向上
H	子数组
I	左移 2
J	冲向黄金城
K	排列
L	漫步野径
M	意大利美食

本试题册共 13 题，17 页。
如果您的试题册缺少页面，请立即通知志愿者。

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. 两星级竞赛

教育专家们出于某种原因，准备对 n 项竞赛进行评级。专家们已经决定了每项竞赛的评级结果，其中第 i 项竞赛被评为 s_i 星级竞赛。

据说每项竞赛都会依据 m 种属性进行评级，其中第 i 项竞赛的第 j 种属性记为 $p_{i,j}$，每种属性的取值范围从 0 到 k（含两端）。一项竞赛的分数是其所有 m 种属性的总和。也就是说，令 v_i 表示第 i 项竞赛的分数，我们有 $v_i = \sum_{j=1}^{m} p_{i,j}$。

如果一项星级更高的赛事有更高的分数，看起来会比较自然。专家们要求，对于任意两项竞赛 $1 \le i, j \le n$，若 $s_i > s_j$，则必须有 $v_i > v_j$。不幸的是，专家们忘了采集一些竞赛部分（甚至全部）属性的数据。作为专家们的助手，您被要求填充这些不存在的属性值，使得上述限制条件对任意两项竞赛都成立。

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（$2 \le n \le 4 \times 10^5$，$1 \le m \le 4 \times 10^5$，$n \times m \le 4 \times 10^5$，$1 \le k \le 10^9$）表示竞赛的数量，每项竞赛有几种属性，以及每种属性取值的上限。

对于接下来 n 行，第 i 行首先输入一个整数 s_i（$1 \le s_i \le 10^9$）表示第 i 项竞赛被评定的星级。接下来输入 m 个整数 $p_{i,1}, p_{i,2}, \cdots, p_{i,m}$（$-1 \le p_{i,j} \le k$）。若 $p_{i,j} = -1$ 则第 i 项竞赛的第 j 种属性值不存在，您需要填充该属性值；否则若 $p_{i,j} \ge 0$ 则第 i 项竞赛的第 j 种属性值已被给定，您不应该更改它。

保证所有数据 $n \times m$ 之和不超过 4×10^5。

Output

对于每组数据：

如果可以填充所有不存在的属性值并满足限制条件，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 $q_{i,1}, q_{i,2}, \cdots, q_{i,m}$（$0 \le q_{i,j} \le k$），表示完成填充之后的第 i 项竞赛的 m 种属性值。若 $p_{i,j} = -1$，那么 $q_{i,j}$ 就是您填充的值；否则若 $p_{i,j} \ge 0$，那么 $q_{i,j} = p_{i,j}$。如果有多种答案，您可以输出任意一种。

如果无法满足限制条件，仅需输出一行 `No`。

Example

standard input	standard output
5 3 4 5 5 1 3 -1 -1 2 -1 5 -1 5 3 3 -1 -1 4 2 3 10 10000 5 0 -1 1 10 10 10 2 3 10 10 1 2 3 100 4 5 6 2 3 10 100 1 2 3 10 4 5 6 2 3 10000 100 -1 -1 -1 1 -1 -1 -1	Yes 1 3 5 4 0 5 0 5 3 3 2 4 No Yes 1 2 3 4 5 6 No Yes 2024 5 26 11 45 14

Note

对于第二组样例数据，即使我们将唯一的 -1 填入最大的可能值 10，第一项竞赛的分数也只有 15 分，并不大于第二项竞赛的分数 30 分。

Problem B. 金牌

真是忙碌的一周！这个周末，将会有 n 场程序设竞赛同时进行。

每场竞赛将会从每 k 支参加该竞赛的队伍中颁发一枚金牌。也就是说，如果有 t 支队伍参加竞赛，将会颁发 $\lfloor \frac{t}{k} \rfloor$ 枚金牌，其中 $\lfloor x \rfloor$ 是不超过 x 的最大整数。目前第 i 场竞赛有 a_i 支队伍参加。

