Problem A. IncDec Sequence

Time limit 1000 ms
Mem limit 131072 kB

Description

给定一个长度为 n 的数列 $a_1, a_2, \cdots, a_n$，每次可以选择一个区间$[l, r]$，使这个区间内的数都加 1 或者都减 1。

请问至少需要多少次操作才能使数列中的所有数都一样，并求出在保证最少次数的前提下，最终得到的数列有多少种。

Input

第一行一个正整数 n
接下来 n 行,每行一个整数,第 $i+1$行的整数表示 a_i。

Output

第一行输出最少操作次数
第二行输出最终能得到多少种结果

Sample 1

Input	Output
4 1 1 2 2	1 2

Hint

对于 100% 的数据，$n \le 100000, 0 \le a_i \le 2^{31}$。

Problem B. 灵茶八题 - 子数组 ^w+

Time limit 2000 ms
Mem limit 262144 kB

Description

给你一个长为 n 的数组 a，输出它的所有连续子数组的异或和的和。

例如 $a = [3, 5]$ 有三个连续子数组 $[3], [5], [3, 5]$，异或和分别为 $3, 5, 6$，所以答案为 $3 + 5 + 6 = 14$。

Input

第一行输入一个整数 n $(1 \le n \le 10^5)$。

第二行输入 n 个非负整数，表示数组 a 中的元素 $(0 \le a[i] \le 10^6)$。

Output

一个整数，表示 a 的所有连续子数组的异或和的和。

Sample 1

Input	Output
2 3 5	14

Sample 2

Input	Output
1 1	1

Hint

其余题目见《灵茶八题》题目列表

Problem C. 幽默的世界。

Time limit 1000 ms

Mem limit 524288 kB

Background

@【数据删除】：大家觉得呢 || @【数据删除】：oi 生活总是充满了幽默。 不过学文化课或许也好不了多少？

Description

给定一个长为 n 的序列 $a_1, a_2, \cdots, a_n$，定义 a 的一个连续子序列 $a_l, a_{l+1}, \cdots, a_r$ 是幽默的，当且仅当：

- $\sum\limits_{i=l}^{r} a_i > 0$；
- 对于所有 $l \le x \le y < r$，满足 $\sum\limits_{i=x}^{y} a_i \le 0$。

q 次询问，每次给定两个整数 l, r，查询满足以下条件的数对 (l', r') 个数：

- $l \le l' \le r' \le r$；
- 连续子序列 $a_{l'}, a_{l'+1}, \cdots a_{r'}$ 是幽默的。

Input

第一行输入两个整数 n, q。

接下来一行输入 n 个整数，第 i 个整数代表 a_i。

接下来 q 行，每行输入两个整数 l, r，代表一次询问。

Output

对于每组询问，输出一行一个整数，代表答案。

Sample 1

Input	Output
`4 3`	`1`
`3 -4 -1 2`	`2`
`1 2`	`3`
`3 4`	
`1 4`	

Sample 2

Input	Output
`7 6`	`1`
`-1 2 -1 -1 -1 2 -1`	`2`
`2 5`	`4`
`4 7`	`0`
`1 7`	`2`
`5 5`	`1`
`1 3`	
`2 4`	

Hint

对于所有数据，保证 $1 \le n, q \le 2 \times 10^5$，$1 \le l \le r \le n$，$|a_i| \le 10^9$。

子任务

#	特殊性质	分值
0	样例	0
1	$n, q \le 50$	15
2	$n, q \le 3 \times 10^3$	20
3	对于所有询问，$r = n$	15
4	对于所有询问，$l = 1$	15
5	-	35

Problem D. Pond

Time limit 2000 ms
Mem limit 1048576 kB

Problem Statement

The land of a park *AtCoder* is an $N \times N$ grid with east-west rows and north-south columns. The height of the square at the i-th row from the north and j-th column from the west is given as $A_{i,j}$.

Takahashi, the manager, has decided to build a square pond occupying $K \times K$ squares in this park.

To do this, he wants to choose a square section of $K \times K$ squares completely within the park whose **median** of the heights of the squares is the lowest. Find the median of the heights of the squares in such a section.

