

Problem A. 摇啊摇

Input file: standard input
Output file: standard output
Time limit: 2 seconds
Memory limit: 1024 megabytes

一天，小 Z 在路上遇到了一棵参天大树，挡住了他的去路。没办法，他只能想办法把树全部移走，才能继续前进。小 Z 力大无穷，每当他抓住树的节点，一直摇个不停，这个点就会从树上脱落，变成一棵小树，同时小树的每个叶子都飘落到地上消失了。

现在小 Z 想知道，他需要摇多少次，才能让所有的树的所有节点都飘落到地上，但这太容易了。他发现这棵树并没有很高，所以请你同时告诉小 Z，他有多少种方式，可以用最少的摇树次数让所有的树的所有节点都飘落到地上。

形式化的，有一棵包含 n 个节点的有根树，节点编号为 1 至 n 。树的根为节点 1，且树的高度不超过 100。每次操作时可以选择一个节点 u ，使节点 u 与其父节点断开 (如果有)，成为一颗新树的根节点，然后删除以节点 u 为根的树中的所有叶节点。

现在小 Z 想知道删除所有节点所需的最少操作次数。你还需要回答通过最少次操作删除所有节点的方案数，对 $10^9 + 7$ 取模。

两种操作方案被视为相同的，当且仅当两种方案操作次数相同且每次操作的节点编号均相同。

Input

第一行一个整数 n ($1 \leq n \leq 2 \times 10^5$)，表示树的节点数。

接下来 $n - 1$ 行，每行两个整数 u, v ($1 \leq u, v \leq n$)，表示树上的一条边。保证给出的边构成一棵树。保证树的高度不超过 100。

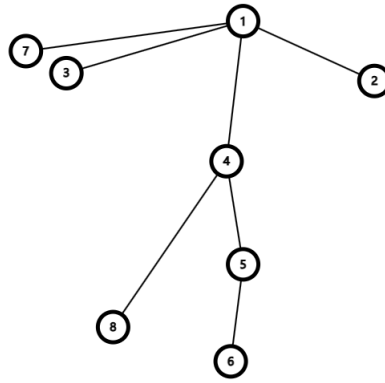
Output

输出一行两个整数，表示最少操作次数和方案数对 $10^9 + 7$ 取模的结果。

Example

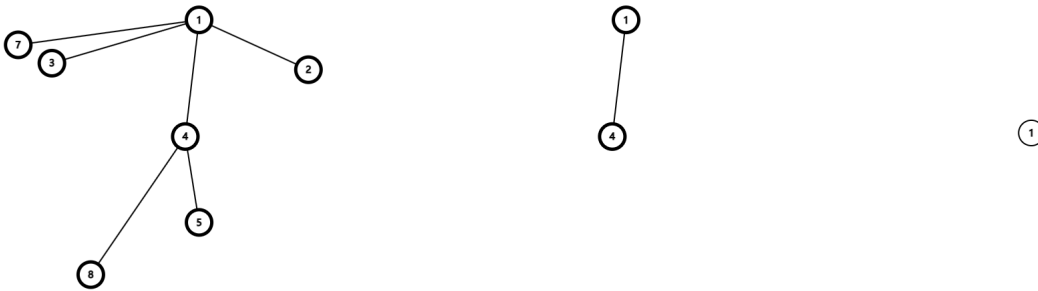
standard input	standard output
8 1 2 1 3 1 4 4 5 5 6 1 7 4 8	4 8

Note



样例中的树

样例中，至少需要 4 次操作才能删除所有节点，且通过 4 次操作删除所有节点的方案一共有 8 种。
这些方案分别为: $[6, 1, 4, 1]$, $[6, 1, 1, 1]$, $[1, 5, 4, 1]$, $[1, 5, 1, 1]$, $[1, 4, 1, 4]$, $[1, 1, 1, 1]$, $[1, 4, 4, 1]$, $[1, 1, 4, 1]$
按顺序删除 $[6, 1, 4, 1]$ 的方案中，所有节点会按如下过程被删除：



选择节点 1 后，所有点都被删除。

Problem B. 无数的我

Input file: standard input
Output file: standard output
Time limit: 1 second
Memory limit: 1024 megabytes

我是小蝶，小玛利亚的一只平平无奇的飞马——呃，在与紫悦她们经历了许许多多的事情后，也许并没有那么平凡。

比如今天，我回到家中，映入我眼帘的是，无数个我。

庆幸的是，我不是萍琪派，她们没有蹦蹦跳跳叽叽喳喳，仅仅是安静的看着我，并给了我一个问题：给定一个长度为 n 的非负整数数组 a_i ，我可以进行如下操作至多 n 次：

