Problem A. Your Shine Your Be!

Input file: standard input
Output file: standard output

给定一个长度为 n 的非负整数序列 $a_1, a_2, \ldots, a_n$。

现有 q 次查询，每次查询给定一个区间 $[l, r]$。对于每次查询，你需要从区间 $[l, r]$ 所包含的元素构成的多重集 V 中挑选一个大小最大的子集 M，使得 M 中任意两个元素的权值互不相同。随后，你可以从选出的集合 M 中再任选一个子集 T（允许为空集）。设 $|M|$ 为集合 M 的大小，X_T 为集合 T 中所有元素权值的按位异或和（空集的异或和规定为 0），求出 $|M| \oplus X_T$ 的最小值。

本题强制在线。每次查询给出的 l', r' 是加密后的结果。记上一次查询的答案为 $lastans$（初始时 $lastans = 0$）。实际的查询区间端点 l, r 可通过如下方式解密得到:

$$l = (l' \oplus lastans) \bmod n + 1$$

$$r = (r' \oplus lastans) \bmod n + 1$$

解密后，若 $l > r$，你需要交换 l 和 r 的值。

Input

第一行包含两个整数 n, q $(1 \le n, q \le 5 \cdot 10^5)$。

第二行包含 n 个整数 $a_1, a_2, \ldots, a_n$ $(0 \le a_i \le n)$。

接下来 q 行，每行两个整数 l', r' $(0 \le l', r' \le 2 \cdot 10^9)$，表示加密后的查询区间端点。

Output

对于每组询问，输出一行一个数字表示结果。

Example

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

Note

初始 lastans $= 0$。

- 第一次查询：解密得 l $= 1$, r $= 1$。区间内元素构成的集合 S $= 4$，|S| $= 1$。选空集 （异或和为 0），答案为 1 0 $= 1$。更新 lastans $= 1$。
- 第二次查询：解密得 l $= 4$, r $= 6$。区间内元素构成的集合 S $= 0, 6$，|S| $= 2$。选空集 （异或和为 0），答案为 2 0 $= 2$。更新 lastans $= 2$。
- 第三次查询：解密得 l $= 5$, r $= 2$，交换端点得 [2, 5]。区间内元素构成的集合 S $= 0, 4, 6$，|S| $= 3$。选子集 4, 6（异或和为 2），答案为 3 2 $= 1$。更新 lastans $= 1$。

本题输入输出数据量较大，建议使用较快的输入输出方式。C++ 选手建议使用 `scanf/printf`，或在使用 `cin/cout` 时取消同步流:

```
std::ios::sync_with_stdio(false);
std::cin.tie(nullptr);
```

Problem B. Bus博弈

Input file: standard input
Output file: standard output

在一条直线上有 n 个连续的公交车座位，编号为 1 到 n。由于年久失修，其中有 m 个座位是损坏的。

Alice 和 Bob 轮流安排游客入座，Alice 先手。每次操作，当前玩家必须选择一个未损坏且当前为空的座位 i 安排游客入座，且必须满足以下条件：

所有满足 $|j-i| \le k$ 且 $j \neq i$ 的座位 j 都是空的（即与 i 距离不超过 k 的其他座位均未被占用，注意：损坏的座位不影响此距离判断，可以跨越坏座位计算距离）。

游客坐下后，座位 i 即被占用。当当前玩家无法找到任何合法的空座位安排游客时，游戏直接结束。Alice 希望最终入座的游客尽可能多，Bob 希望尽可能少。两人都采取最优策略。

请以字符串的形式输出最终公交车上可能的座位状态。

Input

第一行包含三个整数 n、k、m ($1 \le n \le 20$, $1 \le k \le 20$, $0 \le m \le n$)，分别代表总座位数、距离限制条件以及损坏的座位数量。

第二行包含 m 个互不相同的整数，表示这 m 个损坏的座位编号。

Output

对于每组数据，输出一行一个长度为 n 的字符串，表示公交车最终的座位状态：

- 字符 `'a'` 表示该座位由 Alice 安排的游客占据。
- 字符 `'b'` 表示该座位由 Bob 安排的游客占据。
- 字符 `'x'` 表示该座位是损坏的（无法入座）。
- 字符 `'.'` 表示该座位完好但最终为空。

如果有多种可能的结果，输出任意一种即可。

Examples

standard input	standard output
1 1 0	a
5 2 1 3	a.xb.