堡堡是一所大学的教练，该大学有 m 支队伍，并且他还没有决定每支队伍应该参加哪场竞赛。请帮助他为每支队伍分配一场竞赛，使得所有竞赛颁发的金牌总数最多。

Input

有多组测试数据。第一行输入一个整数 T（$1 \le T \le 100$）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（$1 \le n \le 100$，$1 \le k \le 10^9$），表示竞赛的数量和金牌的比例。

第二行输入 n 个整数 $a_1, a_2, \cdots, a_n$（$1 \le a_i \le 10^9$），其中 a_i 表示目前有几支队伍参加第 i 场竞赛。

第三行输入一个整数 m（$1 \le m \le 10^9$），表示您需要为多少支队伍安排竞赛。

Output

每组数据输出一行一个整数，表示所有竞赛最多一共颁发多少金牌。

Example

standard input	standard output
2 3 10 239 141 526 6 2 1 300 100 1000	91 1400

Note

对于第一组样例数据，派 2 支队伍去第 1 场竞赛，4 支队伍去第 3 场竞赛。金牌总数为 $\lfloor \frac{239+2}{10} \rfloor + \lfloor \frac{141+0}{10} \rfloor + \lfloor \frac{526+4}{10} \rfloor = 24 + 14 + 53 = 91$。

Problem C. 阻止城堡 2

一块有 10^9 行和 10^9 列的棋盘上放着 n 座城堡与 m 个障碍物。每座城堡或每个障碍物恰好占据一个格子，且被占据的格子两两不同。两座城堡可以互相攻击，若它们位于同一行或同一列，且它们之间没有障碍物或其它城堡。更正式地，令 (i,j) 表示位于第 i 行第 j 列的格子，位于 (i_1,j_1) 和 (i_2,j_2) 的两座城堡可以互相攻击，若以下条件中有一条成立：

- $i_1=i_2$，且对于所有 $\min(j_1,j_2)<j<\max(j_1,j_2)$，不存在位于 (i_1,j) 的障碍物或城堡。
- $j_1=j_2$，且对于所有 $\min(i_1,i_2)<i<\max(i_1,i_2)$，不存在位于 (i,j_1) 的障碍物或城堡。

您需要从棋盘上移除 k 个障碍物，但您不希望太多城堡可以互相攻击。从棋盘上移除恰好 k 个障碍物之后，最小化可以互相攻击的城堡对数。

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（$1\le n,m\le 10^5$，$1\le k\le m$）表示城堡的数量，障碍物的数量，以及您需要移除的障碍物的数量。

对于接下来 n 行，第 i 行输入两个整数 r_i 和 c_i（$1\le r_i,c_i\le 10^9$），表示第 i 座城堡位于第 r_i 行第 c_i 列。

对于接下来 m 行，第 i 行输入两个整数 r'_i 和 c'_i（$1\le r'_i,c'_i\le 10^9$），表示第 i 个障碍物位于第 r'_i 行第 c'_i 列。

保证被占据的格子两两不同。同时保证所有数据 n 之和与 m 之和均不超过 10^5。

Output

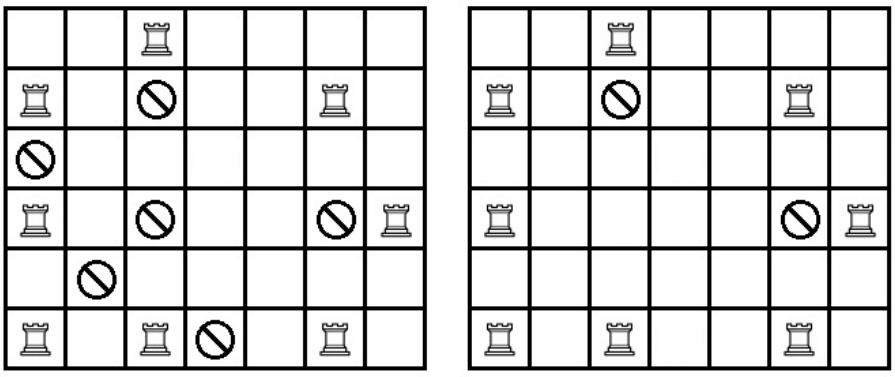
每组数据首先输出一行一个整数，表示恰好移除 k 个障碍物之后，最少有几对城堡可以互相攻击。接下来输出一行 k 个不同的由单个空格分隔的整数 $b_1,b_2,\cdots,b_k$（$1\le b_i\le m$），表示您移除的障碍物的编号。如果有多种合法答案，您可以输出任意一种。

