Problem A. 不要玩弄字符串

小 X 和小 Y 都是字符串大师，同样作为字符串大师，他们决定玩一个游戏来分出高下。这一天，他们挑出了 k 个 01 串并对它们的优美程度进行衡量，其中第 i 个 01 串 t_i 的优美度为 v_i。（注意，v_i 可能为负数）

他们初始选定了一个 01 串 S，并轮流往 S 末尾添加一个 0 或 1，小 X 先手。当一名玩家添加完字符后，若某些 t_i **第一次**成为 S 的子串，那么该玩家获得这些 01 串优美度之和的得分。

当所有 t_i 都成为 S 的子串时，游戏结束。小 X 和小 Y 都希望最大化游戏结束时自己的得分减去对方的得分。

或许你注意到了，在有些情况下游戏永远不会结束。于是小 X 和小 Y 新增了求和机制，若一名玩家认为继续游戏不会增大自己的收益（即自己得分减去对面得分），那么他将发起求和。若两名玩家都发起求和，则游戏立刻结束。

小 X 和小 Y 玩弄字符串的水平不分上下，初始串 S 的选取很可能影响游戏的公平性。于是小 X 和小 Y 准备了 q 个 01 串 $S_1, S_2, \ldots, S_q$ 找到了你，想知道当初始串 $S = S_i$ 时，若小 X 和小 Y 采取最优策略，那么最终小 X 的得分减去小 Y 的得分为多少。

Input

第一行一个正整数 k。

接下来 k 行，每行一个 01 串以及一个整数。第 i 行的 01 串表示 t_i，整数表示 v_i。

接下来一行一个正整数 q。

接下来 q 行，每行一个 01 串。第 i 行的 01 串表示 S_i。

保证 $1 \le k \le 18, 1 \le q \le 524286, 1 \le |t_i| \le 8, 1 \le |S_i| \le 18, |v_i| \le 10^5$。

Output

输出 q 行，第 i 行输出当 $S = S_i$ 时，最终小 X 的得分减去小 Y 的得分的值。

Example

standard input	standard output
3 11 1 0 2 000 3 2 0 1	-1 3

Note

对于样例 1：

- 当 S 初始为 0。若小 X 第一步添加 0，则小 Y 添加 0，此时 S 为 000，小 X 得分减去小 Y 得分为 $0 - v_3 = -3$，易知接下来 v_1 不可能被任何人获得。若小 X 第一步添加 1，则小 Y 添加 1，此时 S 为 011，小 X 得分减去小 Y 得分为 $0 - v_1 = -1$，易知接下来 v_3 不可能被任何人获得。故最优策略下答案为 -1。
- 当 S 初始为 1。一种最优策略为小 X 第一步添加 0，此时等价于小 Y 先手，初始串为 0 的情况。由上讨论，最优策略下答案为 $v_1 + v_2 = 3$。

Problem B. 组合数

有 t 次询问。每次询问给出四个整数 l, r, n, m，问有多少个整数 $i \in [l, r]$ 满足存在两个整数 $0 \le u \le n, 0 \le v \le m$ 满足 $i = \binom{u}{v}$。这里，$\binom{u}{v}$ 的定义为：若 $u \ge v$，则 $\binom{u}{v} = \frac{u!}{v!(u-v)!}$。否则，$\binom{u}{v} = 0$。

Input

第一行一个整数 t $(1 \le t \le 10^5)$。

接下来 t 行，每行四个整数 l, r, n, m，描述一次询问 $(1 \le l, r, n, m \le 10^9, r \ge l)$。

Output

对于每组询问输出一行一个整数表示答案。

Example

standard input	standard output
2 1 6 4 2 30 60 9 3	5 3

Problem C. 放苹果

有 m 个盘子和 n 个苹果，现在要把每个苹果都放到 m 个盘子里面的一个，总共 m^n 情况。定义一次操作为将一个苹果移动到某个相邻的盘子里。对于一种放苹果的情况，代价为最少操作的次数，使得通过这么多次操作能将所有苹果放到同一个盘子里。求所有 m^n 种情况的代价的和，对 998244353 取模。

Input

输入一行两个整数 $n, m (1 \le n \le 2 \times 10^5, 1 \le m \le 10^9)$。

Output

输出一行一个整数，表示所有情况代价之和对 998244353 取模的结果。

Examples

standard input	standard output
2 3	8
3 3	36

Problem D. 树

五一假期到了，Alice 和 Bob 都到了城市 P 旅游，城市 P 中有 n 个景点，而这 n 个景点之间的道路构成了一棵树。

Alice 的旅行规划中，有 c 个目标景点，一开始，Alice 会直接出现在景点 a_1，然后顺着最短路，沿着城市 P 中的道路到达 a_2（她同样会路过最短路上的其它景点），然后再从 a_2 前往 a_3，……，最后 Alice 抵达 a_c 过后，她会直接离开城市 P。