Note

在第一组样例中，Alice 安排游客入座座位 1。随后没有合法的空座位，Bob 无法操作，游戏直接结束。最终状态为 `a`。

在第二组样例中，座位 3 损坏，初始状态为 `..x..`。

- Alice 可以选择座位 1，状态变为 `a.x..`。由于距离限制 $k=2$，此时座位 2 变得不可用。
- Bob 可以选择座位 4，状态变为 `a.xb.`。同理，座位 5 变得不可用。
- 双方均无合法座位可供选择，游戏结束。

在此对局中双方均完美发挥。注意，输出 `a.x.b` 也是一种合法且正确的最终状态。

Problem C. 穿越鳌太线

Input file: standard input
Output file: standard output

鳌太线是一条连接秦岭鳌山与太白山的高海拔徒步路线，以其变幻莫测的气候、陡峭的地形和稀薄的氧气而闻名，被称为"行走在中华龙脊"上的艰险之路。在这条路线上，每一处地形都对应着不同的探险难度和资源补给情况。

钒钒作为一名经验丰富的探险者，需要规划一条从起点到终点的安全路径。每到达一个地点，钒钒会消耗一定体力并消耗/补充部分物资。我们用一个 $n \times m$ 的山区地图来模拟这条路线，每个地点都有一个数值标记。

你需要从左上角 $(1,1)$ 出发，到达右下角 (n,m)。你可以向上下左右四个方向移动，且可以重复经过同一个地点。

设探险者路径上的地点值依次为 $a_1, a_2, \ldots, a_k$（其中 a_1 是起点 $(1,1)$ 的值，a_k 是终点 (n,m) 的值），定义：

消耗的体力为 $OR = a_1 \mid a_2 \mid \ldots \mid a_k$（按位或运算）

剩下的物资为 $AND = a_1 \& a_2 \& \ldots \& a_k$（按位与运算）

要求在保证消耗的体力最小的情况下，最大化剩下的物资。请输出满足条件的最小体力消耗和最大物资剩余。

Input

第一行包含一个整数 T（$1 \le T \le 100$），表示测试用例的数量。

对于每个测试用例：

第一行包含两个整数 n 和 m（$1 \le n, m \le 10^5$ 并且 $n \times m \le 2 \cdot 10^5$），表示矩阵的行数和列数。