Example

standard input	standard output
3 8 6 4 1 3 2 1 2 6 4 1 4 7 6 1 6 3 6 6 2 3 3 1 4 3 4 6 5 2 6 4 3 2 1 10 12 10 10 10 11 1 4 1 5 1 3 2 1 1 2 1 2 2 2 3	4 6 3 2 5 2 1 0 1 2

Note

对于第一组样例数据，左边的图片展示了原本的棋盘，右边的图片展示了移除 4 个障碍物之后的棋盘。在移除障碍物之后，可以互相攻击的城堡对有：第 2 和第 4 座城堡，第 4 和第 6 座城堡，第 6 和第 7 座城堡，第 7 和第 8 座城堡。



对于第三组样例数据，由于只有 1 座城堡，不存在可以互相攻击的城堡对。

Problem D. 生成字符串

给定一个长度为 n 的模板字符串 $S = s_1 s_2 \cdots s_n$。一个生成字符串指的是由模板字符串 S 的若干子串连接而成的字符串。更正式地，每个生成字符串 $T = f(k, \{l_i\}_{i=1}^{k}, \{r_i\}_{i=1}^{k})$ 可以被描述为一个正整数 k 以及 k 对整数 (l_i, r_i)，其中 $T = s[l_1 : r_1] + s[l_2 : r_2] + \cdots + s[l_k : r_k]$。这里我们用 $s[l : r]$ 表示子串 $s_l s_{l+1} \cdots s_r$，用 $+$ 表示字符串连接。

您需要维护一个由字符串构成的可重集合 $\mathbb{A}$，支持以下三种操作：

- `+` $k\ l_1\ r_1\ l_2\ r_2\ \cdots\ l_k\ r_k$：将 $f(k, \{l_i\}_{i=1}^{k}, \{r_i\}_{i=1}^{k})$ 加入可重集合 $\mathbb{A}$。
- `-` t：将第 t 次操作加入的字符串从可重集合 $\mathbb{A}$ 里删除。保证第 t 次操作是一次加入操作，且该字符串目前还没有被删除。
- `?` $k\ l_1\ r_1\ l_2\ r_2\ \cdots\ l_k\ r_k\ m\ u_1\ v_1\ u_2\ v_2\ \cdots u_m\ v_m$：求可重集合 $\mathbb{A}$ 里有几个字符串以 $f(k, \{l_i\}_{i=1}^{k}, \{r_i\}_{i=1}^{k})$ 为开头，且以 $f(m, \{u_i\}_{i=1}^{m}, \{v_i\}_{i=1}^{m})$ 为结尾。

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 q（$1 \le n, q \le 10^5$），表示 S 的长度和操作的数量。

第二行输入一个由小写英文字母构成的字符串 $s_1 s_2 \cdots s_n$，表示模板字符串。

对于接下来的 q 行，第 i 行输入一个操作，格式如上所述。保证 $1 \le l_i \le r_i \le n$，$1 \le u_i \le v_i \le n$。另外保证所有 `+` 类型的操作的 k 之和，加上所有 `?` 类型的操作的 k 之和，加上所有 `?` 类型的操作的 m 之和不超过 3×10^5。

Output

每次 `?` 类型的操作输出一行一个整数表示答案。

Example

standard input	standard output
8 7 abcaabbc + 3 1 3 2 4 3 8 + 2 1 4 1 8 + 1 2 4 ? 1 5 6 1 7 8 - 3 + 1 2 5 ? 1 2 3 1 5 5	2 1