Here, the **median** of the heights of the squares in a $K \times K$ section is the height of the $(\left\lfloor \frac{K^2}{2} \right\rfloor + 1)$-th highest square among the K^2 squares in the section, where $\lfloor x \rfloor$ is the greatest integer not exceeding x.

Constraints

- $1 \leq K \leq N \leq 800$
- $0 \leq A_{i,j} \leq 10^9$
- All values in input are integers.

Input

Input is given from Standard Input in the following format:

$$\begin{array}{l} N \quad K \\ A_{1,1} \quad A_{1,2} \quad \ldots \quad A_{1,N} \\ A_{2,1} \quad A_{2,2} \quad \ldots \quad A_{2,N} \\ : \\ A_{N,1} \quad A_{N,2} \quad \ldots \quad A_{N,N} \end{array}$$

Output

Print the answer.

Sample 1

Input	Output

3 2
1 7 0
5 8 11
10 4 2
	4

Let (i, j) denote the square at the i-th row from the north and j-th column from the west. We have four candidates for the 2×2 section occupied by the pond:
$\{(1,1),(1,2),(2,1),(2,2)\},\{(1,2),(1,3),(2,2),(2,3)\},\{(2,1),(2,2),(3,1),(3,2)\},\{(2,2),(2,3),(3,2),(3,3)\}$
.
When $K = 2$, since $\left\lfloor \frac{2^2}{2} \right\rfloor + 1 = 3$, the median of the heights of the squares in a section is the height of the 3-rd highest square, which is 5, 7, 5, 4 for the candidates above, respectively. We should print the lowest of these: 4.

Sample 2

Input	Output

3 3
1 2 3
4 5 6
7 8 9
	5

Problem E. White Pawn

Time limit 2000 ms
Mem limit 1048576 kB

Problem Statement

Let N be a positive integer. We have a $(2N+1) \times (2N+1)$ grid where rows are numbered 0 through $2N$ and columns are also numbered 0 through $2N$. Let (i, j) denote the square at Row i and Column j.

We have one white pawn, which is initially at $(0, N)$. Also, we have M black pawns, the i-th of which is at (X_i, Y_i).

When the white pawn is at (i, j), you can do one of the following operations to move it:

- If $0 \leq i \leq 2N-1, 0 \leq j \leq 2N$ hold and $(i+1, j)$ **does not** contain a black pawn, move the white pawn to $(i+1, j)$.
- If $0 \leq i \leq 2N-1, 0 \leq j \leq 2N-1$ hold and $(i+1, j+1)$ **does** contain a black pawn, move the white pawn to $(i+1, j+1)$.
- If $0 \leq i \leq 2N-1, 1 \leq j \leq 2N$ hold and $(i+1, j-1)$ **does** contain a black pawn, move the white pawn to $(i+1, j-1)$.

You cannot move the black pawns.

Find the number of integers Y such that it is possible to have the white pawn at $(2N, Y)$ by repeating these operations.

Constraints

- $1 \leq N \leq 10^9$
- $0 \leq M \leq 2 \times 10^5$
- $1 \leq X_i \leq 2N$
- $0 \leq Y_i \leq 2N$
- $(X_i, Y_i) \neq (X_j, Y_j)$ $(i \neq j)$
- All values in input are integers.

Input

Input is given from Standard Input in the following format:

$N \ M$
$X_1 \ Y_1$
$\vdots$
$X_M \ Y_M$

Output

Print the answer.

Sample 1

Input	Output

2 4
1 1
1 2
2 0
4 2
	3

We can move the white pawn to $(4, 0)$, $(4, 1)$, and $(4, 2)$, as follows:

- $(0, 2) \to (1, 1) \to (2, 1) \to (3, 1) \to (4, 2)$
- $(0, 2) \to (1, 1) \to (2, 1) \to (3, 1) \to (4, 1)$
- $(0, 2) \to (1, 1) \to (2, 0) \to (3, 0) \to (4, 0)$

On the other hand, we cannot move it to $(4, 3)$ or $(4, 4)$. Thus, we should print 3.