接下来 n 行，每行包含 m 个非负整数 $a_{ij} < 2^{30}$，表示矩阵中的元素。

保证对于一个测试点的所有数据， $n \cdot m$ 的和 不超过 $2 \cdot 10^5$

Output

对于每个测试用例，输出一行，包含两个整数，分别表示满足条件的最小体力消耗和最大物资剩余。

Examples

standard input	standard output
1 2 2 11 2 3 7	15 3
1 2 1 14 100	110 4

Note

对于样例一，

(1,1) -> (1,2) -> (2,2) 的路线 消耗的体力为 15 ，剩余的物资为 2 ；

(1,1) -> (2,1) -> (2,2) 的路线 消耗的体力为 15 ，剩余的物资为 3 ；

除此并无消耗体力更少的方案，所以选择剩余物资更多的路线，故答案为 15 3。

Problem D. 塔菲的LCM

Input file: standard input
Output file: standard output

塔菲给你了一个整数 c。

你需要帮塔菲找到两个正整数 a 和 b，满足 $a+b=c$，并且使它们的最小公倍数 $\mathrm{lcm}(a,b)$ 最大。

两个正整数 a 和 b 的最小公倍数是指同时能被 a 和 b 整除的最小正整数。

Input

输入包含多组数据。

第一行包含一个整数 t $(1 \le t \le 10^4)$

每个测试用例只有一行，包含一个整数 c $(2 \le c \le 10^9)$。

Output

对于每个测试用例，输出一行一个整数 —— $\mathrm{lcm}(a,b)$ 的最大可能值。

Example

standard input	standard output
4 4 6 9 114514	3 5 20 3278364045

Problem E. 简单数学

Input file: standard input
Output file: standard output

众所周知，琪露诺最喜欢的数字就是 9 啦！

今天，琪露诺拿到了一个正整数 x。她突发奇想：能不能找到另一个正整数 k，使得两者的乘积 $x \cdot k$ 变成一个每一位上全都是 9 的数字（比如 9、99、999999 等）呢?

Input

输入包含多组数据。

首先输入一行一个整数 t $(1 \le t \le 10^4)$，表示数据的组数。

对于每组数据，第一行包含一个正整数x $(1 \le x \le 10^{12})$ 。

保证对于一个测试点的所有数据 x 的和不超过 10^{12}。

Output

对于每组测试数据:

如果存在满足条件的整数 k，请输出一行 YES。

如果不存在满足条件的整数 k，请输出一行 NO。

你可以以任意大小写形式输出答案。例如，字符串 yEs、yes、Yes 均会被视为肯定答案。

Example

standard input	standard output
4 3 10 7 12	YES NO YES NO

Note

- **在第一组测试数据中**，$x = 3$。当选择 $k = 3$ 时，两者的乘积为 $3 \times 3 = 9$，符合条件，因此答案为 YES。
- **在第二组测试数据中**，$x = 10$。可以证明，不存在任何正整数 k 使得 $10 \cdot k$ 的结果完全由数字 9 组成，因此答案为 NO。
- **在第三组测试数据中**，$x = 7$。当选择 $k = 142857$ 时，两者的乘积为 $7 \times 142857 = 999999$，符合条件，因此答案为 YES。
- **在第四组测试数据中**，$x = 12$。可以证明，不存在任何正整数 k 满足乘积每一位全为 9 的条件，因此答案为 NO。

Problem F. 钒钒的括号序列

Input file: standard input
Output file: standard output

钒钒有一个合法的括号序列 s，我们对其中的所有括号对进行编号，编号范围为 1 到 n。编号的顺序是按照左括号出现的顺序进行的，即第一个出现的左括号与其对应的右括号组成编号为 1 的括号对，第二个出现的左括号与其对应的右括号组成编号为 2 的括号对，以此类推。

钒钒根据这个合法括号序列构造了一棵二叉树，构造规则如下：

对于编号为 p 的括号对（对应的括号字符位置为 i 和 j，其中 s_i 是左括号，s_j 是右括号），我们考察 s_{i+1} 和 s_{j+1}：

如果 s_{i+1} 是左括号，那么将其对应的括号对编号 l 作为 p 的左孩子；

如果 s_{j+1} 是左括号，那么将其对应的括号对编号 r 作为 p 的右孩子；

如果考察的括号字符为该括号序列的末尾即 s_{2n} 就忽略此考察。

现在，钒钒丢失了原括号序列 s，但保留了构造出的二叉树结构。请你根据给定的二叉树，构造出原来的合法括号序列 s。

Input

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

对于每组测试用例，第一行包含一个整数 n $(1 \le n \le 10^5)$，表示括号对的数量。

接下来 n 行，每行包含两个整数 l 和 r，分别表示编号为 1 到 n 的每个括号对的左孩子和右孩子。如果某个括号对没有左孩子或右孩子，则对应的 l 或 r 为 0 。

保证输出的二叉树的根节点的编号为 1。

保证对于一个测试点的所有数据 n 的和不超过 $2 \cdot 10^5$

Output

对于每组测试数据，输出一行，包含一个由 (和) 组成的字符串，表示构造出的合法括号序列 s。

Example

standard input	standard output
1 3 2 3 0 0 0 0	(())()