Problem E. 学而时习之

给定长度为 n 的正整数序列 $a_1, a_2, \cdots, a_n$ 以及一个非负整数 k，您可以执行以下操作至多一次：选择两个整数 l 和 r 满足 $1 \le l \le r \le n$，之后对于每个 $l \le i \le r$，将 a_i 变为 $(a_i + k)$。

最大化整个序列的最大公因数。

称整数 g 是整个序列的公因数，若对于所有 $1 \le i \le n$ 都满足 a_i 能被 g 整除。

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（$1 \le n \le 3 \times 10^5$，$0 \le k \le 10^{18}$）。

第二行输入 n 个整数 $a_1, a_2, \cdots, a_n$（$1 \le a_i \le 10^{18}$）表示序列。

保证所有数据 n 之和不超过 3×10^5。

Output

每组数据输出一行一个整数，表示整个序列最大的最大公因数。

Example

standard input	standard output
2 6 2 5 3 13 8 10 555 3 0 3 6 9	5 3

Note

对于第一组样例数据，选择 $l = 2$ 以及 $r = 4$。序列会变为 $\{5, 5, 15, 10, 10, 555\}$。序列的最大公因数是 5。

Problem F. 收集硬币

有 10^9 个格子排成一行，从左到右编号从 1 到 10^9。两个机器人正在格子里巡逻。每个机器人的最大速度是 v 格每秒（v 是整数），表示如果机器人目前在格子 p，它在下一秒可以移动到任意一个格子 p'，只要满足 $1 \le p' \le n$ 且 $|p' - p| \le v$。

将有 n 枚硬币在格子中出现。第 i 枚硬币会在第 t_i 秒出现在格子 c_i。如果有一个机器人同时也位于那个格子，它就会把硬币捡起来。否则硬币会马上消失。

更正式地，在每一秒内，以下事件会依次发生：

- 每个机器人可以移动到距离不超过 v 的格子（也可以待在当前的格子里）。
- 硬币在格子中出现。
- 如果至少一个机器人和一枚硬币在同一个格子里，这枚硬币就会被收集。
- 所有未被收集的硬币消失。

您需要决定两个机器人在第 1 秒开始前的初始位置，并合理地移动它们，使得所有硬币都能被收集，并且 v 尽可能小。输出 v 的最小值。

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：

第一行输入一个整数 n（$1 \le n \le 10^6$）表示格子里将会出现的硬币数量。

对于接下来 n 行，第 i 行输入两个整数 t_i 和 c_i（$1 \le t_i, c_i \le 10^9$）表示第 i 枚硬币出现的时间以及第 i 枚硬币出现的位置。保证对于所有 $1 \le i < n$ 有 $t_i \le t_{i+1}$。同时保证对于所有 $i \ne j$ 有 $t_i \ne t_j$ 或 $c_i \ne c_j$。

保证所有数据 n 之和不超过 10^6。

Output

每组数据输出一行一个整数，表示机器人最大速度的最小值。如果不可能收集所有硬币，输出 `-1`。

Example

standard input	standard output
3 5 1 1 3 7 3 4 4 3 5 10 1 10 100 3 10 100 10 1000 10 10000	2 0 -1

Problem G. 乐观向上

求字典序最小的序列 $p_0, p_1, p_2 \cdots, p_{n-1}$，该序列是 $0, 1, 2, \cdots, (n-1)$ 的一个排列，且满足以下限制：对于所有 $0 \le i < n$，都有 $p_0 \oplus p_1 \oplus \cdots \oplus p_i > 0$。这里 $\oplus$ 是按位异或运算。

称一个长度为 n 的序列 $p_0, p_1, \cdots, p_{n-1}$ 的字典序小于另一个长度为 n 的序列 $q_0, q_1, \cdots, q_{n-1}$，若存在一个整数 $0 \le k < n$ 使得对于所有 $0 \le i < k$ 有 $p_i = q_i$，以及 $p_k < q_k$。