- 选择 i, j, x 满足 $1 \leq i, j \leq n, 0 \leq x \leq 10^9$ ，令 $a_i \leftarrow a_i + x, a_j \leftarrow a_j - x$

注意，需要保证每次操作完成后所有的 a_i 仍是非负整数，一个我这么提醒着。

我需要最小化操作后 $a_1 | a_2 | \dots | a_n$ 的值。其中 $|$ 代表按位或符号， $x | y$ 表示两个非负整数 x 和 y 的按位或。

按位或运算是按位比较两个二进制数，在每个位置上，如果相应的位中至少有一个为 1，则结果为 1，否则为 0。例如，二进制数 $10 = 1010_2$ 和 $12 = 1100_2$ 进行按位或运算的结果为 $14 = 1110_2$ ， $3 = 0011_2$ 和 $5 = 0101_2$ 的按位或结果为 $7 = 0111_2$ 。



Input

第一行一个整数 n ($1 \leq n \leq 2 \times 10^5$)，表示数组长度。

第二行 n 个非负整数 a_i ($0 \leq a_i \leq 10^9$)。

Output

一行一个整数表示答案，即最小化的 $a_1 | a_2 | \dots | a_n$ 的值。

Example

standard input	standard output
7 1 9 1 9 8 1 0	5

Note

在样例中，一种可行的最终结果为 $[5, 5, 5, 5, 5, 4, 0]$ ，其具有的最小结果为 5，可以证明，不存在更优的结果。

一种可行的达到此结果方法为 $[1, 9, 1, 9, 8, 1, 0] \rightarrow [5, 5, 1, 9, 8, 1, 0] \rightarrow [5, 5, 5, 5, 8, 1, 0] \rightarrow [5, 5, 5, 5, 5, 4, 0]$ ，共进行了 3 步操作。

Problem C. 书包与最长上升子序列

Input file: standard input
Output file: standard output
Time limit: 1 second
Memory limit: 1024 megabytes

书包喜欢在梦里打国际知名编程竞赛网站 CF (ColaFries) 上的比赛，并且每一次都能在梦里拿到个位数的 Rank。有一天，他突然对他每一次比赛后的 Rank 组成的数组的最长上升子序列感到好奇，于是，想出了下面的问题：

给定一个整数 x ，构造一个仅由数字字符 $0, 1, 2, \dots, 8, 9$ 构成且长度不超过 10^5 的字符串 s ，使得 s 的所有最长严格上升子序列的值的和为 x 。

由于书包梦醒之后是大笨蛋，于是想让你帮他解决这个问题。

一个仅由十进制的数字组成的字符串的值被定义为它所表示的十进制数的值。例如，字符串 001243 的所有最长严格上升子序列之和为 $0124 + 0123 + 0124 + 0123 = 494$ 。

字符串 t 是 s 的子序列，当且仅当 t 可以通过删除 s 中的若干个字符 (删除字符数可以为 0) 而不改变其他字符的顺序得到。例如 1011101 的子序列可以是 0、1、11111、101、0111，但不能是 000、101010、11100。

字符串 t 是 s 的严格上升子序列，当且仅当 t 是 s 的一个子序列，并且对于任意的 $1 \leq i < |t|$ ($|t|$ 表示 t 的长度)，均有 $t_i < t_{i+1}$ 。在这里，字符按其代表的数字比较大小。

字符串 t 是 s 的最长严格上升子序列，当且仅当 t 是 s 的一个严格上升子序列并且 s 没有比 t 更长的严格上升子序列。

Input

一行一个整数 x ($0 \leq x \leq 10^{13}$)，表示你需要构造的字符串的所有最长严格上升子序列的值的和。

Output

输出一个长度不超过 10^5 的字符串 s ，满足其所有最长严格上升子序列的值的和为 x 。

可以证明至少存在一种满足条件的答案。

如果存在多种符合条件的答案，输出其中任意一个。

Examples

standard input	standard output
494	001243
0	0

Note

第一个样例中，字符串 001243 的所有最长严格上升子序列的值的和 $0124 + 0123 + 0124 + 0123 = 494$

Problem D. 国际大胃王锦标赛

Input file: standard input
 Output file: standard output
 Time limit: 3 seconds
 Memory limit: 1024 megabytes

夜雀食堂举办了新一届国际大胃王锦标赛 (International Championship of Portion Consumption, 简称 ICPC)。虽然米斯蒂娅并未为去年破坏了比赛场地的白玉楼大学 (Whitejade House University, 简称 WHU) 分配参赛名额, 但该校还是擅自派出了一支队伍参赛, 其队长为幽幽子。