Note

样例输入对应的二叉树结构如下：

编号为 1 的括号对的左孩子是 2，右孩子是 3；

编号为 2 和 3 的括号对没有孩子。

对应的合法括号序列为 (())()，其中：

编号 1 的括号对是第 1 个字符（左括号）和第 4 个字符（右括号）；

编号 2 的括号对是第 2 个字符（左括号）和第 3 个字符（右括号）；

编号 3 的括号对是第 5 个字符（左括号）和第 6 个字符（右括号）。

Problem G. 塔菲大战哥布林

Input file: standard input
Output file: standard output

法师塔菲正在与 N 个哥布林进行战斗。她将释放 N 枚魔法飞弹，第 i 枚飞弹固定攻击第 i 个哥布林。

已知第 i 枚飞弹的穿透力为 A_i，潜在伤害为 C_i；第 i 个哥布林的防御力为 B_i。当且仅当飞弹的穿透力大于等于哥布林的防御力（即 $A_i \geq B_i$）时，才能对其造成 C_i 点伤害；否则造成 0 伤害。

塔菲拥有一次使用秘法的机会：在结算伤害前，她可以选择 **至多** 一枚飞弹，将其穿透力强制修改为 X（她也可以选择不修改任何飞弹）。

现在有 Q 次 **独立** 的询问。每次询问会给定一个修改值 X，请你计算：对于当前给定的 X，塔菲在最优的修改策略下，最多能对哥布林造成多少总伤害?

Input

输入包含多组测试数据。

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。

对于每组数据:

第一行包含两个整数 N, Q $(1 \leq N, Q \leq 2 \cdot 10^5)$，分别表示哥布林（飞弹）的数量以及询问的次数。

第二行包含 N 个整数 $A_1, A_2, \ldots, A_N$。

第三行包含 N 个整数 $B_1, B_2, \ldots, B_N$。

第四行包含 N 个整数 $C_1, C_2, \ldots, C_N$。

接下来 Q 行，每行包含一个整数 X，表示一次询问给定的修改值。

数据范围:

- $1 \leq A_i, B_i, C_i, X \leq 10^9$
- 保证对于单个测试点，所有数据中 N 的总和以及 Q 的总和分别不超过 $2 \cdot 10^5$。

Output

对于每组数据，输出共 Q 行，每行包含一个整数，表示对于对应的修正值 X，塔菲能够对哥布林造成的最大总伤害。

Example

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

Note

初始不使用秘法时，仅第 2 枚飞弹命中，基础伤害为 5。

对于第一次查询 $X = 3$：修改第 1 枚飞弹，满足 $3 \geq B_1$，额外造成 1 点伤害。最大总伤害为 $5+1 = 6$。

对于第二次查询 $X = 1$：修改任何未命中飞弹均无法使其满足命中条件，不修改最优。最大总伤害维持基础的 5。

对于第三次查询 $X = 7$：修改第 3 枚飞弹最优，满足 $7 \geq B_3$，额外造成 5 点伤害。最大总伤害为 $5 + 5 = 10$。

对于第四次查询 $X = 8$：依然修改第 3 枚飞弹最优（总伤害 10）；若修改第 4 枚飞弹总伤害仅为 $5 + 2 = 7$。最大总伤害仍为 10。

Problem H. 小 t 的递推

Input file: standard input
Output file: standard output

请注意本题特殊的时间限制和空间限制

小 t 最近对奇妙的数字序列和二进制表示产生了浓厚的兴趣。

他定义了一个带参数 a 的广义递推数列 $f(i)$，形式化地表示为：

$$f(i) = \begin{cases} 0 & \text{if } i = 0, \\ 1 & \text{if } i = 1, \\ a \cdot f(i-1) + f(i-2) & \text{if } i \geq 2. \end{cases}$$

同时，小 t 定义函数 $g(x)$ 为正整数 x 在二进制表示下含有 `1` 的个数（即 popcount）。例如 $g(5) = g(101_{(2)}) = 2$。

现在，小 t 手头有一个极其庞大的正整数 n，由于它太大了，小 t 只能把它写成一个没有前导零的二进制字符串 s。他还给定了递推数列的参数 a。

他非常想知道下面这个求和式的值：

$$\sum_{i=1}^{n} f(g(i)) \pmod{10^9 + 7}$$

由于数字 n 实在是太大了，小 t 的电脑算冒烟了也没能算出结果。你能写一个程序帮助他吗?