Bob 同样也来到了城市 P，在整个五一期间，他和 Alice 在 m 个景点相遇过，其中第 i 次是在景点 b_i 相遇。

Bob 想要知道，Alice 在五一期间最少定下了多少个目标景点，也即可能的 c 的最小值。Bob 的记忆已经比较模糊了，所以他会修改多次数组 b，你需要给出每一次修改后的最小的可能的 c。

Input

第一行两个正整数 n, m, q，表示树的节点数，序列 b 的长度，以及修改个数。

下面 $n-1$ 行，每行两个正整数 u_i, v_i，依次描述了树的每一条边。

下面一行 m 个正整数表示数组 b。

下面 q 行，每行两个正整数 p_i, w_i，表示将 b_{p_i} 修改为 w_i。

$1 \le n, m, q \le 2 \times 10^5, 3 \le n$

$1 \le u_i, v_i, w_i \le n, 1 \le p_i \le m$。

对于任意时刻，$\forall 1 \le i < m : b_i \ne b_{i+1}$。

Output

一共 q 行，每行一个正整数表示答案。

Example

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

Note

对于第一组样例第一次修改后的 b，一组可能的最短的 a 为 $[3, 5, 4, 3]$。此时 Alice 经过城市的顺序为：$[3, 2, 1, 5, 1, 4, 1, 2, 3]$

Problem E. 骰子

有 n 枚 $m+1$ 面的骰子，$m+1$ 个面上分别写着 $0 \sim m$ 这 $m+1$ 个数字，丢出第 i 枚骰子写着数字 j 的面朝上的概率永远是 $p_{i,j}$。

你打算投 q 次骰子，第 i 次丢出编号在 $[l_i, r_i]$ 内的骰子，把每个骰子向上的面写的数字加起来，设总和是 sum，如果 $sum \le m$，你会得到 b_{sum} 的开心度，否则你将什么都得不到，请问每一轮你得到的期望开心度是多少？答案对 10^9+7 取模。

Input

第一行三个正整数 $n, m, q (1 \le n \le 1500, 1 \le m \le 200, 1 \le q \le 6 \times 10^5)$。

第二行 $m+1$ 个非负整数，分别表示 $b_0, \ldots, b_m (0 \le b_i < 10^9+7)$。

接下来 n 行，每行 $m+1$ 个非负整数 $p'_{i,0}, \ldots, p'_{i,m} (p'_{i,j} \ge 0, \sum_{j=0}^{m} p'_{i,j} = 10^9+8)$，表示 $p_{i,j} = \frac{p'_{i,j}}{10^9+8}$。

接下来 q 行，每行两个正整数 $l_i, r_i (1 \le l_i \le r_i \le n)$。

Output

q 行，每行一个非负整数表示答案对 10^9+7 取模的结果。

Examples

standard input	standard output
3 3 3 4 3 2 1 0 1 0 1000000007 0 500000004 0 500000004 0 0 500000004 500000004 1 1 1 2 1 3	3 1 0
3 3 6 4 3 2 1 1000000007 0 1 0 1000000007 1 0 0 1000000007 0 1 0 1 1 1 2 1 3 2 2 2 3 3 3	2 1 0 3 1 2

Problem F. 保区间最小值一次回归问题

给定一个正整数序列 $[a_1, \ldots, a_n]$，再给定 m 个三元组 (l_j, r_j, v_j) $(1 \le j \le m)$。

你需要求出一个正整数序列 $[b_1, \ldots, b_n]$ 满足 $\forall 1 \le j \le m, \min_{l_j \le k \le r_j} b_k = v_j$。在此基础上，最小化 $\sum_{1 \le i \le n} |a_i - b_i|$ 的值。你只需要输出 $\sum_{1 \le i \le n} |a_i - b_i|$ 之最小值就可以了。

若不存在满足所有三元组的 b，输出 -1。

Input

本题多测，第一行一个正整数 t $(1 \le t \le 1000)$ 表示数据组数。接下来输入 t 组数据。

对于每组数据：第一行两个整数 n, m $(1 \le n \le 5 \times 10^5, 1 \le m \le 5 \times 10^5)$。

接下来一行 n 个正整数 $a_1, \ldots, a_n$ $(1 \le a_i \le 10^9)$。

接下来 m 行，每行三个正整数 l_j, r_j, v_j $(1 \le l_j \le r_j \le n, 1 \le v_j \le 10^9)$。

保证所有数据中 n 的和不超过 10^6，m 的和不超过 10^6。

Output

对于每组数据输出一行一个整数表示答案。

Example

standard input	standard output
1 3 2 2023 40 41 1 1 2022 2 3 39	2

Note

样例的一组最优解是 $b = [2022, 39, 41]$。

Problem G. 后继

这是一道交互题。

给定一个长度为 n 的序列 a，**保证序列 a 中元素互不相同**，通过序列 a 可以构造一个长度为 n 的序列 b，满足 $b_i = a_i \operatorname{xor} x$，$x$ 是一个满足 $0 \le x < 2^{30}$ 的非负整数。你知道序列 a 但不知道 x 和序列 b，你希望通过若干次询问确定 b 所有元素组成的全序关系。