在长桌上有 n 个座位排成一排, 其中从左至右的第 i 个座位上放有一道份量为 a_i 的菜品。幽幽子初始位于第 s 个座位, 并且在每秒结束时能够移动至任意一个与当前座位相邻的座位 (她也可以待在原座位不动)。任何幽幽子到达过的座位上的菜品都会被她吃掉。

幽幽子想知道她能够吃掉的菜品的份量总和的最大值。请你对于所有满足 $1 \leq s \leq n, 1 \leq t \leq 2n$ 的正整数 s, t 求出幽幽子从第 s 个座位出发移动 t 秒时能吃掉的菜品的份量总和的最大值。

Input

第一行一个正整数 n ($1 \leq n \leq 5000$)。

第二行 n 个整数 a_1, a_2, \dots, a_n ($0 \leq a_i \leq 10^9$)。

Output

令 $F_{i,j}$ 为幽幽子从第 i 个座位出发, 移动 j 秒时能吃掉的菜品份量总和的最大值。为了验证你已求出满足 $1 \leq i \leq n, 1 \leq j \leq 2n$ 的全部 $F_{i,j}$ 的值, 请你输出一个整数 $\bigoplus_{i=1}^n \left(i + \bigoplus_{j=1}^{2n} j \cdot F_{i,j} \right)$ 。

其中 \oplus 为求异或和符号。具体地, 如果用 $a \oplus b$ 表示两个非负整数 a 和 b 的按位异或, 则 $\bigoplus_{i=1}^n x_i = x_1 \oplus x_2 \oplus \dots \oplus x_n$ (x_1, x_2, \dots, x_n 为非负整数)。

该输出方式仅是为了减小程序输出量, 与题目解法无关。

Examples

standard input	standard output
3 1 2 3	61
6 7 2 1 3 0 8	59

Note

第一个样例中, $F_{i,j}$ 的值为

$$F = \begin{bmatrix} 3 & 6 & 6 & 6 & 6 & 6 \\ 5 & 5 & 6 & 6 & 6 & 6 \\ 5 & 6 & 6 & 6 & 6 & 6 \end{bmatrix}$$

第二个样例中, $F_{i,j}$ 的值为

$$F = \begin{bmatrix} 9 & 10 & 13 & 13 & 21 & 21 & 21 & 21 & 21 & 21 & 21 \\ 9 & 9 & 10 & 14 & 14 & 21 & 21 & 21 & 21 & 21 & 21 \\ 4 & 10 & 12 & 13 & 14 & 14 & 21 & 21 & 21 & 21 & 21 \\ 4 & 11 & 13 & 13 & 13 & 14 & 21 & 21 & 21 & 21 & 21 \\ 8 & 8 & 11 & 13 & 14 & 21 & 21 & 21 & 21 & 21 & 21 \\ 8 & 11 & 12 & 14 & 21 & 21 & 21 & 21 & 21 & 21 & 21 \end{bmatrix}$$

Problem E. 回旋镖

Input file: standard input
Output file: standard output
Time limit: 1 second
Memory limit: 1024 megabytes

我是小蝶，我遇到了无数个我，虽然她们一点儿也不吵闹，但我还是带着她们去寻找我的朋友们，我必须弄明白这一切。

在云中城，我找到了我的朋友云宝，她是一匹帅气的天马，和我一点儿也不一样。遗憾的是，云宝也遇到了一个问题，她告诉我：

组织 WHU(World Horse Unite) 经常在新闻中传播一些消息，但是很遗憾，有些消息经过一段时间的传播后，被证实是一个假消息，为了挽回颜面，WHU 不得不重新散播一个消息进行辟谣。

“但是她们总是让我去辟谣”，云宝大声说，“可是我有点弄不清楚。”

整理了一下云宝的描述，简便起见，我认为消息在一棵 n 个点和 $n - 1$ 条边的无向无权树上传播。

WHU 散播的假消息自时间 0 开始从节点 r 开始传播，每个单位时间消息会扩散至相邻的点。形式化的，在时间 t 时，所有距离 r 不超过 t 的节点均接收到了这个消息，即点集 $V(r, t) = \{v | \text{dis}(r, v) \leq t\}$ ，其中 $\text{dis}(u, v)$ 表示树上两点 u 和 v 间的唯一简单路径的边数。

在时间 t_0 开始时，WHU 会委托云宝选择一个新的节点 r' 进行辟谣，每个单位时间辟谣会扩张至距离当前节点距离不超过 k 的节点上，即辟谣的速度是假消息的 k 倍。形式化的，在时间 t ($t \geq t_0$) 时，所有距离 r' 不超过 $k(t - t_0)$ 的节点接收到了辟谣，即点集 $V'(r', t) = \{v | \text{dis}(r', v) \leq k(t - t_0)\}$ 。

