G. Yash And Trees

time limit per test: 4 seconds
memory limit per test: 512 megabytes
input: standard input
output: standard output

Yash loves playing with trees and gets especially excited when they have something to do with prime numbers. On his 20th birthday he was granted with a rooted tree of n nodes to answer queries on. Hearing of prime numbers on trees, Yash gets too intoxicated with excitement and asks you to help out and answer queries on trees for him. Tree is rooted at node 1. Each node i has some value a_i associated with it. Also, integer m is given.

There are queries of two types:

1. for given node v and integer value x, increase all a_i in the subtree of node v by value x
2. for given node v, find the number of prime numbers p less than m, for which there exists a node u in the subtree of v and a non-negative integer value k, such that $a_u = p + m \cdot k$.

Input

The first of the input contains two integers n and m ($1 \le n \le 100\,000, 1 \le m \le 1000$) — the number of nodes in the tree and value m from the problem statement, respectively.

The second line consists of n integers a_i ($0 \le a_i \le 10^9$) — initial values of the nodes.

Then follow n - 1 lines that describe the tree. Each of them contains two integers u_i and v_i ($1 \le u_i, v_i \le n$) — indices of nodes connected by the i-th edge.

Next line contains a single integer q ($1 \le q \le 100\,000$) — the number of queries to proceed.

Each of the last q lines is either `1 v x` or `2 v` ($1 \le v \le n, 0 \le x \le 10^9$), giving the query of the first or the second type, respectively. It's guaranteed that there will be at least one query of the second type.

Output

For each of the queries of the second type print the number of suitable prime numbers.

Examples

input

```
8 20
3 7 9 8 4 11 7 3
1 2
1 3
3 4
4 5
4 6
4 7
5 8
4
2 1
1 1 1
2 5
2 4
```

output

```
3
1
1
```

input

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

output

```
2
```