每次询问，你需要给出一个下标 $1 \le p \le n$，交互库会回答序列 b 所有 $b_i > b_p$ 的 b_i 中，值最小的 b_i 的 **下标**。形式化地，记 $res = \min\limits_{1 \le i \le n, b_i > b_p} b_i$，交互库会回答唯一的下标 i 满足 $b_i = res$。如果不存在 $b_i > b_p$，交互库会回答 -1。

为了减小输出量，最终你只需要输出最小的非负整数 x_0 使得由 x_0 构造的长度为 n 的序列 c 满足 $c_i = a_i \operatorname{xor} x_0$ 和 b 的全序关系完全相同即可。换句话说，c 应当满足 $\forall 1 \le i < j \le n, (b_i - b_j)(c_i - c_j) > 0$。

每轮交互中，你可以进行至多 30 次询问。每个测试点中你需要进行 m 轮交互，不同轮交互中的 n, a 是相同的，但 x 和序列 b 可能不同。

Interaction Protocol

首先读入两个正整数 $n, m (1 \le n \le 4 \times 10^5, 1 \le m \le 3 \times 10^3)$，表示序列长度和交互轮数。

接下来再读入 n 个非负整数，表示序列 $a (0 \le a_i < 2^{30})$。

接下来进行 m 轮交互，每轮的格式分别为：

— 首先可以进行不超过 30 次询问，每轮询问的格式为 `? p`，交互库会回答一个整数 q。如果 $q = -2$ 代表超出了询问次数上界或进行了非法操作，此时需要立即退出并且评测结果会返回 Wrong answer。如果 $q = -1$ 或 $1 \le q \le n$ 那么交互库正常回答了你的询问。

— 当你确定了 x_0 后，你可以以 `!` x_0 的格式提交结果，无论该结果是否正确本轮交互都会结束，进入下一轮交互（如果已经是最后一轮交互则结束）。**提交答案不算在询问次数中**。

注意在 m 轮交互的过程中和输出答案后，每次进行输出后都要清空缓存区。这在各个语言中可通过如下方式实现：

— 对 C/C++，使用 `fflush(stdout)` 或 `cout.flush()`.

— 对 Java, 使用 `System.out.flush()`.

— 对 Python, 使用 `stdout.flush()`.

在交互过程中，你需要保证询问时给出的 p 满足 $1 \le p \le n$，提交答案时的 x_0 满足 $0 \le x_0 < 2^{30}$，不然可能会出现无法预知的错误。

Example

standard input	standard output
5 1 1 2 3 4 5 5 1 2 -1 4	? 1 ? 2 ? 3 ? 4 ? 5 ! 3

Note

显然这样就确定了 $b_4 > b_5 > b_1 > b_2 > b_3$，可以发现 $x_0 = 3$ 是最小的满足这一全序关系的 x_0，所以最后应回答 3。

事实上的 x 可能并不是 3，比如可以发现 $x = 11$ 生成的序列 b 也满足这一全序关系，但这并不重要，因为不要求还原 x。

Problem H. BFS 序 0

给定一棵 n 个点的树，根为 1。

进行 q 次询问，每次询问给出一个序列，问是否可能是树的一个 BFS 序的子序列。

这里 BFS 序的定义是：维护一个 queue，初始放入根。每次 pop 一个元素 u，按一定的顺序 push 进 u 的所有儿子。

不同询问对应的 BFS 序可以是不同的，即询问独立。

Input

第一行，一个个正整数 n。

第二行，$n-1$ 个正整数，表示 2 到 n 的父亲，保证一个点的父亲编号小于自己。

下面一行，一个正整数 q 表示询问个数。

接下去 q 行，每行首先一个正整数 m，表示询问序列的长度。接下去 m 个正整数，表示询问序列。

注意，询问序列的点可能有重复。

$1 \le n \le 3 \times 10^5, 1 \le q, \sum m \le 5 \times 10^5$。

Output

一共 q 行，每行一个 Yes 或者 No，表示你的答案。

大小写均可。

Example

standard input	standard output
6 1 1 3 2 4 10 4 3 6 2 5 1 4 3 2 4 5 5 2 5 4 6 3 3 1 4 2 3 5 6 3 5 4 5 2 6 1 4 4 3 2 5 4 4 6 2 3 3 3 2 6	No Yes Yes No No No No No No Yes