现在 r 和 t_0 都是确定的，但是云宝并不确定需要选择从哪里开始辟谣，也不确定用多快的速度辟谣，因此，我需要对所有满足 $1 \leq k \leq n$ 的 k 回答，任取 r' 时，假消息最早什么时候能全部被辟谣。形式化的，找到一个最早的时间点 t ，满足辟谣所覆盖的点完全包含了假消息所覆盖的点即 $V(r, t) \subseteq V'(r', t)$ 。



Input

第一行一个整数 n ($1 \leq n \leq 2 \times 10^5$)，表示树的点数。

接下来 $n - 1$ 行，每行两个整数 u, v ，表示树上的一条边。

接下来一行两个整数 r ($1 \leq r \leq n$)， t_0 ($1 \leq t_0 \leq n$)，表示上述题意中的参数。

Output

一行 n 个整数，第 i 个整数表示当 $k = i$ 时，上述询问的最早时间的值。

Examples

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

Note

第一个样例中，辟谣比假消息晚两个单位时间，假消息在 $t = 4$ 时将会覆盖全部的点。当 $k = 1$ 时，仅能选取 $r' = 3$ ，最早 $t = 4$ 时完成覆盖；当 $k = 2$ 时，可以选取 $r' = 2, 3$ ， $t = 3$ 时，假消息覆盖了点 $\{1, 2, 3, 4\}$ ，辟谣分别覆盖了点 $\{1, 2, 3, 4\}$ 和 $\{1, 2, 3, 4, 5\}$ ，均满足完全覆盖；当 $k = 3$ 时，可以选取 $r' = 2, 3, 4$ ， $t = 3$ 时，所有点都被辟谣覆盖；当 $k = 4$ 时，任取 r' ，总能时间 3 时所有点都被覆盖。

第二个样例中，对于 $k = 1, 2, \dots, n$ ，一种可行的 r' 选取为 $1, 2, \dots, 2$ 。

Problem F. 订制服装

Input file: standard input
Output file: standard output
Time limit: 1 second
Memory limit: 1024 megabytes

这是一道交互题。

哦，可怜的欧珀，看着团团围住白色波斯猫的小蝶们，我心里不禁叹息着。

珍奇总是很忙，而且说实话的，珍奇应该也很难解决无数个我这个问题，她不太精通这方面的魔法。

正在做订制服装的她，甚至反过来给了我一个问题，我将她的问题抽象了出来：

有一个 $n \times n$ 的正整数方阵 $a_{i,j}$ ($1 \leq a_{i,j} \leq n \times n$)，其满足若 $i > 1$ ，则 $a_{i,j} \geq a_{i-1,j}$ ；若 $j > 1$ ，则 $a_{i,j} \geq a_{i,j-1}$ 。

我只知道方阵的大小 n ，但不知道其中的元素内容，但是我可以询问方阵的某个位置 $a_{i,j}$ 是否不大于某个整数 x 。

我需要使用不超过 50000 次询问找出 $\{a_{i,j}\}$ 中的第 k 大值。

注意， $a_{i,j}$ 可能存在相同的值，第 k 大值被定义为将 $n \times n$ 个元素从大到小排序后的第 k 个元素的值。



Interaction Protocol

首先从标准输入读入给出的参数 n, k ($1 \leq n \leq 1000, 1 \leq k \leq n \times n$)。

接着可以进行不超过 50000 次询问，每次询问形如 "? i j x" ($1 \leq i, j \leq n, 1 \leq x \leq n \times n$)，并从标准输入读取回答。若 $a_{i,j} \leq x$ ，标准输入将返回 1，否则返回 0。

需要给出答案时，在单独的一行输出 "! x"，表示方阵的第 k 大值，回答不计入询问次数。给出答案后，立即结束程序。

输出一个询问后，不要忘记输出换行符并刷新输出。这在各个语言中可通过如下方式实现：

- `fflush(stdout)` 或 `cout.flush()` 对于 C++;
- `System.out.flush()` 对于 Java;
- `stdout.flush()` 对于 Python;

保证方阵在交互时不会发生变化。

Example

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

Note

样例中，一种可能的方阵为 $\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$ 。

Problem G. 包装

Input file: standard input
Output file: standard output
Time limit: 2 seconds
Memory limit: 1024 megabytes

你有 n 件 A 物品和 m 件 B 物品。每件 A 物品的价值均为 a ，每件 B 物品的价值均为 b 。

你现在需要确定一种商品的包装方案，包含若干件 A 物品和若干件 B 物品 (允许仅含 A 物品或仅含 B 物品)，满足物品的价值和为 k 。在该方案下，包装出尽可能多的商品，直到剩下的 A、B 物品不足以包装成一件商品。