Sample 2

Input	Output
1 1 1 1	0

We cannot move the white pawn from $(0, 1)$.

Problem F. Red-Blue Operations (Hard Version)

Time limit 2000 ms
Mem limit 262144 kB

The only difference between easy and hard versions is the maximum values of n and q.

You are given an array, consisting of n integers. Initially, all elements are red.

You can apply the following operation to the array multiple times. During the i-th operation, you select an element of the array; then:

- if the element is red, it increases by i and becomes blue;
- if the element is blue, it decreases by i and becomes red.

The operations are numbered from 1, i. e. during the first operation some element is changed by 1 and so on.

You are asked q queries of the following form:

- given an integer k, what can the largest minimum in the array be if you apply **exactly** k operations to it?

Note that the operations don't affect the array between queries, all queries are asked on the initial array a.

Input

The first line contains two integers n and q ($1 \le n, q \le 2 \cdot 10^5$) — the number of elements in the array and the number of queries.

The second line contains n integers $a_1, a_2, \ldots, a_n$ ($1 \le a_i \le 10^9$).

The third line contains q integers $k_1, k_2, \ldots, k_q$ ($1 \le k_j \le 10^9$).

Output

For each query, print a single integer — the largest minimum that the array can have after you apply **exactly** k operations to it.

Examples

Input	Output

4 10
5 2 8 4
1 2 3 4 5 6 7 8 9 10
	3 4 5 6 7 8 8 10 8 12

Input	Output

5 10
5 2 8 4 4
1 2 3 4 5 6 7 8 9 10
	3 4 5 6 7 8 9 8 11 8

Input	Output

2 5
2 3
10 6 8 1 3
	10 7 8 3 3

Problem G. A Game with Traps

Time limit 3000 ms
Mem limit 262144 kB

You are playing a computer game, where you lead a party of m soldiers. Each soldier is characterised by his agility a_i.

The level you are trying to get through can be represented as a straight line segment from point 0 (where you and your squad is initially located) to point $n+1$ (where the boss is located).

The level is filled with k traps. Each trap is represented by three numbers l_i, r_i and d_i. l_i is the location of the trap, and d_i is the danger level of the trap: whenever a soldier with agility lower than d_i steps on a trap (that is, moves to the point l_i), he gets instantly killed. Fortunately, you can disarm traps: if you move to the point r_i, you disarm this trap, and it no longer poses any danger to your soldiers. Traps don't affect you, only your soldiers.

You have t seconds to complete the level — that is, to bring some soldiers from your squad to the boss. Before the level starts, you choose which soldiers will be coming with you, and which soldiers won't be. After that, you have to bring **all of the chosen soldiers** to the boss. To do so, you may perform the following actions:

- if your location is x, you may move to $x+1$ or $x-1$. This action consumes one second;
- if your location is x and the location of your squad is x, you may move to $x+1$ or to $x-1$ with your squad in one second. You may not perform this action if it puts some soldier in danger (i. e. the point your squad is moving into contains a non-disarmed trap with d_i greater than agility of some soldier from the squad). This action consumes one second;
- if your location is x and there is a trap i with $r_i = x$, you may disarm this trap. This action is done instantly (it consumes no time).

Note that after each action both your coordinate and the coordinate of your squad should be integers.

You have to choose the maximum number of soldiers such that they all can be brought from the point 0 to the point $n+1$ (where the boss waits) in no more than t seconds.

Input

The first line contains four integers m, n, k and t ($1 \le m, n, k, t \le 2 \cdot 10^5$, $n < t$) — the number of soldiers, the number of integer points between the squad and the boss, the number of traps and the maximum number of seconds you may spend to bring the squad to the boss, respectively.

The second line contains m integers a_1, a_2, ..., a_m ($1 \le a_i \le 2 \cdot 10^5$), where a_i is the agility of the i-th soldier.