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（$1 \le n \le 10^6$）。

保证所有数据 n 之和不超过 10^6。

Output

每组数据输出一行 n 个由单个空格分隔的整数，表示字典序最小的且满足限制的排列。如果不存在这种排列，输出 `impossible`。

Example

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

Problem H. 子数组

给定长度为 n 的整数序列 $a_1, a_2, \cdots, a_n$，称一个连续子数组 $a_l, a_{l+1}, \cdots, a_r$ 是好的，若子数组里的最大元素在子数组里恰好出现了 k 次。对每个 $1 \le k \le n$ 计算好的子数组的数量。

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（$1 \le n \le 4 \times 10^5$）表示序列的长度。

第二行输入 n 个整数 $a_1, a_2, \cdots, a_n$（$1 \le a_i \le 10^9$）表示序列。

保证所有数据 n 之和不超过 4×10^5。

Output

令 c_i 表示 $k = i$ 时好的子数组的数量。为了减少输出的大小，每组数据只需要输出一行一个整数，表示 $\sum_{i=1}^{n}(i \times c_i^2)$ 对 998244353 取模的结果。

Example

standard input	standard output
3 11 1 1 2 1 2 2 3 3 2 3 1 3 2024 5 26 3 1000000000 1000000000 1000000000	2564 36 20

Note

令 $[l, r]$ 表示连续子数组 $a_l, a_{l+1}, \cdots, a_r$。对于第一组样例数据:

- $c_1 = 27$，$c_2 = 22$，$c_3 = 17$，而 $c_4, c_5, \cdots, c_{11}$ 均为 0。所以答案为 $(1 \times 27^2 + 2 \times 22^2 + 3 \times 17^2) \bmod 998244353 = 2564$。
- 当 $k = 1$ 时，一些好的子数组的例子有 $[3, 3]$（最大元素 2 出现了一次），$[4, 5]$（最大元素 2 出现了一次），以及 $[9, 10]$（最大元素 3 出现了一次）。
- 当 $k = 2$ 时，一些好的子数组的例子有 $[1, 2]$（最大元素 1 出现了两次），$[4, 6]$（最大元素 2 出现了两次），以及 $[6, 9]$（最大元素 3 出现了两次）。
- 当 $k = 3$ 时，一些好的子数组的例子有 $[3, 6]$（最大元素 2 出现了三次），$[2, 6]$（最大元素 2 出现了三次），以及 $[1, 11]$（最大元素 3 出现了三次）。

Problem I. 左移 2

给定一个由小写字母组成的字符串，称该字符串是美丽的，若字符串中每一对相邻的字符都不相同。例如，`icpc` 和 `kunming` 是美丽的，但 `hello` 不是，因为它的第 3 个和第 4 个字符相同。

给定由小写英文字母组成的，长度为 n 的字符串 $S = s_0 s_1 \cdots s_{n-1}$，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 $g(S, d)$ 表示将 $f(S, d)$ 变得美丽的最小操作次数。每次操作中，您可以将 $f(S, d)$ 中的任意一个字符改为任意小写字母。

找到一个非负整数 d 最小化 $g(S, d)$，并输出这个最小化的值。

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:

第一行输入一个仅由小写字母组成的字符串 $s_0 s_1 \cdots s_{n-1}$（$1 \le n \le 5 \times 10^5$）。

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，表示最小的 $g(S, d)$。

Example

standard input	standard output
3 abccbbbbd abcde x	2 0 0

Note

对于第一组样例数据，考虑 $d = 5$。有 $f(S, 5) =$ `bbbdabccb`。对于这个字符串，我们可以将它的第 2 个字符改成 `x`，并将它的第 8 个字符改成 `y`。这样字符串就会变成 `bxbdabcyb`，是一个美丽的字符串。