请你求出在所有可能的包装方案中，A、B 物品剩余数量之和的最小值。

Input

有多组测试数据。第一行输入一个整数 T ($1 \leq T \leq 50$) 表示测试数据组数，对于每组测试数据：第一行五个正整数 n, m, a, b, k ($1 \leq n, m \leq 10^9, 1 \leq a, b \leq 10^9, 1 \leq k \leq 10^9$)。

Output

每组数据输出一行一个整数，表示 A、B 物品剩余数量之和的最小值。

Example

standard input	standard output
8	3
10 8 2 3 12	2
3 3 2 2 8	4
1 3 2 2 10	0
6 6 3 5 16	122
341 329 5 6 741	48
727 521 18 9 576	2764422
290646493 622766369 133 76 578504001	5914615
285261289 308082376 109 3 922747797	

Note

样例的第一组数据中，有 10 个价值为 2 的 A 物品和 8 个价值为 3 的 B 物品，可以按照 3 个 A 物品和 2 个 B 物品这样的构成包装成商品，这样能包装成 3 个商品，最后剩下 1 个 A 物品和 2 个 B 物品。

样例的第二组数据中，有 3 个价值为 2 的 A 物品和 3 个价值为 2 的 B 物品，可以按照 2 个 A 物品和 2 个 B 物品这样的构成包装成商品，这样能包装成 1 个商品，最后剩下 1 个 A 物品和 1 个 B 物品。

Problem H. 结晶之翼

Input file: standard input
Output file: standard output
Time limit: 3 seconds
Memory limit: 1024 megabytes

所有人都知道芙兰朵露有一对挂着彩色结晶的翅膀。然而，鲜有人知道芙兰朵露的翅膀其实是她自己制造的。

芙兰朵露有一棵由 n 个结晶构成的树，其中有 $n - 1$ 条边连接着结晶。这样，每个结晶都直接或间接地与其他结晶连通。第 i 个结晶的美丽度可以表示为 a_i 。

一对"翅膀"就是一个任意长度的无重复元素的序列 b_1, b_2, \dots, b_k ($1 \leq b_i \leq n$)，满足对于任意整数 $1 \leq i < k$ 树上的第 b_i 个结晶和第 b_{i+1} 个结晶都在树上由一条边相连。一对翅膀的美丽度就是其包含的所有结晶的美丽度的和的平方，即 $\left(\sum_{i=1}^k a_{b_i}\right)^2$ 。

芙兰想要用树上的结晶制造一些翅膀，保证每个结晶至多仅被用于一对翅膀。请你帮她求出她能制造出的翅膀的美丽度的和的最大值。

Input

第一行一个整数 n ($1 \leq n \leq 2 \times 10^5$)。

第二行 n 个整数 a_1, a_2, \dots, a_n ($0 \leq a_i \leq 5000$)。

接下来 $n - 1$ 行每行两个整数 u, v ($1 \leq u, v \leq n$)，表示树上一条连接第 u 个和第 v 个结晶的边。保证给出的边构成一棵树。

Output

输出一个整数，翅膀的美丽度的和的最大值。

Examples

standard input	standard output
1 19	361
3 100 100 500 1 2 1 3	490000
6 10 20 20 10 20 20 1 2 1 3 4 5 4 6 1 4	5000
9 0 1 0 1 1 0 1 1 1 1 2 1 3 1 4 1 5 2 6 2 7 3 8 6 9	14

Note

第一个样例中，芙兰朵露可以制造一对由唯一的一个结晶构成的翅膀，美丽度为 $19^2 = 361$ 。

第二个样例中，芙兰朵露可以制造一对由所有 n 个结晶构成的翅膀，美丽度为 $(100 + 100 + 500)^2 = 490000$ 。

第三个样例中，芙兰朵露可以制造两对翅膀 $[2, 1, 3]$ 和 $[5, 4, 6]$ 。它们的美丽度之和为 $(20 + 10 + 20)^2 + (20 + 10 + 20)^2 = 5000$ 。

Problem 1. 循环苹果串

Input file: standard input
Output file: standard output
Time limit: 1 second
Memory limit: 1024 megabytes

率领着小蝶们，我们来到甜苹果园。

“只见她压低身体，两个前蹄抵住地面，腰一扭，腿一蹬，两条强健的后腿如出膛子弹一般，打击在树干上。一声闷响夹带着树叶沙沙的声音，整树的苹果一个不落的掉进了筐里。”一个小蝶低声地说。

