Problem G. Mici

Input file:	*standard input*
Output file:	*standard output*
Time limit:	2 seconds
Memory limit:	1024 mebibytes

The Romanian dish called "Mici" refers to a kind of thick sausage made out of minced meat and without any outer layer. Preparing Mici is too easy to be worth explaining, and knowing how to grill them properly is an art, so we will not even attempt to explain this to you. What you should know about Mici is that Romanian people are crazy about them, so they learned a trick to call the Mici to them. Also, each Romanian person, at a given point in time, has a sweet spot for how many Mici they want to eat.

At a traditional family barbecue with n of the closest relatives, the table is set, and each person has a plate with some Mici on it. You can imagine the table as an unrooted tree with n nodes where each person sits in one node.

The family members, true Romanians, are only concerned about one thing: how many Mici they are going to get. For that purpose, the person sitting in node u can use the magic trick that calls the Mici to them; we will call this operation "`CJ` u". When the trick is applied, the effect is that all the Mici on the table move one step closer to node u: a Mic that is in node v will move to the neighbor of v in the tree that has the smallest travel distance to u in the tree. The Mici in node u stay in place.

Every once in a while, a family member feels very strongly that they would like to eat exactly x Mici, so they wonder if it is true that they have exactly x Mici on their plate. This operation is denoted as "`JS` u x", meaning that the person sitting in node u asks if there are x Mici in node u or not.

Because the family is too hungry to think properly right now, they ask for your help in answering these questions. Note that the family members don't actually eat the Mici in this problem.

Input

The first line of the input contains two integers, n and q: the number of family members and the number of operations that happen on the dinner table, respectively ($2 \le n \le 10^5$, $1 \le q \le 10^5$).

The next line contains an array of n integers $m_1, m_2, \ldots, m_n$: the initial number of Mici in nodes $1, 2, \ldots, n$ ($0 \le m_i \le 10^9$). The total number of Mici on the table will be at most 10^9.

The next $n-1$ lines describe the layout of the table as a tree. Each of these lines will contain two integers, u and v, meaning that there is an edge between u and v ($1 \le u, v \le n$, $u \ne v$). The given edges form a tree.

Each of the next q lines describes one of the possible operations:

- "`CJ` u": the person in node u calls the Mici to them ($1 \le u \le n$).
- "`JS` u x": the person in node u asks if there are exactly x Mici on their plate ($1 \le u \le n$, $0 \le x \le 10^9$).

Output

For each query of type "`JS`", print a line with a single integer: `1` if the answer to the query is "yes" or `0` otherwise.

Example

standard input	*standard output*
6 4 1 1 1 2 2 2 1 2 2 3 2 4 4 5 3 6 CJ 6 CJ 6 JS 6 4 JS 6 3	1 0