Input

第一行包含一个仅由 `0` 和 `1` 组成的字符串 s $(1 \leq |s| \leq 10^7)$，表示数字 n 的二进制形式。保证首字符为 `1`。

第二行包含一个正整数 a $(1 \leq a \leq 10^9)$。

Output

输出一行一个整数，表示小 t 要求的求和式对 $10^9 + 7$ 取模后的结果。

Example

standard input	standard output
101 2	7

Note

在样例中，字符串 `101` 对应的十进制数字为 $n = 5$。参数 $a = 2$。

数列 f 的前几项计算如下：

- $f(0) = 0$
- $f(1) = 1$
- $f(2) = 2 \times 1 + 0 = 2$

我们需要计算 i 从 1 到 5 的 $f(g(i))$ 之和：

- $i = 1$ $(1_{(2)})$: $g(1) = 1 \implies f(1) = 1$

- $i = 2$ ($10_{(2)}$): $g(2) = 1 \implies f(1) = 1$
- $i = 3$ ($11_{(2)}$): $g(3) = 2 \implies f(2) = 2$
- $i = 4$ ($100_{(2)}$): $g(4) = 1 \implies f(1) = 1$
- $i = 5$ ($101_{(2)}$): $g(5) = 2 \implies f(2) = 2$

所以最终答案为 $1 + 1 + 2 + 1 + 2 = 7$。

本题输入输出数据量较大，建议使用较快的输入输出方式。C++ 选手建议使用 `scanf/printf`，或在使用 `cin/cout` 时取消同步流:

```
std::ios::sync_with_stdio(false);
std::cin.tie(nullptr);
```

Problem I. HZAUACM 集训室的奶茶

Input file: standard input
Output file: standard output

周日下午，HZAUACM 集训室里键盘声此起彼伏。为了犒劳正在刻苦刷题的小登们，凡哥决定给大家点"蜜雪冰城"。

现在集训室里一共有 n 名小登，凡哥需要买**恰好** n 杯奶茶 。

打开拼好饭，凡哥发现这家店有两种购买方式：

- 单点：买 1 杯奶茶需要 a 元。
- 双人套餐：买 2 杯奶茶只需 b 元。

现在，请你帮凡哥算算，他最少花多少钱能**恰好**买到 n 杯奶茶?

Input

输入包含多组数据。

首先输入一行一个整数 t $(1 \le t \le 10^4)$，表示数据的组数。

对于每组数据，输入一行，包含三个整数 n, a, b $(1 \le n, a, b \le 10^9)$，分别表示需要的奶茶数量、单杯价格和双人套餐价格。

Output

对于每组数据，输出一个整数，表示买到 n 杯奶茶的最小花费。

Example

standard input	standard output
4 5 2 5 4 3 4 5 3 4 1 10 5	10 8 11 10

Note

在第一组数据中，最优策略是全部单点，买 5 杯的最小花费为 $5 \times 2 = 10$ 元。

在第二组数据中，最优策略是买 2 个双人套餐，最小花费为 $2 \times 4 = 8$ 元。

在第三组数据中，最优策略为买一个双人套餐和单点一杯，最小花费为 $2 \times 4 + 1 \times 3 = 11$ 元。

在第四组数据中，最小花费为 10 元。

Problem J. 树

Input file: `standard input`
Output file: `standard output`

给定一棵以 1 为根、包含 N 个节点的树。每个节点 i 有一个初始权值 M_i。 系统在任意时刻均严格维护以下性质：对于任意非根节点 x，其权值绝不会超过其父节点的权值，即 $M_{\text{parent}(x)} \geq M_x$。