“小蝶，你怎么有这么多个？”，我会这样说话吗，我才刚升起这个疑惑，就被苹果嘉儿远远传过来的声音打断了。

“没什么不对的。”，随着另一个小蝶酷酷地说，苹果嘉儿筐子里的苹果也莫名其妙的一个个飞了起来。

奇怪，为什么有的我还会魔法？我连忙帮苹果嘉儿收拾苹果，并制止小蝶们的行动。

形式化的，制止行为可以被抽象成一个如下的问题：

给定一个长度为 n 的仅由 0 和 1 组成的字符串 s 。我可以进行如下操作任意次：选择 s 的一个子串和一个正整数 k ，并将子串向左循环移动 k 位。

字符串 a 是字符串 b 的子串，当且仅当 a 能通过删除 b 的开头和结尾的若干个字符 (可以为 0 个或全部) 得到。

将串 $s = s_0s_1 \dots s_{m-1}$ 向左循环移动 k 位，其会变为 $s_{k'}s_{k'+1} \dots s_{m-1}s_0s_1 \dots s_{k'-2}s_{k'-1}$ ，其中 $k' = k \bmod m$ 。

一个仅由 0 和 1 组成的字符串 s 是有序的，当且仅当对所有 $1 \leq i < n$ ，有 $s_i \leq s_{i+1}$ 。在这里，字符按其代表的数字比较大小。

例如，将串 0101100 的子串 1011 循环左移两位得到 0111000。

我要求出将 s 变为有序的所需的最小操作次数。



Input

一行一个仅由 0 和 1 组成的字符串 s ($1 \leq |s| \leq 10^5$)。

Output

输出一行一个整数，表示使 s 有序所需的最小操作次数。

Examples

standard input	standard output
01010101	3
11001010001	3

Note

第一个样例中，一种可行的最小操作是 $01010101 \rightarrow 00110101 \rightarrow 00011101 \rightarrow 00001111$ ，操作分别为对子串 $[1, 2]$ 左移 1 位，对子串 $[2, 4]$ 左移 2 位，对子串 $[3, 6]$ 左移 3 位。请注意，字符串的下标从 0 开始。

第二个样例中，一种可行的最小操作是 $1100101001 \rightarrow 0011101001 \rightarrow 0011100011 \rightarrow 0000011111$ ，操作分别为对子串 $[0, 3]$ 左移 2 位，对子串 $[6, 8]$ 左移 1 位，对子串 $[2, 7]$ 左移 3 位。

Problem J. 幻想乡高速公路

Input file: standard input
Output file: standard output
Time limit: 6 seconds
Memory limit: 1024 megabytes

夜雀食堂在幻想乡一共有 n 家分店，编号为 1 至 n 。在这 n 家分店之间有 m 条长度均为 1 的单向道路，其中第 i 条从 u_i 号分店通往 v_i 号分店。

为了加强分店之间的联系，密斯蒂娅决定再修建一些道路，于是她聘请了 k 支施工队。其中，密斯蒂娅会为第 i 支施工队均匀随机地在 $[1, m] \cap \mathbb{Z}$ 中选择一个整数 j ，并安排第 i 支施工队修建 a_i 条从 u_j 号分店通往 v_j 号分店的长度为 1 的单向道路。为所有施工队选择的整数 j 是彼此独立的。

夜雀食堂最大的两家分店是 1 号分店和 n 号分店。请你帮助密斯蒂娅计算道路修建完成后，从 1 号分店到 n 号分店的最短路的个数的期望值，答案对 998244353 取模。

Input

第一行三个整数 n, m, k ($2 \leq n \leq 2 \times 10^5, 1 \leq m \leq 2 \times 10^5, 1 \leq k \leq 2 \times 10^5$)。

接下来 m 行，第 i 行两个整数 u_i, v_i ($1 \leq u_i, v_i \leq n$)，表示一条从 u_i 号分店通往 v_i 号分店的单向道路。

接下来一行 k 个整数 a_1, a_2, \dots, a_k ($1 \leq a_i \leq 10^9$)，表示每支施工队修建道路的个数。

保证存在至少一条从 1 号分店到 n 号分店的路径。

图中可能存在重边、自环。

Output

输出一行一个整数，表示从 1 号分店到 n 号分店的最短路的个数的期望值，答案对 998244353 取模。

形式化地，令 $M = 998244353$ 。可以证明答案能表示为一个既约分数 $\frac{p}{q}$ ，其中 p 和 q 为整数且 $q \not\equiv 0 \pmod{M}$ 。你只需输出一个满足 $0 \leq x < M$ 且 $x \cdot q \equiv p \pmod{M}$ 的整数 x 。

Examples

standard input	standard output
4 4 1 1 2 1 3 2 4 3 4 2	4
3 5 2 1 1 1 2 1 2 2 1 2 3 5 3	399297752
9 8 5 1 3 3 2 1 4 4 2 4 5 2 9 5 9 7 8 1 2 3 4 5	308052020

Problem K. 派对游戏

Input file: standard input
Output file: standard output
Time limit: 2 seconds
Memory limit: 1024 megabytes

好不容易把甜苹果园的小蝶们收拢起来，我们来到萍琪派的糕点糖果铺。

“Wheee!!! 小蝶们，我们来办派对吧!”

