Codeforces Round #430 (Div. 2)

A. Kirill And The Game

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

Kirill plays a new computer game. He came to the potion store where he can buy any potion. Each potion is characterized by two integers — amount of experience and cost. The efficiency of a potion is the ratio of the amount of experience to the cost. Efficiency may be a non-integer number.

For each two integer numbers a and b such that $l \le a \le r$ and $x \le b \le y$ there is a potion with experience a and cost b in the store (that is, there are $(r - l + 1) \cdot (y - x + 1)$ potions).

Kirill wants to buy a potion which has efficiency k. Will he be able to do this?

Input

First string contains five integer numbers l, r, x, y, k ($1 \le l \le r \le 10^7$, $1 \le x \le y \le 10^7$, $1 \le k \le 10^7$).

Output

Print "YES" without quotes if a potion with efficiency exactly k can be bought in the store and "NO" without quotes otherwise.

You can output each of the letters in any register.

Examples

input
1 10 1 10 1
output
YES

input
1 5 6 10 1
output
NO

B. Gleb And Pizza

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

Gleb ordered pizza home. When the courier delivered the pizza, he was very upset, because several pieces of sausage lay on the crust, and he does not really like the crust.

The pizza is a circle of radius r and center at the origin. Pizza consists of the main part — circle of radius $r - d$ with center at the origin, and crust around the main part of the width d. Pieces of sausage are also circles. The radius of the i -th piece of the sausage is r_i, and the center is given as a pair (x_i, y_i).

Gleb asks you to help determine the number of pieces of sausage caught on the crust. A piece of sausage got on the crust, if it completely lies on the crust.

Input

First string contains two integer numbers r and d $(0 \le d < r \le 500)$ — the radius of pizza and the width of crust.

Next line contains one integer number n — the number of pieces of sausage $(1 \le n \le 10^5)$.

Each of next n lines contains three integer numbers x_i, y_i and r_i ($-500 \le x_i, y_i \le 500$, $0 \le r_i \le 500$), where x_i and y_i are coordinates of the center of i-th peace of sausage, r_i — radius of i-th peace of sausage.

Output

Output the number of pieces of sausage that lay on the crust.

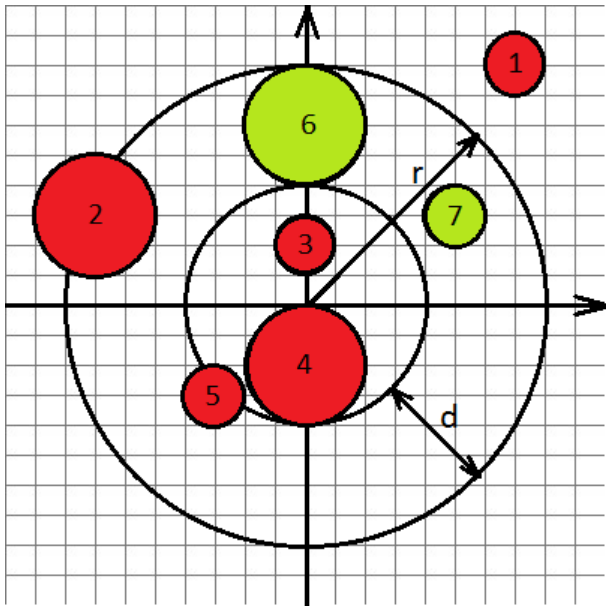
Examples

input
8 4 7 7 8 1 -7 3 2 0 2 1 0 -2 2 -3 -3 1 0 6 2 5 3 1
output
2

input
10 8 4 0 0 9 0 0 10 1 0 1 1 0 2
output
0

Note

Below is a picture explaining the first example. Circles of green color denote pieces of sausage lying on the crust.


1
6
r
2
7
3
4
5
d

C. Ilya And The Tree

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

Ilya is very fond of graphs, especially trees. During his last trip to the forest Ilya found a very interesting tree rooted at vertex 1. There is an integer number written on each vertex of the tree; the number written on vertex i is equal to a_i.

Ilya believes that the beauty of the vertex x is the greatest common divisor of all numbers written on the vertices on the path from the root to x, including this vertex itself. In addition, Ilya can change the number in one arbitrary vertex to 0 or leave all vertices unchanged. Now for each vertex Ilya wants to know the maximum possible beauty it can have.

For each vertex the answer must be considered independently.

The beauty of the root equals to number written on it.

Input

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

Next line contains n integer numbers a_i ($1 \le i \le n$, $1 \le a_i \le 2 \cdot 10^5$).

Each of next $n - 1$ lines contains two integer numbers x and y ($1 \le x, y \le n$, $x \ne y$), which means that there is an edge (x, y) in the tree.

Output

Output n numbers separated by spaces, where i-th number equals to maximum possible beauty of vertex i.

Examples

input
2 6 2 1 2
output
6 6

input
3 6 2 3 1 2 1 3
output
6 6 6

input
1 10
output
10

D. Vitya and Strange Lesson

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

Today at the lesson Vitya learned a very interesting function — *mex*. *Mex* of a sequence of numbers is the minimum non-negative number that is not present in the sequence as element. For example, $mex([4, 33, 0, 1, 1, 5]) = 2$ and $mex([1, 2, 3]) = 0$.

Vitya quickly understood all tasks of the teacher, but can you do the same?

You are given an array consisting of n non-negative integers, and m queries. Each query is characterized by one number x and consists of the following consecutive steps:

- Perform the bitwise addition operation modulo 2 (*xor*) of each array element with the number x.
- Find *mex* of the resulting array.

Note that after each query the array changes.

Input

First line contains two integer numbers n and m ($1 \le n, m \le 3 \cdot 10^5$) — number of elements in array and number of queries.

Next line contains n integer numbers a_i ($0 \le a_i \le 3 \cdot 10^5$) — elements of then array.

Each of next m lines contains query — one integer number x ($0 \le x \le 3 \cdot 10^5$).

Output

For each query print the answer on a separate line.

Examples

input
2 2 1 3 1 3
output
1 0

input
4 3 0 1 5 6 1 2 4
output
2 0 0

input
5 4 0 1 5 6 7 1 1 4 5
output
2 2 0 2

E. Nikita and game

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

Nikita plays a new computer game. There are m levels in this game. In the beginning of each level a new class appears in the game; this class is a child-class of the class y_i (and y_i is called parent-class for this new class). Thus, the classes form a tree. Initially there is only one class with index 1.

Changing the class to its neighbour (child-class or parent-class) in the tree costs 1 coin. You can not change the class back. The cost of changing the class a to the class b is equal to the total cost of class changes on the path from a to b in the class tree.

Suppose that at i -th level the maximum cost of changing one class to another is x. For each level output the number of classes such that for each of these classes there exists some other class y, and the distance from this class to y is exactly x.

Input

First line contains one integer number m — number of queries ($1 \le m \le 3 \cdot 10^5$).

Next m lines contain description of queries. i -th line ($1 \le i \le m$) describes the i -th level and contains an integer y_i — the index of the parent-class of class with index $i + 1$ ($1 \le y_i \le i$).

Output

Suppose that at i -th level the maximum cost of changing one class to another is x. For each level output the number of classes such that for each of these classes there exists some other class y, and the distance from this class to y is exactly x.

Examples

input
4 1 1 2 1
output
2 2 2 3

input
4 1 1 2 3
output
2 2 2 2