Problem J. 冲向黄金城

某个国家有 n 座城市以及 m 条连接城市的双向铁路。第 i 条铁路由第 c_i 家铁路公司运营，铁路的长度是 l_i。

您想要从城市 1 开始进行全国旅行。您已经为旅行购买了 k 张车票。第 i 张车票可以记为两个整数 a_i 和 b_i，表示如果您使用了这张车票，就可以一次性经过若干条均由公司 a_i 运营的，且总长度不超过 b_i 的铁路。即使您使用了车票，也可以选择待在当前城市。您同时只能使用一张车票，且每张车票只能使用一次。

由于决定车票的使用顺序太麻烦了，您打算直接按现有的顺序使用车票。更正式地，您将执行 k 次操作。在第 i 次操作中，您可以选择待在当前的城市 u；也可以选择一座不同的城市 v，满足城市 u 和 v 之间存在一条路径，且路径上的所有铁路均由公司 a_i 运营，且铁路总长不超过 b_i，然后移动到城市 v。

对于每座城市，判断在使用 k 张车票之后能否到达该城市。

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（$2 \le n \le 5 \times 10^5$，$1 \le m \le 5 \times 10^5$，$1 \le k \le 5 \times 10^5$）表示城市的数量，铁路的数量以及车票的数量。

对于接下来 m 行，第 i 行输入四个整数 u_i，v_i，c_i 和 l_i（$1 \le u_i, v_i \le n$，$u_i \ne v_i$，$1 \le c_i \le m$，$1 \le l_i \le 10^9$），表示第 i 条铁路连接了城市 u_i 和 v_i，该铁路由公司 c_i 运营，且铁路长度为 l_i。注意，可能有多条铁路连接同一对城市。

对于接下来 k 行，第 i 行输入两个整数 a_i 和 b_i（$1 \le a_i \le m$，$1 \le b_i \le 10^9$），表示如果您使用了第 i 张车票，就可以一次性经过若干条均由公司 a_i 运营的，且总长度不超过 b_i 的铁路。

保证所有数据 n 之和，m 之和与 k 之和均不超过 5×10^5。

Output

每组数据输出一行一个长度为 n 的字符串 $s_1 s_2 \cdots s_n$，其中每个字符要么是 0，要么是 1。如果您可以用 k 张车票从城市 1 到达城市 i，则 $s_i = 1$；否则 $s_i = 0$。

Example

standard input	standard output
2 5 6 4 1 2 1 30 2 3 1 50 2 5 5 50 3 4 6 10 2 4 5 30 2 5 1 40 1 70 6 100 5 40 1 30 3 1 1 2 3 1 10 1 100	11011 100

Note

对于第一组样例数据：

- 为了到达城市 4，您可以使用第 1 张车票从城市 1 移动到城市 2，然后在使用第 2 张车票的时候待在城市 2，然后使用第 3 张车票从城市 2 移动到城市 4，然后在使用第 4 张车票的时候待在城市 4。
- 为了到达城市 5，您可以使用第 1 张车票，经由第 1 条和第 6 条铁路从城市 1 移动到城市 5，然后在使用后续车票的时候待在城市 5。
- 由于您不能更改使用车票的顺序，您无法到达城市 3。

Problem K. 排列

这是一道交互题。

有一个隐藏的 n 的排列。请回忆：n 的排列指的是一个序列，从 1 到 n（含两端）的每个整数在序列中都恰好出现一次。Piggy 准备通过若干次询问找出这个隐藏的排列。

每次询问包含一个长度为 n 的整数序列（可以不是排列），每个元素的取值范围从 1 到 n（含两端）。每次询问之后，Piggy 会收到一个答案 x，表示他询问的序列中，有几个位置和隐藏的排列相同。例如，假设隐藏的排列是 $\{1,3,4,2,5\}$，Piggy 询问的序列是 $\{2,3,5,2,5\}$，那么他将收到 3 作为回答。

然而 Piggy 最近太忙了，所以他把问题交给了您。您需要在 6666 次询问内找出这个排列。

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（$1 \le n \le 10^3$），表示隐藏的排列的长度。