Problem I. 括号序列

对于一个长为 $2n$ 的合法括号串 S，我们要对它进行 n 次操作。每次操作可以是：

1. 从 S 中删去一个连续子串 ()。位置不同视为不同的操作。例如，$()() \to ()$。
2. 从 S 中删去一个连续子串)(。位置不同视为不同的操作。但是，这里删去的)(必须一开始就在 S 中相邻！

例如，这一串操作是合法的 $()()() \to ()() \to ()$；这一串操作不合法 $()()() \to ()() \to ()$。

显然，n 次操作后 S 会被删空。设 n 次操作删空 S 的方案数为 $f(S)$。

对所有合法括号串 S，求 $\sum f(S) \bmod 998244353$。

Input

输入仅包含一个正整数 n $(1 \le n \le 250000)$。

Output

输出一个整数，表示答案。

Example

standard input	standard output
3	28

Note

- ((())) 有 1 种删空方法。
- (()()) 有 3 种删空方法。
- (())() 有 5 种删空方法。
- ()(()) 有 5 种删空方法。
- ()()() 有 14 种删空方法。

总共 28 种。

Problem J. 卡牌游戏

Alice 和 Bob 在玩卡牌游戏。

每张卡牌上标有 1 至 N 的数字，Alice 拥有的标有数字 i 的卡牌数量为 A_i，Bob 拥有的标有数字 i 的卡牌数量为 B_i。

游戏分为若干轮，每轮游戏由前一轮的胜者开始（如果是第一轮，则 Alice 开始）轮流进行以下决策:

- 选择出一张和本轮前一张卡牌标有相同数字的卡牌，如果本轮还没有出过牌，则可以任意选择。
- 或者选择放弃，此轮结束，由对手获得这一轮的胜利。注意，如果该轮还没有出过牌，你不可以选择放弃。

在任意时刻，如果有一名玩家打出了所有卡牌，则这名玩家获得整局游戏的胜利，游戏结束。

现在有 T 组互不相关的询问，假设 Alice 和 Bob 都知道对方拥有的卡牌，且都采取最优策略，请回答谁会获胜。

Input

第一行一个整数 T（$1 \le T \le 10^6$），表示输入中的询问组数。

每组询问按照以下格式输入:

第一行一个整数 N（$1 \le N \le 10^6$）。

第二行 N 个整数 $A_1, A_2, \ldots, A_N$（$0 \le A_i \le 10^9$，$\sum A_i > 0$）。

第三行 N 个整数 $B_1, B_2, \ldots, B_N$（$0 \le B_i \le 10^9$，$\sum B_i > 0$）。

保证所有询问中 N 的总和不超过 10^6。

Output

每组询问一行，如果 Alice 获胜则输出 `Alice`，如果 Bob 获胜则输出 `Bob`。

Example

standard input	standard output
5 1 100 100 2 1 1 1 1 2 1 1 0 1 3 1 1 4 5 1 4 10 116 104 101 114 101 32 97 114 101 32 102 105 118 101 32 99 97 115 101 115	Alice Bob Alice Alice Alice

Note

第一组询问中，一种可能的游戏过程是：Alice 和 Bob 轮流打出标有数字 1 的卡牌，Alice 在 Bob 还剩一张牌时打出了最后一张牌，获得游戏胜利。

第二组询问中，一种可能的游戏过程是：Alice 打出标有数字 1 的牌，Bob 打出标有数字 1 的牌，Alice 放弃，Bob 获得第一轮的胜利；Bob 打出最后一张标有数字 2 的牌，获得游戏胜利。

Problem K. 乘二

你有 n 个数 $a_1, a_2, \cdots, a_n$，现在你要操作 k 次，每次操作为选择一个数 a_i，将 a_i 乘 2。现在需要求出 k 次操作之后，所有数的和最小是多少。

由于答案可能很大，你只需要输出其对 10^9+7 取模的结果即可。

Input

第一行输入两个数 $n, k (1 \le n \le 2 \times 10^5, 0 \le k \le 10^9)$。

第二行输入 n 个整数，第 i 个整数表示 $a_i (1 \le a_i \le 10^9)$。

Output

输出一行一个整数，表示 k 次操作后所有数之和最小值对 10^9+7 取模的结果。注意是最小值对 10^9+7 取模，而不是对 10^9+7 取模后的最小值。

Example

standard input	standard output
3 3 7 2 1	15