你需要在线处理 Q 次操作，操作分为以下两种：

- `1 u w`：**查询**。从节点 u 开始，沿着树边向根节点遍历，查找并输出第一个满足权值 $\geq w$ 的节点编号（包括 u 本身）。若直到根节点都不满足，输出 `-1`。
- `2 x v`：**修改**。尝试将节点 x 的权值加上 v（v 可以为负数），即令 $M_x \leftarrow M_x + v$。为了不破坏树的性质，修改前必须进行合法性检查：
  - 若 $x \neq 1$ 且 $M_x + v > M_{\text{parent}(x)}$，则修改失败。
  - 若 x 不为叶子节点且 $M_x + v < \max_{y \in \text{children}(x)} M_y$，则修改失败。

  若上述两项检查均通过，则实际执行该修改并输出 `SUCCESS`；否则，拒绝本次修改，树的状态保持不变，并输出 `FAILED`。

Input

输入包含多组数据。

首先输入一行一个整数 t $(1 \leq t \leq 10^4)$，表示数据的组数。

对于每组数据，第一行包含两个整数 N, Q。

第二行包含 N 个整数 $M_1, M_2, \ldots, M_N$，保证初始权值满足上述性质。

接下来 $N-1$ 行，每行两个整数 u, v，表示树上的一条边。

接下来 Q 行，每行三个整数 opt, a, b 表示一次操作：当 $opt = 1$ 时代表 `1 u w` 操作；当 $opt = 2$ 时代表 `2 x v` 操作。

数据范围：

- $1 \leq N, Q \leq 10^5$
- $0 \leq M_i, w \leq 10^9$
- $-10^9 \leq v \leq 10^9$
- 保证任意时刻 $M_i \geq 0$

保证对于单个测试点，所有数据中 N 的总和以及 Q 的总和分别不超过 $2 \cdot 10^5$。

Output

对于每组测试数据，输出 Q 行，每行对应一次操作的处理结果：

- 对于 `1 u w` 操作，输出符合条件的节点编号，若不存在则输出 `-1`。
- 对于 `2 x v` 操作，若修改成功输出 `SUCCESS`，若修改失败输出 `FAILED`。

Example

standard input	standard output
1 5 6 100 80 50 30 20 1 2 1 3 2 4 2 5 1 4 40 1 5 150 2 2 30 2 2 -60 2 4 10 1 4 40	2 -1 FAILED FAILED SUCCESS 4

Note

在样例中，树的结构为：节点 1 的子节点是 2 和 3；节点 2 的子节点是 4 和 5。

初始权值为 $M = [100, 80, 50, 30, 20]$。

- **第 1 次操作** `1 4 40`：从节点 4 开始向上查找权值 ≥ 40 的节点。节点 4 当前权值为 $30 < 40$，其父节点 2 权值为 $80 \geq 40$ 满足条件，输出 `2`。
- **第 2 次操作** `1 5 150`：从节点 5 开始向上查找权值 ≥ 150 的节点。沿途直到根节点 1（权值为 100）都不满足条件，输出 `-1`。
- **第 3 次操作** `2 2 30`：尝试将节点 2 的权值增加 30，修改后为 110。由于 $110 > M_1$ (100)，违反了父节点权值 $\geq$ 子节点权值的性质，修改失败，输出 `FAILED`。
- **第 4 次操作** `2 2 -60`：尝试将节点 2 的权值减少 60，修改后为 20。由于 $20 < \max(M_4, M_5)$ (30)，同样违反了树的权值单调性质，修改失败，输出 `FAILED`。
- **第 5 次操作** `2 4 10`：尝试将节点 4 的权值增加 10，修改后为 40。满足 $40 \leq M_2$ (80)，且节点 4 为叶子节点，修改成功，更新 $M_4 = 40$，输出 `SUCCESS`。
- **第 6 次操作** `1 4 40`：再次从节点 4 向上查找权值 ≥ 40 的节点。由于上一步修改成功，此时 $M_4 = 40 \geq 40$ 已满足条件，直接输出 `4`。

