Problem A. Ball momentum

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

There are n baseball balls and n machines for serving them. Each ball has its weight w_i and each machine has its power of serving p_i. We can choose a ball and machine to serve it. Let's say that *difficulty* of catching i-th ball served using machine j is $w_i \times p_j$. You want to pick a pair $(i; j)$ such that any other pair $(i'; j')$ ($i \neq i'$ and $j \neq j'$) has difficulty of catching less or equal to the difficulty of catching the pair$(i; j)$. Tell the number of such pairs.

Input

The first line of input contains a single integer n ($1 \leq n \leq 2 \cdot 10^5$) — number of balls and serving machines.

The second line of input contains n integers w_i ($1 \leq w_i \leq 10^9$) — weitghts of balls.

The third line of input contains n integers p_i ($1 \leq p_i \leq 10^9$) — powers of serving machines.

Output

Print one integer — answer to the problem.

Scoring

1. (7 points): $n \leq 3$;
2. (7 points): all w_i are equal and all p_i are equal;
3. (8 points): all w_i are equal;
4. (9 points): $w_i, p_i \leq 1000$;
5. (23 points): $n \leq 100$
6. (24 points): $n \leq 1000$
7. (22 points): no additional restrictions.

Examples

standard input	standard output
2 1 3 2 4	2
4 1 5 9 6 7 4 5 3	2

Problem B. Typical Query Problem

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

You are given two integers C and L and array a consisting of n integers. You are given q queries of format $l\ r\ x$. After each query, you have to:

1. subtract x from all elements on a segment $[l; r]$ of array a;
2. tell whether there exist a `good` segment S on the array a. A segment S is considered `good` if $len(S) \geq L$ and $avg(S) \leq C$, where $len(S)$ denotes the length of the segment S and $avg(S)$ denotes the average value on segment S.

Input

The first line of input contains four integers n, L ($1 \leq L \leq n \leq 2 \cdot 10^5$), q ($1 \leq q \leq 2 \cdot 10^5$), C ($-10^9 \leq C \leq 10^9$).

The second line of input contains n integers a_i ($-10^9 \leq a_i \leq 10^9$) — given array a.

The following q lines contain three integers l, r ($1 \leq l \leq r \leq n$), x ($0 \leq x \leq 10^7$) each — description of the queries.

Output

Print q lines, each consisting of `YES`, if there exists a `good` segment S, or `NO` otherwise.

Scoring

1. (11 points): $n \leq 3$;
2. (12 points): $n \leq 100$, $q \leq 1000$;
3. (14 points): $q = 1$, $C = 0$;
4. (15 points): $C = 0$;
5. (19 points): $q = 1$;
6. (29 points): no additional restrictions.

Example

standard input	standard output
4 2 3 2 5 6 3 4 1 2 2 1 3 1 1 4 2	NO NO YES

Problem C. Tree queries

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

You are given a connected graph with n vertexes and $n-1$ edges. In other words, you are given a tree. Vertex 1 is the root of the tree. In vertex v there is an integer a_v written on it. Let's define cost of the tree as `XOR` sum of all its values. You were given q queries, containing two integers v and x. Each query makes $a_u = a_u \oplus x$, where u belongs to the subtree of the vertex v and $\oplus$ denotes `XOR` operation. As we all know, this problem can be easily solved by our participants, but if all was that easy, this problem wouldn't appear at this Olympiad. You know that some queries were changed by angry Anton. Thus, you are asked to find the sum of costs modulo 998244353, of all 2^q trees that can be obtained by applying some (possibly none) queries on the tree.

Input

The first line of input contains two integers n ($1 \le n \le 2 \cdot 10^5$) and q ($1 \le q \le 2 \cdot 10^5$) — number of vertexes in the tree and number of queries accordingly.

The second line of input contains n integers a_i ($0 \le a_i < 2^{60}$) — values written on the vertex of the tree.

The following $n-1$ lines contain two integers u and v each — edges of the given tree.

The following q lines contain two integers v ($1 \le v \le n$) and x ($0 \le x < 2^{60}$) each — description of queries.

Output

Print one integer in the first line — answer to the problem.

Scoring

1. (7 points): $q \le 2$;
2. (12 points): $q \le 10$;
3. (16 points): $q \le 20$;
4. (18 points): $a_i, x < 2$;
5. (13 points): $a_i, x < 1024$;
6. (15 points): $a_i, x < 2^{18}$;
7. (19 points): no additional restrictions.

Example

standard input	standard output
5 3 1 0 2 3 1 1 2 1 3 3 4 3 5 3 2 4 3 1 5	28

Problem D. Anton the Guard

Input file:	`standard input`
Output file:	`standard output`
Time limit:	3 seconds
Memory limit:	256 megabytes

Anton is a security guard responsible for n objects numbered from 1 to n. There are also $n-1$ roads, where the i-th road connects objects u_i and v_i and has a length of w_i.

At the beginning, Anton is located at object 1.

Let's define the priority of each object as the minimum number of roads that need to be traveled from this object to object 1. For example, for object 1 this number will be 0; for all objects directly connected to 1, it will be 1, and so on.

Anton needs to visit all objects. He needs to first visit all objects that have a priority of 1, then those that have a priority of 2, and so on. If there are several objects with the same priority, Anton can choose the order in which to visit them. Note that he can only visit any object with priority k after he has visited all objects with priority $k-1$.

Find the minimum distance that he will have to travel.

Input

The first line contains an integer n ($1 \le n \le 10^6$).

The second line contains $n-1$ integers p_i which indicate that there is a road from object $i+1$ to object p_i ($1 \le p_i \le n$).

The third line contains $n-1$ integers w_i which are the lengths of roads between objects $i+1$ and p_i ($1 \le w_i \le 10^9$).

It is guaranteed that it is possible to reach all other objects from object 1.

Output

Output a single integer — the answer to the problem.

Scoring

1. (4 points): $1 \le n \le 10$;
2. (5 points): $1 \le n \le 22$;
3. (13 points): no more than 6 objects have the same priority;
4. (10 points): no more than 10 objects have the same priority;
5. (15 points): $w_i = 1$;
6. (15 points): $1 \le n \le 10^5$;
7. (38 points): without additional restrictions.

Examples

standard input	standard output
7 3 1 1 1 5 6 14 10 6 5 7 3	85
5 1 2 3 4 5 6 10 21	42
4 1 1 1 4 10 30	58