Then k lines follow, containing the descriptions of traps. Each line contains three numbers l_i, r_i and d_i ($1 \le l_i \le r_i \le n$, $1 \le d_i \le 2 \cdot 10^5$) — the location of the trap, the location where the trap can be disarmed, and its danger level, respectively.

Output

Print one integer — the maximum number of soldiers you may choose so that you may bring them all to the boss in no more than t seconds.

Examples

Input	Output

```
5 6 4 14
1 2 3 4 5
1 5 2
1 2 5
2 3 5
3 5 3
```
	3

Note

In the first example you may take soldiers with agility 3, 4 and 5 with you. The course of action is as follows:

- go to 2 without your squad;
- disarm the trap 2;
- go to 3 without your squad;
- disartm the trap 3;
- go to 0 without your squad;
- go to 7 with your squad.

The whole plan can be executed in 13 seconds.

Problem H. Silver Fox vs Monster

Time limit	2000 ms
Mem limit	1048576 kB

Problem Statement

Silver Fox is fighting with N monsters.

The monsters are standing in a row, and we can assume them to be standing on a number line. The i-th monster, standing at the coordinate X_i, has the *health* of H_i.

Silver Fox can use bombs to attack the monsters. Using a bomb at the coordinate x decreases the healths of all monsters between the coordinates $x - D$ and $x + D$ (inclusive) by A. There is no way other than bombs to decrease the monster's health.

Silver Fox wins when all the monsters' healths become 0 or below.

Find the minimum number of bombs needed to win.

Constraints

- $1 \leq N \leq 2 \times 10^5$
- $0 \leq D \leq 10^9$
- $1 \leq A \leq 10^9$
- $0 \leq X_i \leq 10^9$
- $1 \leq H_i \leq 10^9$
- X_i are distinct.
- All values in input are integers.

Input

Input is given from Standard Input in the following format:

```
N D A
X_1 H_1
:
X_N H_N
```

Output

Print the minimum number of bombs needed to win.

Sample 1

Input	Output

3 3 2
1 2
5 4
9 2
	2

First, let us use a bomb at the coordinate 4 to decrease the first and second monsters' health by 2.

Then, use a bomb at the coordinate 6 to decrease the second and third monsters' health by 2 .

Now, all the monsters' healths are 0. We cannot make all the monsters' health drop to 0 or below with just one bomb.

Sample 2

Input	Output

9 4 1
1 5
2 4
3 3
4 2
5 1
6 2
7 3
8 4
9 5
	5

We should use five bombs at the coordinate 5.

Sample 3

Input	Output

3 0 1
300000000 1000000000
100000000 1000000000
200000000 1000000000
	3000000000

Watch out for overflow.

Problem I. 2-variable Function

Time limit	1000 ms
Mem limit	131072 kB

Given an integer n, find the smallest integer x that satisfies two conditions:

- x is greater than or equal to n;
- There is a pair of non-negative integers (a, b) such that $x = a^3 + a^2 \cdot b + a \cdot b^2 + b^3$.

Input

One nonnegative integer n $(n \leq 10^{18})$.

Output

Print the smallest value of x.

Examples

Input	Output
9	15

Input	Output
0	0

Problem J. 合并饭团

Time limit 1000 ms
Mem limit 256000 kB

Description

译自 CCC2016 Senior T4「Combining Riceballs」

Alphonse 有 N 个美味的饭团，它们大小不一，摆放成一行。他想把最大的饭团让给自己的基友。他可以执行以下操作：

- 如果两个**相邻的**饭团大小相同，Alphonse 可以把它们合并成一个新的饭团。新饭团的大小是两个原饭团大小之和。它将占据两个原饭团先前占据的位置。
- 如果两个饭团大小相同，且它们之间只有一个饭团，Alphonse 也可以把它们合并成一个新的饭团。（中间的饭团大小没有规定。）新饭团的大小是三个原饭团大小之和，并占据三个原饭团先前的位置。