Problem K. MEX

Input file: standard input
Output file: standard output

给定一棵包含 n 个节点的无根树，节点编号为 1 到 n。树上的每个节点 i 都有一个权值 p_i。保证所有节点的权值 $p_1, p_2, \ldots, p_n$ 恰好构成一个排列。

对于树上的一个连通块（即由树上若干个节点及它们之间的边构成的连通子图），我们定义它的 MEX 值为：该连通块内所有节点的权值集合中，未出现过的最小非负整数。

现在有 q 次查询。每次查询给定两个整数 A 和 B，请你计算出在这棵树的所有连通块中，有多少个连通块同时满足以下两个条件：

1. 该连通块包含节点 A。
2. 该连通块的 MEX 值恰好等于 B。

由于符合条件的连通块数量可能非常大，请将每次查询的答案对 $10^9 + 7$ 取模后输出。

- **排列**：如果一个包含 n 个元素的序列，恰好包含了从 0 到 $n-1$ 的每一个整数各一次，那么称该序列为一个 0 到 $n-1$ 的排列。例如，$[2, 0, 3, 1]$ 是一个排列；而 $[0, 1, 1, 3]$ 不是一个排列。

Input

输入包含多组数据。

首先输入一行一个整数 t $(1 \le t \le 10^4)$，表示数据的组数。

对于每组数据，第一行包含两个整数 n, q $(1 \le n, q \le 2 \cdot 10^5)$。

第二行包含 n 个整数 $p_1, p_2, \ldots, p_n$ $(0 \le p_i < n)$。保证这些数构成了一个 0 到 $n-1$ 的排列。

接下来 $n-1$ 行，每行两个整数 u, v $(1 \le u, v \le n)$，表示树上的一条边。

接下来 q 行，每行两个整数 A, B $(1 \le A \le n, 0 \le B \le n)$，表示一次查询。

保证对于一个测试点的所有数据，n 以及 q 的和分别不超过 $2 \cdot 10^5$。

Output

对于每组数据，输出共 q 行，每行一个整数，表示该次查询的答案对 $10^9 + 7$ 取模后的结果。

Example

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

Note

在第一组样例中：

- 对于第一次查询 $(A = 2, B = 0)$：节点 2 的权值为 0。任何包含节点 2 的连通块的 MEX 值都必然大于 0，因此不存在满足条件的连通块。
- 对于第二次查询 $(A = 4, B = 1)$：满足条件的连通块必须包含节点 4 以及权值为 0 的节点 2，且不能包含权值为 1 的节点 3。由于在树上连接节点 4 和节点 2 的唯一路径必须经过节点 3，因此无法构造出合法的连通块。
- 对于第三次查询 $(A = 4, B = 2)$：满足条件的连通块有 $\{2, 3, 4\}$ 与 $\{2, 3, 4, 5\}$，它们的 MEX 值均为 2 且都包含节点 4，共 2 种方案。
- 对于第四次查询 $(A = 1, B = 3)$：满足条件的连通块有 $\{1, 2, 3\}$ 与 $\{1, 2, 3, 5\}$，共 2 种方案。

Problem L. 龙卷风摧毁停车场

Input file: standard input
Output file: standard output

你被指派去疏导一个严重拥堵的网格停车场。该停车场可以看作一个 $N \times M$ 的矩阵。矩阵中的每个车位要么是空地（用 . 表示），要么停着一辆带有固定行驶方向的汽车，方向分为四种：U（上）、D（下）、L（左）、R（右）。

你可以按照任意顺序向这些汽车发送"驶离"信号。一辆汽车收到信号后，如果从它当前位置出发，沿着其车头朝向一直到停车场边界的直线路径上没有任何其他汽车阻挡，它就会成功驶离停车场，并将当前车位变成空地（.）。

请问是否存在一种发送信号的顺序，能够将停车场内的所有汽车全部疏散？如果可以，请输出依次发送信号的汽车坐标序列；如果因为车辆互相卡死（死锁）而无法完全疏散，请输出 -1。

