Problem A. Sparse table

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	256 megabytes

You are given an array consisting of n integers. Write a program that answers the queries of the following type: find the minimum between u-th and v-th element, inclusive.

Input

The first line contains three integers: n, m and a_1 ($1 \leqslant n \leqslant 10^5$; $1 \leqslant m \leqslant 10^7$; $0 \leqslant a_1 < 16\,714\,589$) — the number of integers in the given array, the number of queries, and the first element of the given array, respectively.

The second line contains two integers u_1, and v_1 ($1 \leqslant u_1, v_1 \leqslant n$) — the first query.

For the sake of the input size, the array and the queries should be generated.

The array elements $a_2, a_3, \ldots, a_n$ are generated with the following formula:

$$a_{i+1} = (23 \cdot a_i + 21563) \bmod 16714589.$$

For instance, if $n = 10$, $a_1 = 12345$ the following array should be generated: $a = (12345, 305498, 7048017, 11694653, 1565158, 2591019, 9471233, 570265, 13137658, 1325095)$.

The queries are generated in the following way:

$$u_{i+1} = \big((17 \cdot u_i + 751 + r_i + 2i) \bmod n\big) + 1,$$
$$v_{i+1} = \big((13 \cdot v_i + 593 + r_i + 5i) \bmod n\big) + 1,$$

where r_i — the answer for query i.

Be careful, u_i can be greater than v_i.

Output

Print three integers u_m, v_m and r_m (the last query, and the answer to it).

Examples

standard input	standard output
10 8 12345 3 9	5 3 1565158

Note

Notice that you can get rid of saving u, v, and r arrays into memory.

The queries, and the answers to them:

i	u_i	v_i	r_i
1	3	9	570265
2	10	1	12345
3	1	2	12345
4	10	10	1325095
5	5	9	570265
6	2	1	12345
7	3	2	305498
8	5	3	1565158

This problem probably is not solvable by using standard Python 2 and Python 3 interpreters. Use PyPy compilers instead.

Problem B. LCA

Input file:	`standard input`
Output file:	`standard output`
Time limit:	5 seconds
Memory limit:	256 megabytes

You are given a tree rooted for the first vertex. You need to answer m queries "`find the LCA of two vertices`".

LCA of two vertices u and v in a rooted tree is the most distant from root vertex that lies on both paths from u and from v to the root.

Input

The first line of input contains one integer n — the number of vertices in the tree ($1 \le n \le 2 \cdot 10^5$).

Each of next $n - 1$ lines contains one integer x. Integer x on line i means that x is a parent of vertex i ($x < i$).

Then you are given an integer m.

Then you are given m ($0 \le m \le 5 \cdot 10^5$) queries (u, v) — find the LCA of vertices u and v ($1 \le u, v \le n$; $u \ne v$).

Output

For each query, print the LCA of two vertices in a separate line.

Examples

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

Problem C. The cheapest edge

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	256 megabytes

You are given a rooted tree, with vertex 1 as the root. All edges have cost. You are asked to answer m queries: find the minimum cost edge on the path between two vertices.

Input

The first line contains a single integer n — the number of vertices.

Each of the following $n-1$ lines contains two integers x and y. The meaning of x and y on the i-th of these lines is that x is the parent of vertex $i+1$, and y is the cost of the edge between x and $i+1$.

$x \le i, |y| \le 10^6$.

The following line contains an integer m — the number of queries.

Then m lines containing queries are given, each of them consists of two integers x and y: find the minimum cost edge on the path between vertices x and y ($x \neq y$).

The constraints: $2 \le n \le 5 \cdot 10^4, 0 \le m \le 5 \cdot 10^4$.

Output

Print m lines — the answers to each query.

Example

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

Problem D. Rectangles

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	256 megabytes

You are given a grid T of size $N \times M$. Each element of this grid is a rectangle T_{ij}, for $0 \le i < N$ and $0 \le j < M$. Rectangle T_{ij} is described by four integers (x_1^{ij}, y_1^{ij}, x_2^{ij}, y_2^{ij}), (x_1^{ij}, y_1^{ij}) and (x_2^{ij}, y_2^{ij}) are the coordinates of the corners of rectangle. Sides of the rectangle are parallel to the axis.

Then you are given queries. Each query consists of four integers: (r_1, c_1, r_2, c_2). Answer to this query is the area of a shape, equal to the intersection of all such rectangles T_{ij}, that $\min(r_1, r_2) \le i \le \max(r_1, r_2)$ and $\min(c_1, c_2) \le j \le \max(c_1, c_2)$. There are a lot of queries, so we ask you to find the sum of answers to all queries, modulo $10^9 + 7$.