没有办法，我们只好和萍琪派小姐玩了起来。

萍琪派发明了一种双人游戏，我们将轮流与她对决，具体的游戏内容如下：

有 n 个整数 $1, 2, 3, \dots, n$ 从左到右顺序排成一行，某个我和萍琪派将依次尝试进行如下操作：

- 若剩下的整数的异或和不为 0，移走这一行整数中最左边的数或最右边的数，并不改变其余数字的顺序。

若当前行动者无法操作，那么其输掉游戏。

游戏一共进行 T 轮，假设每轮“我”都先手且我们都以获胜为目标进行最优操作，问每轮游戏最终“我”是否能取得胜利。

若干个整数 a_1, a_2, \dots, a_m 的异或和为 $a_1 \oplus a_2 \oplus \dots \oplus a_m$ ，特别地，定义一个空数集的异或和为 0。

其中 \oplus 异或运算是一种二元运算，表示按位比较两个二进制数，在每个位置上，如果相应的位中不全为 1 或不全为 0，则结果为 1，否则为 0。



Input

第一行一个整数 T ($1 \leq T \leq 10^5$)，表示游戏轮数。

接下来 T 行，每行一个整数 n ($1 \leq n \leq 10^6$)，表示这轮游戏中整数的个数。

Output

对于每一轮游戏输出一行，若“我”能够取得胜利输出 "Fluttershy"，否则输出 "Pinkie Pie"（不含引号）。

Example

standard input	standard output
3	Fluttershy
1	Pinkie Pie
2	Pinkie Pie
3	

Note

样例的第一组数据中，“我”选择拿走 1，萍琪派无法操作，“我”获胜。

样例的第二组数据中，无论“我”选择拿走 1 还是 2，萍琪派都可以拿走剩下的数字，使得“我”无法操作，输掉游戏。

样例的第三组数据中，初始时异或和即为 0，“我”无法操作，输掉游戏。

Problem L. 魔法妖精

Input file: standard input
Output file: standard output
Time limit: 1 second
Memory limit: 1024 megabytes

“小蝶，这确实不是无序的把戏。”从中心城图书馆归来的紫悦对我说，“这只是一群调皮的魔法妖精。”
“魔法妖精？”
“她们是一种纯粹的魔法生物，可以随意地利用自然魔法场域自行构造外在躯壳。她们模仿了你的样子找到你，大概只是非常喜欢你。”
“是这样吗？”我望向小蝶们，她们微微点头，然后开始散发出微光，模糊了视线。
“结束了，当我们知道这一切的时候，但是她们留下了一个谜题。”我们一同注视着它们渐渐变成了一群闪烁着光芒的蝴蝶。
“我们一起来看看吧。”蝴蝶们在阳光中渐行渐远，最终融入了光影之中。

具体地，妖精们在数轴上搭起了 n 个宽度为 1 的紧挨着的矩形柱子，位置 i 的柱子高度为 h_i 。

特别地，在第一个柱子前侧和最后一个柱子后侧，被认为存在无限高度的柱子，即 h_0, h_{n+1} 视为无限高。同时，不存在两个柱子高度相同。

接下来给出 q 次独立的询问 x, V ，表示从 x 位置无限高处，垂直降下 V 单位面积的水，我们需要回答在水面稳定后，有多少个柱子顶端被水浸没，即其上覆盖了厚度大于 0 的水。

特别地，方便起见，保证所有的 V 是偶数。

水的流动：假定水是理想的流体，不存在大气压力，因此水总是倾向于流向严格意义上更低的地方，即使水处于平面柱面上也是如此。特别的，若水处于某个高度均大于两侧的柱子的柱面上，水将等量的流向两侧。由于柱子高度两两不同，因此这个情况仅会在最初降水的柱面上发生。