Alphonse 可以按照他的意愿执行任意次操作。

在执行 0 或更多次操作后，确定他应该把哪个饭团让给基友。

Input

第一行，一个整数 $N(1 \leq N \leq 400)$。

第二行，N 个以空格分隔的整数，表示每个饭团的大小，按照从左到右的顺序给出。每个整数的上界为 1 000 000。

Output

输出 Alphonse 可以得到的最大的饭团大小。

Sample 1

Input	Output
7 47 12 12 3 9 9 3	48

Sample 2

Input	Output
4 1 2 3 1	3

Hint

样例解释 1

有一种可能的合并方案为：合并大小同为 12 的两个饭团，得到一个大小为 24 的饭团。然后合并大小同为 9 的两个饭团，得到一个大小为 18。接着合并大小为 $3, 18$ 和 3 的三个饭团，得到一个大小为 24 的饭团。最后合并大小同为 24 的两个饭团，得到一个大小为 48 的饭团。

样例解释 2

我们无法进行操作，所以答案为 3。

对于 $\frac{1}{15}$ 的数据，$N = 4$。

对于另外 $\frac{2}{15}$ 的数据，$N \leq 10$。

对于另外 $\frac{5}{15}$ 的数据，$N \leq 50$。

Problem K. 可持久化线段树 2

Time limit	1000 ms
Mem limit	1048576 kB

Background

这是个非常经典的可持久化权值线段树入门题——静态区间第 k 小。

数据已经过加强，请使用可持久化权值线段树。同时请注意常数优化。

Description

如题，给定 n 个整数构成的序列 a，将对于指定的闭区间 $[l, r]$ 查询其区间内的第 k 小值。

Input

第一行包含两个整数，分别表示序列的长度 n 和查询的个数 m。
第二行包含 n 个整数，第 i 个整数表示序列的第 i 个元素 a_i。
接下来 m 行每行包含三个整数 l, r, k , 表示查询区间 $[l, r]$ 内的第 k 小值。

Output

对于每次询问，输出一行一个整数表示答案。

Sample 1

Input	Output

5 5
25957 6405 15770 26287 26465
2 2 1
3 4 1
4 5 1
1 2 2
4 4 1
	6405
	15770
	26287
	25957
	26287

Hint

样例 1 解释

$n = 5$，数列长度为 5，数列从第一项开始依次为$\{25957, 6405, 15770, 26287, 26465\}$。

- 第一次查询为 $[2, 2]$ 区间内的第一小值，即为 6405。
- 第二次查询为 $[3, 4]$ 区间内的第一小值，即为 15770。
- 第三次查询为 $[4, 5]$ 区间内的第一小值，即为 26287。
- 第四次查询为 $[1, 2]$ 区间内的第二小值，即为 25957。
- 第五次查询为 $[4, 4]$ 区间内的第一小值，即为 26287。

数据规模与约定

- 对于 20% 的数据，满足 $1 \le n, m \le 10$。
- 对于 50% 的数据，满足 $1 \le n, m \le 10^3$。
- 对于 80% 的数据，满足 $1 \le n, m \le 10^5$。
- 对于 100% 的数据，满足 $1 \le n, m \le 2 \times 10^5$，$0 \le a_i \le 10^9$，$1 \le l \le r \le n$，$1 \le k \le r - l + 1$。

Problem L. Dynamic Rankings

Time limit 3000 ms

Mem limit 524288 kB

Description

给定一个含有 n 个数的序列 $a_1, a_2 \ldots a_n$，需要支持两种操作：

- `Q l r k` 表示查询下标在区间 $[l, r]$ 中的第 k 小的数
- `C x y` 表示将 a_x 改为 y

Input

第一行两个正整数 n, m，表示序列长度与操作个数。
第二行 n 个整数，表示 $a_1, a_2 \ldots a_n$。
接下来 m 行，每行表示一个操作，都为上述两种中的一个。

Output

对于每一次询问，输出一行一个整数表示答案。

Sample 1

Input	Output
5 3 3 2 1 4 7 Q 1 4 3 C 2 6 Q 2 5 3	3 6

Hint

【数据范围】

对于 10% 的数据，$1 \leq n, m \leq 100$；
对于 20% 的数据，$1 \leq n, m \leq 1000$；
对于 50% 的数据，$1 \leq n, m \leq 10^4$；