Input

The first line contains two integers N and M — the dimensions of T. ($1 \le N, M \le 127$). The next N lines contain a description of T: in the $(i+1)$-th line, the $(j+1)$-th tuple of four integers x_1^{ij} y_1^{ij} x_2^{ij} y_2^{ij} is describing the rectangle T_{ij}. It is guaranteed that $|x_k^{ij}|, |y_k^{ij}| \le 10^6$.

In the separate line, you are given four integers. The first one of them is an integer Q — the number of queries ($1 \le Q \le 5 \cdot 10^6$). The next three integers are A, B, v_0 ($0 \le A, B, v_0 < 10^9 + 7$). Using these integers, the sequence $\{v_i\}$ is generated, $v_i = (A \cdot v_{i-1} + B) \bmod (10^9 + 7)$.

Then, the k-th query (the queries are one-indexed) is described by four integers: ($v_{4k-3} \bmod N, v_{4k-2} \bmod M, v_{4k-1} \bmod N, v_{4k} \bmod M$).

Output

Print the sum of answers to all queries, modulo $10^9 + 7$.

Examples

standard input	standard output
2 2 0 0 2 2 1 1 3 3 0 3 2 1 1 2 3 0 1 500000003 4 2	1
3 2 8 -1 -7 6 6 8 9 10 -4 -10 4 9 -3 -8 6 9 -2 -9 3 8 -5 7 7 3 5 303164476 273973578 65779139	85

Note

In the first example, the query is of the form `(1,0,0,1)`, so this query is about the whole grid. The intersection of all rectangles in the grid is a square with corners in points $(1, 1)$ and $(2, 2)$. Its area is equal to 1.

In the second example, queries are `(0,1,1,1)`, `(1,0,2,0)`, `(0,0,2,1)`, `(0,1,1,1)`, `(0,1,0,0)`. The answer to the first query is 85, the answers to all other queries are 0.

Problem E. Path increment

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	256 megabytes

You are given a tree. Each vertex has a variable in it. Initially the value of each variable is zero. You need to answer two types of queries: add a number to each vertex on some path, and print the variable.

Input

The first line contains an integer n — the number of vertices in the tree ($1 \le n \le 3 \cdot 10^5$).

Each of the following $n-1$ lines contains two integers v and u — the edge between v and u ($1 \le v, u \le n$).

The following line contains an integer m — the number of queries ($1 \le m \le 5 \cdot 10^5$).

The following m lines describe queries, one query in a line:

- `+ v u d` — add d to each variable on a simple path between vertices v and u ($1 \le v, u \le n$; $1 \le d \le 10^9$);
- `? v` — print the value of variable in vertex v ($1 \le v \le n$).

Output

Print answers to each query.

Examples

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

Problem F. Chip and Dale

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	256 megabytes

Chip and Dale are rushing to the rescue! But attentive viewers know that help is usually needed by Chip and Dale themselves, so today you will need to play the role of a smart Gadget. So, Chip and Dale are back in the clutches of Fat Cat. The cat doesn't like rodents, so he prepared a very hard task for them. He's going to put them in a labyrinth and see if they can get out of it. The labyrinth is a tree in which each edge is directed. The Gadget overheard Fat Cat talking to his accomplices and now she knows a few possible options: which point in the maze Fat Cat will put her friends, and where the exit will be. For each of these options, she wants to know whether Chip and Dale can find a way out or not.

Input

The first line of input contains one integer n ($n \le 10^5$) — the number of vertices in the tree. Next $n-1$ lines contain a description of tree edges. The $(i+1)$-th line contains two vertices a_i, b_i, describing a directed edge in a tree from vertex a_i to b_i.

The next line contains one integer m ($m \le 10^5$) — the number of queries. The next m lines contain the description of queries, the $(n+1+i)$-th of them contains two integers x_i and y_i.

Output

For each query, in a separate line, print "`Yes`", if there is a path between vertices x_i and y_i, and "`No`", otherwise.

Example

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

Problem G. Genealogy

Input file:	`standard input`
Output file:	`standard output`
Time limit:	4 seconds
Memory limit:	256 megabytes

During discussions in parliament, lords usually divide into groups with similar views on how to solve the problem. The outcome of the discussion depends on the decision of the most influential group of Lords. That is why the calculation of the influence of the group is the most important task.