Input

第一行一个整数 n ($1 \leq n \leq 2 \times 10^5$)，表示柱子高度。

第二行 n 个整数 h_1, h_2, \dots, h_n ($0 \leq h_i \leq 10^9$)，表示柱子高度。保证柱子高度两两不同。

第三行一个整数 q ($1 \leq q \leq 2 \times 10^5$)，表示询问次数。

接下来 q 行，每行两个整数 x, V ($1 \leq x \leq n, 2 \leq V \leq 10^9$, V 是偶数)，表示下雨的位置和降雨量。

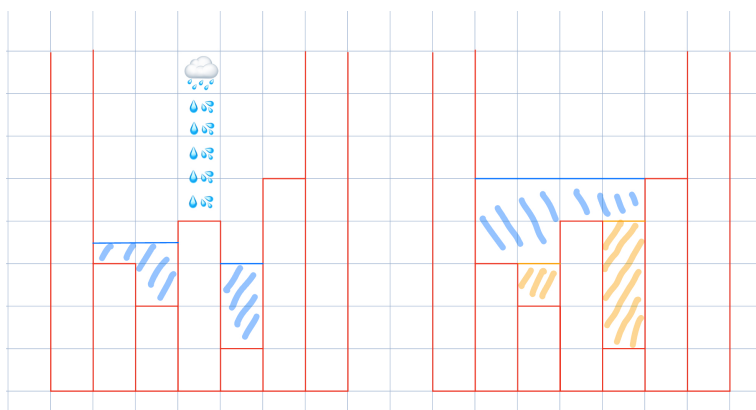
Output

输出 q 行，每行一个整数，代表每个询问的答案。

Example

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

Note



对于样例的第一个询问，左侧蓝色线标注了其水面线。4 单位面积的水等量的从位置 3 流向了两侧，最终位置 1,2,4 上具有积水，高度分别为 3.5,3.5,3。

对于样例的第二个询问，右侧蓝色线标注了其水面线。具体地，最终位置 1,2,3,4 覆盖有高度为 5 的水平面。

对于样例的第三个询问，右侧橙色线标注了其水面线。具体地，最终位置 2,4 覆盖有高度分别为 3,4 的水平面，注意到虽然有水曾经流过位置 3，但是最终位置 3 并没有积水，因此不存在覆盖。

Problem M. 合并

Input file: **standard input**
Output file: **standard output**
Time limit: 4 seconds
Memory limit: 1024 megabytes

Siri 在玩微信小游戏，由于是双人对抗小游戏，他喊来了大谷老师陪他一起玩游戏。

整个游戏流程如下，游戏开始时为双方生成若干士兵，Siri 将获得 n 位士兵，其中第 i 名士兵的战斗力为 a_i ；大谷老师也将获得若干名士兵。每一回合双方都将召唤出一名仍然存活的士兵进行战斗，战斗力高的一方将在消灭对方士兵的同时直接斩杀对方将军并获得游戏胜利，如果双方战斗力相等则双方士兵同时死亡并进行下一回合，最后如果双方士兵均已阵亡则判定为平局。

为了更轻松地获得游戏的胜利，Siri 植入了外挂，从而可以进行任意多次这样的操作：将两位战斗力相差恰好为 1 的士兵合并成为一位新的士兵，其战斗力为原来两位士兵的战斗力之和。

显然，在该游戏中，将两名玩家的士兵的战斗力组成的序列中的元素从大到小排序，序列的字典序大的玩家有必胜策略。请你帮助 Siri 利用外挂合并士兵，使得他的士兵的战斗力组成的序列在元素从大到小排序后字典序尽量大。你只需按任意顺序输出合并后士兵们的战斗力。

序列 $a = [a_1, a_2, \dots, a_{|a|}]$ 的字典序小于序列 $b = [b_1, b_2, \dots, b_{|b|}]$ 的字典序，当且仅当以下至少一项成立：

- a 为空序列且 b 不为空序列。
- a 和 b 均不为空序列且 $a_1 < b_1$ 。
- a 和 b 均不为空序列且 $a_1 = b_1$ 且 $[a_2, a_3, \dots, a_{|a|}]$ 的字典序小于 $[b_2, b_3, \dots, b_{|b|}]$ 的字典序。

Input

第一行两个整数 n ($1 \leq n \leq 2 \times 10^5$)，表示 Siri 的士兵数。

第二行 n 个整数 a_1, a_2, \dots, a_n ($1 \leq a_i \leq 10^{18}$)，其中第 i 个数 a_i 表示 Siri 的士兵中第 i 位士兵的战斗力。

Output

第一行一个整数 k ，表示合并后 Siri 士兵的数量。

第二行 k 个整数，表示合并后 Siri 的士兵们的战斗力。你可以按任意顺序输出士兵们的战斗力。

Examples

standard input	standard output
4 1 2 3 4	2 9 1
5 2 3 3 4 4	2 9 7
4 1 2 2 4	1 9

Note

第一个样例中，最优方案是

- 将 2,3 合并为 5
- 将 4,5 合并为 9

第二个样例中，最优方案是

- 将 2,3 合并为 5
- 将 4,5 合并为 9
- 将 3,4 合并为 7

第三个样例中，最优方案是

- 将 1,2 合并为 3
- 将 2,3 合并为 5
- 将 4,5 合并为 9