Input

第一行包含两个整数 N 和 M（$1 \le N, M \le 2000$），分别表示停车场的行数和列数。

接下来的 N 行，每行包含一个长度为 M 的字符串，仅由字符 U, D, L, R, . 组成。

保证停车场中至少有一辆车。

Output

如果可以疏散所有汽车，输出 K 行（K 为初始时停车场内汽车的总数）。每行包含两个整数 r 和 c（$1 \le r \le N$，$1 \le c \le M$），表示依次发送信号的汽车所在的行号和列号（坐标从 1 开始编号）。如果有多种合法的疏散顺序，输出任意一种即可。

如果无论如何都无法疏散所有汽车，仅输出一行 -1。

Examples

standard input	standard output
3 3 D.. R.. .UL	2 1 3 2 1 1 3 3
2 2 RD UL	-1

Note

在第一个样例中，我们可以按照以下顺序疏散车辆：

1. 向位于 (2, 1) 且朝向右侧 (R) 的汽车发送信号。由于其右侧直线路径上没有其他汽车阻挡，它成功驶离。

2. 向位于 (3, 2) 且朝向上方 (U) 的汽车发送信号。其上方路线畅通，成功驶离。

3. 向位于 (1, 1) 且朝向下方 (D) 的汽车发送信号。原本阻挡它的 (2, 1) 处的汽车已经驶离，所以它现在能成功驶离。

4. 向位于 (3, 3) 且朝向左侧 (L) 的汽车发送信号。原本阻挡它的 (3, 2) 处的汽车已经驶离，它也成功驶离。

至此，所有汽车均被成功疏散。需要注意的是，合法的输出顺序可能不止一种。

在第二个样例中，位于 (1, 1) 的汽车被 (1, 2) 处的汽车阻挡；(1, 2) 处的汽车被 (2, 2) 处的汽车阻挡；(2, 2) 处的汽车被 (2, 1) 处的汽车阻挡；而 (2, 1) 处的汽车又恰好被 (1, 1) 处的汽车阻挡。初始状态下没有任何一辆汽车可以移动，因此发生死锁，无法完全疏散，答案输出 -1。

Problem M. 小 t 的GCD

Input file: standard input
Output file: standard output

小 t 给你一个包含 n 个正整数的序列 $a_1, a_2, \ldots, a_n$，以及一个常数 P。你需要将这个序列划分成 m 个（$1 \le m \le n$）连续且非空的子区间。每一个元素恰好属于一个子区间。假设你划分出的某一个子区间为 $[L, R]$（$1 \le L \le R \le n$）。对于任意一个给定的子区间 $[L, R]$，我们定义其 $W(L, R)$ 为该区间所有下标对应的正奇数之和。形式化而言：

$$W(L,R) = \sum_{i=L}^{R} (2i-1)$$

同时，定义该区间的 $G(L, R)$ 为该区间内所有元素的最大公约数。形式化而言：

$$G(L,R) = \gcd(a_L, a_{L+1}, \ldots, a_R)$$

每一次划分出一个区间，都需要消耗 P 的代价。因此，对于区间 $[L, R]$，其权值计算方式如下：

$$\mathcal{E}(L,R) = W(L,R) \times G(L,R) - P$$

小t希望你找到一种划分方案，使得这 m 个子区间的最终权值之和越大越好。

Input

输入包含多组数据。

首先输入一行一个整数 t $(1 \le t \le 10^4)$，表示数据的组数。

对于每组数据，第一行包含两个整数 n, P $(1 \le n \le 10^5,\ 0 \le P \le 10^{14})$。

第二行包含 n 个整数 $a_1, a_2, \ldots, a_n$ $(1 \le a_i \le 10^5)$。

保证对于一个测试点的所有数据，n 的和不超过 10^5。

Output

对于每组数据，输出一行一个数字表示结果。

Example

standard input	standard output
1 3 25 8 12 6	6