Interaction Protocol

要提出询问，请输出一行。首先输出一个 `0`，之后跟一个空格，然后输出 n 个范围在 1 到 n 之间的整数，整数之间由单个空格分隔。在清空输出缓冲区之后，您的程序需要读入一个整数 x，表示对您的询问的回答。

要猜想排列，请输出一行。首先输出一个 `1`，之后跟一个空格，然后输出一个 n 的排列，整数之间由单个空格分隔。在清空输出缓冲区之后，您的程序应该立即退出。

请注意，每组测试数据的答案都是预先确定的。也就是说，裁判程序并不是适应性的。还请注意，猜想排列不算一次询问。

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

Note

请注意，如果您收到了 Time Limit Exceeded 的评测结果，有可能是因为您的询问不合法，或您的询问次数超出了限制。

Problem L. 漫步野径

堡堡正在一块无穷大的二维平面上散步。对于平面上每个满足 x 和 y 均是整数的点 (x, y)，都有一条双向小径连接点 (x, y) 和 $(x + 1, y)$，还有另一条双向小径连接点 (x, y) 和 $(x, y + 1)$。另外，还有 n 条额外的双向小径，其中第 i 条连接点 (x_i, y_i) 和 $(x_i + 1, y_i + 1)$。

堡堡只能沿着小径移动。令 $f(x, y)$ 表示堡堡从点 $(0, 0)$ 出发到达点 (x, y) 最少需要经过几条小径。给定两个整数 p 和 q，计算

$$\sum_{x=0}^{p}\sum_{y=0}^{q} f(x, y)$$

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，p 和 q（$1 \le n \le 10^6$，$0 \le p, q \le 10^6$）。它们的含义如上所述。

对于接下来的 n 行，第 i 行输入两个整数 x_i 和 y_i（$0 \le x_i, y_i \le 10^6$）表示第 i 条额外小径连接点 (x_i, y_i) 和 $(x_i + 1, y_i + 1)$。保证对于所有 $i \ne j$ 有 $x_i \ne x_j$ 或 $y_i \ne y_j$。

保证所有数据 n 之和不超过 10^6。请注意，p 与 q 之和没有限制。

Output

每组数据输出一行一个整数表示答案。

Example

standard input	standard output
2 3 2 4 1 1 0 2 0 0 1 100 100 1000 1000	34 1020100

Problem M. 意大利美食

堡堡为您准备了一份披萨！这个披萨是一个凸多边形，每条饼边里都有芝士夹心。但这些夹心边很脆弱，导致切披萨的时候只能经过多边形的顶点，而不能从边的中间切开。不幸的是，披萨上有一片您肯定不喜欢的，巨大的圆形菠萝片。

求沿着直线切一刀之后，可以获得的最大的没有菠萝的披萨的面积。称一块披萨上没有菠萝，当且仅当菠萝没有任何部分严格位于披萨块内。也就是说，菠萝和披萨的相交面积为 0。

Input

有多个测试数据。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（$3 \le n \le 10^5$）表示披萨的顶点数。

第二行输入三个整数 x_c，y_c 和 r（$-10^9 \le x_c, y_c \le 10^9$，$1 \le r \le 10^9$）表示菠萝的中心坐标和半径。

对于接下来的 n 行，第 i 行输入两个整数 x_i 和 y_i（$-10^9 \le x_i, y_i \le 10^9$）表示第 i 个顶点的坐标。顶点按逆时针顺序列出。保证任意两点不重合。但可能有三个点在同一直线上。

保证菠萝的任何部分都不会超出披萨的边界。另外保证所有数据 n 之和不超过 10^5。

Output

每组测试数据输出一行一个整数，表示最大的没有菠萝的披萨的面积乘以 2。可以证明这个值始终是一个整数。如果无法切出没有菠萝的披萨块，输出 0。

Example

standard input	standard output
3 5 1 1 1 0 0 1 0 5 0 3 3 0 5 6 2 4 1 2 0 4 0 6 3 4 6 2 6 0 3 4 3 3 1 3 0 6 3 3 6 0 3	5 24 0

Note

样例数据如下图所示。