Naturally, each lord values the antiquity of a family, therefore the influence of the lord is equal to the antiquity of his family. The antiquity of the lord's family is the number of ancestors of the lord: his father, his grandfather, his great-grandfather, etc. To calculate the influence of a group of lords, you need to calculate the number of lords in the group along with their ancestors. Note that if a lord is the ancestor of two or more lords in a group, then this lord should only be counted once.

You have been given the family tree of lords (surprisingly, all lords are descended from one great-lord) and a list of groups. For each group, find its influence.

Input

First line contains a single integer n — the number of lords ($1 \le n \le 100\,000$). Lords are numbered from 1 to n. Next line contains n integers $p_1, p_2, \ldots, p_n$, where lord number p_i is a parent of lord number i. If lord is a founder of a family, then p_i equals to -1. It's guaranteed that input forms a tree.

Third line contains a single integer g — the number of groups ($1 \le g \le 3\,000\,000$). Next g lines contain groups description. Line number j contains an integer k_j — the size of j-th group, followed by k_j different integers — numbers of lords in j-th group. It's guaranteed that sum of all k_j doesn't exceed $3\,000\,000$.

Output

Output g lines. Line number j should contain the influence of j-th group. It's guaranteed that size of output doesn't exceed six megabytes.

Examples

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

Problem H. Yet another sparse table

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	256 megabytes

You are given an array consisting of n integers, and an integer m. Write a program that will answer the queries: find the product of array elements between u-th and v-th modulo m.

Input

The first line contains three integers: n, m and a_1 ($1 \leqslant n \leqslant 10^5$; $1 \leqslant m \leqslant 10^7$; $0 \leqslant a_1 < 16\,714\,589$) — the number of integers in the given array, the number of queries, and the first element of the given array, respectively.

The second line contains two integers u_1, and v_1 ($1 \leqslant u_1, v_1 \leqslant n$) — the first query.

For the sake of the input size, the array and the queries should be generated.

The array elements $a_2, a_3, \ldots, a_n$ are generated with the following formula:

$$a_{i+1} = (23 \cdot a_i + 21563) \bmod 16714589.$$

For instance, if $n = 10$, $a_1 = 12345$ the following array should be generated: $a = (12345, 305498, 7048017, 11694653, 1565158, 2591019, 9471233, 570265, 13137658, 1325095)$.

The queries are generated in the following way:

$$u_{i+1} = \big((17 \cdot u_i + 751 + r_i + 2i) \bmod n\big) + 1,$$
$$v_{i+1} = \big((13 \cdot v_i + 593 + r_i + 5i) \bmod n\big) + 1,$$

where r_i — the answer to query i.

Be careful, u_i can be greater than v_i.

Output

Print three integers u_m, v_m and r_m (the last query, and the answer to it).

Example

standard input	standard output
10 5677 8 12345 3 9	5 3 1631

Problem I. Block Towers

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	512 megabytes

Today Mr. Fox decided to play with some blocks in 2D. Each block is a 1 inch by 1 inch square. There are n towers of blocks in a row, the i-th tower consists of h_i blocks. For example, if $n = 6$ and $h = 3, 1, 5, 4, 1, 6$, then blocks will be arranged as follows:

```
.....X
..X..X
..XX.X
X.XX.X
X.XX.X
XXXXXX
```

Mr. Fox wants to figure out an answer to q queries about his blocks (without actually changing them). The i-th query is: if we consider only towers from a_i to b_i inclusive, getting rid of all other towers, what is the maximum number of square inches of water will they be able to hold? The cell can hold water if it does not contain a block and there is a tower to the left and to the right of that cell, which has at least the same height as the cell. For example, if $a_i = 2$, $b_i = 6$, the picture will look like this (* denotes a cell with water):

```
....X
.X**X
.XX*X
.XX*X
.XX*X
XXXXX
```

Input

The first line of the input contains integers n and q. Then n integers h_i follow. After that, there are q queries given as two integers a_i and b_i.

$1 \le n, q \le 300\,000$,

$1 \le h_i \le 1\,000\,000\,000$,

$1 \le a_i \le b_i \le n$.

Output

Print the sum of answers to all queries modulo $10^9 + 7$.

Examples

standard input	standard output
11 11 2 4 5 3 2 6 1 3 1 8 1 1 1 1 2 1 3 1 4 1 5 1 6 1 7 1 8 1 9 1 10 1 11	60
5 3 10 1 1 1 10 1 5 1 2 4 5	27