对于 100% 的数据，$1 \le n,m \le 10^5$，$1 \le l \le r \le n$，$1 \le k \le r-l+1$，$1 \le x \le n$，$0 \le a_i, y \le 10^9$。

请注意常数优化，但写法正常的整体二分和树套树都可以以大约 1000ms 每个点的时间通过。

来源：bzoj1901

本题数据为洛谷自造数据，使用CYaRon耗时5分钟完成数据制作。

Problem M. 网络

Time limit 2000 ms
Mem limit 131072 kB

Description

一个简单的网络系统可以被描述成一棵无根树。每个节点为一个服务器。连接服务器与服务器的数据线则看做一条树边。两个服务器进行数据的交互时，数据会经过连接这两个服务器的路径上的所有服务器（包括这两个服务器自身）。

由于这条路径是唯一的，当路径上的某个服务器出现故障，无法正常运行时，数据便无法交互。此外，每个数据交互请求都有一个重要度，越重要的请求显然需要得到越高的优先处理权。现在，你作为一个网络系统的管理员，要监控整个系统的运行状态。系统的运行也是很简单的，在每一个时刻，只有可能出现下列三种事件中的一种：

1. 在某两个服务器之间出现一条新的数据交互请求；
2. 某个数据交互结束请求；
3. 某个服务器出现故障。系统会在任何故障发生后立即修复。也就是在出现故障的时刻之后，这个服务器依然是正常的。但在服务器产生故障时依然会对需要经过该服务器的数据交互请求造成影响。

你的任务是在每次出现故障时，维护未被影响的请求中重要度的最大值。注意，如果一个数据交互请求已经结束，则不将其纳入未被影响的请求范围。

Input

第一行两个正整数 n, m，分别描述服务器和事件个数。服务器编号是从 1 开始的，因此 n 个服务器的编号依次是 $1, 2, 3, \cdots, n$。

接下来 $n-1$ 行，每行两个正整数 u, v，描述一条树边。u 和 v 是服务器的编号。

接下来 m 行，按发生时刻依次描述每一个事件；即第 i 行（$i = 1, 2, 3, ..., m$）描述时刻 i 发生的事件。每行的第一个数 type 描述事件类型，共 3 种类型：

1. 若 $\text{type} = 0$，之后有三个正整数 a, b, v，表示服务器 a, b 之间出现一条重要度为 v 的数据交互请求；
2. 若 $\text{type} = 1$，之后有一个正整数 t，表示时刻 t（也就是第 t 个发生的事件）出现的数据交互请求结束；
3. 若 $\text{type} = 2$，之后有一个正整数 x，表示服务器 x 在这一时刻出现了故障。

对于每个 type 为 2 的事件，就是一次询问，即询问“当服务器 x 发生故障时，未被影响的请求中重要度的最大值是多少？”注意可能有某个服务器自身与自身进行数据交互的情况。$2 \le n \le 10^5$，$1 \le m \le 2 \times 10^5$，其他的所有输入值不超过 10^9。

Output

对于每个 $\text{type} = 2$ 的事件，即服务器出现故障的事件，输出一行一个整数，描述未被影响的请求中重要度的最大值。如果此时没有任何请求，或者所有请求均被影响，则输出 -1。

