C. Propagating tree

time limit per test: 2 seconds

memory limit per test: 256 megabytes

input: standard input

output: standard output

Iahub likes trees very much. Recently he discovered an interesting tree named propagating tree. The tree consists of n nodes numbered from 1 to n, each node i having an initial value a_i. The root of the tree is node 1.

This tree has a special property: when a value val is added to a value of node i, the value -val is added to values of all the children of node i. Note that when you add value -val to a child of node i, you also add -(-val) to all children of the child of node i and so on. Look an example explanation to understand better how it works.

This tree supports two types of queries:

- "$1\ x\ val$" — val is added to the value of node x;
- "$2\ x$" — print the current value of node x.

In order to help Iahub understand the tree better, you must answer m queries of the preceding type.

Input

The first line contains two integers n and m $(1 \le n, m \le 200000)$. The second line contains n integers a_1, a_2, ..., a_n $(1 \le a_i \le 1000)$. Each of the next $n–1$ lines contains two integers v_i and u_i $(1 \le v_i, u_i \le n)$, meaning that there is an edge between nodes v_i and u_i.

Each of the next m lines contains a query in the format described above. It is guaranteed that the following constraints hold for all queries: $1 \le x \le n$, $1 \le val \le 1000$.

Output

For each query of type two (print the value of node x) you must print the answer to the query on a separate line. The queries must be answered in the order given in the input.

Examples

input

```
5 5
1 2 1 1 2
1 2
1 3
2 4
2 5
1 2 3
1 1 2
2 1
2 2
2 4
```

output

```
3
3
0
```

Note

The values of the nodes are $[1, 2, 1, 1, 2]$ at the beginning.

Then value 3 is added to node 2. It propagates and value -3 is added to it's sons, node 4 and node 5. Then it cannot propagate any more. So the values of the nodes are $[1, 5, 1, -2, -1]$.

Then value 2 is added to node 1. It propagates and value -2 is added to it's sons, node 2 and node 3. From node 2 it propagates again, adding value 2 to it's sons, node 4 and node 5. Node 3 has no sons, so it cannot propagate from there. The values of the nodes are $[3, 3, -1, 0, 1]$.

You can see all the definitions about the tree at the following link: `http://en.wikipedia.org/wiki/Tree_(graph_theory)`