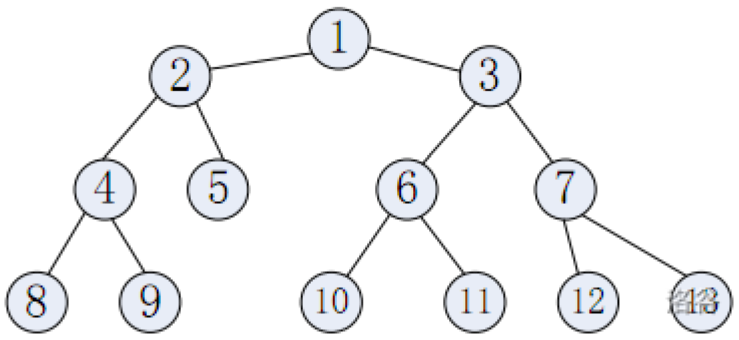
Sample 1

Input	Output

13 23
1 2
1 3
2 4
2 5
3 6
3 7
4 8
4 9
6 10
6 11
7 12
7 13
2 1
0 8 13 3
0 9 12 5
2 9
2 8
2 2
0 10 12 1
2 2
1 3
2 7
2 1
0 9 5 6
2 4
2 5
1 7
0 9 12 4
0 10 5 7
2 1
2 4
2 12
1 2
2 5
2 3
	-1
	3
	5
	-1
	1
	-1
	1
	1
	3
	6
	7
	7
	4
	6

Hint

样例给出的树如下所示：



解释其中的部分询问；下面的解释中用 $(a, b; t, v)$ 表示在 t 时刻出现的服务器 a 和 b 之间的重要度为 v 的请求：

对于第一个询问（在时刻 1），此时没有任何请求，输出 -1。

对于第四个询问（在时刻 6），此时有两条交互$(8, 13; 2, 3), (9, 12; 3, 5)$，所有询问均经过 2 号服务器，输出 -1。

对于第五个询问（在时刻 8），此时有三条交互 $(8, 13; 2, 3), (9, 12; 3, 5), (10, 12; 7, 1)$，只有交互 $(10, 12; 7, 1)$ 没有经过 2 号服务器，因此输出其重要度 1。

对于最后一个询问（在时刻 23），此时有三条交互 $(9, 5; 12, 6), (9, 12; 16, 4), (10, 5; 17, 7)$。当 3 号服务器出现故障时，只有交互 $(9, 5; 12, 6)$ 没有经过 3 号服务器，因此输出 6。

upd 2016.5.20：新加一组 Hack 数据。

Problem N. MET-Meteors

Time limit 2000 ms
Mem limit 524288 kB

Description

Byteotian Interstellar Union (BIU) has recently discovered a new planet in a nearby galaxy. The planet is unsuitable for colonisation due to strange meteor showers, which on the other hand make it an exceptionally interesting object of study.

The member states of BIU have already placed space stations close to the planet's orbit. The stations' goal is to take samples of the rocks flying by.

The BIU Commission has partitioned the orbit into m sectors, numbered from 1 to m, where the sectors 1 and m are adjacent. In each sector there is a single space station, belonging to one of the n member states.

Each state has declared a number of meteor samples it intends to gather before the mission ends. Your task is to determine, for each state, when it can stop taking samples, based on the meter shower predictions for the years to come.

Input

The first line of the standard input gives two integers, n and m $(1 \leq n, m \leq 300\ 000)$, separated by a single space, that denote,respectively, the number of BIU member states and the number of sectors the orbit has been partitioned into.

In the second line there are m integers o_i $(1 \leq o_i \leq n)$,separated by single spaces, that denote the states owning stations in successive sectors.

In the third line there are n integers p_i $(1 \leq p_i \leq 10^9)$,separated by single spaces, that denote the numbers of meteor samples that the successive states intend to gather.

In the fourth line there is a single integer k $(1 \leq k \leq 300\ 000)$ that denotes the number of meteor showers predictions. The following k lines specify the (predicted) meteor showers chronologically. The i-th of these lines holds three integers l_i, r_i, a_i (separated by single spaces), which denote that a meteor shower is expected in sectors $l_i, l_{i+1}, ..., r_i$(if $l_i \leq r_i$) or sectors $l_i, l_{i+1}, ..., m, 1, ..., r_i$ (if $l_i > r_i$) , which should provide each station in those sectors with a_i meteor samples ($1 \leq a_i \leq 10^9$).

Output

Your program should print n lines on the standard output.

The i-th of them should contain a single integer w_i, denoting the number of shower after which the stations belonging to the i-th state are expected to gather at least p_i samples, or the word NIE (Polish for no) if that state is not expected to gather enough samples in the foreseeable future.

Sample 1

Input	Output
3 5	3
1 3 2 1 3	NIE
10 5 7	1
3	
4 2 4	
1 3 1	
3 5 2	