Gaming Trees

The Phantom Lancer is one OP character. He has the ability to create multiple illusions and each illusion has the ability to create more illusions. The gaming tree of the character is a tree with the Phantom Lancer at the root. His illusions form the other nodes of the tree. Each node of the tree has a strength a_i. For simplicity, let us denote the Phantom Lancer as node 1, i.e., the gaming tree is rooted at 1.

This tree has a special property: when v is added to the strength of a node i, $-v$ is added to the strength of all the children of node i. When $-v$ is added to the strength of a child of node i, $-(-v) = v$ is added to the strength of all the children of the child of node i and so on.

The gaming tree supports two types of queries:

1. "1 x v" - v is added to the strength of node x.
2. "2 x" - The current strength of node x is printed.

m queries need to be answered to truly understand your character/illusions, which is required in order to win the game.

Input

The first line contains two integers n and m, the number of nodes in the tree and the number of queries.

The second line contains n integers a_1, a_2, ..., a_n, the initial strengths of the nodes.

Each of the next n–1 lines contain two integers u_i and v_i, meaning that there is an edge between the nodes u_i and v_i.

Each of the next m lines contains a query in the format described above.

Output

For each query of type 2, print the current strength of node x on a separate line.

Constraints

$1 \leq n, m \leq 200000$
$1 \leq a_i \leq 1000$
$1 \leq u_i, v_i \leq n$
For all queries, $1 \leq x \leq n$ and $1 \leq v \leq 1000$

Sample Input 1

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

Sample Output 1

```
4
1
2
```

Sample Explanation 1

The strengths of the nodes are [1, 2, 1, 1] initially.

Then 2 is added to the strength of node 2. This propagates and 2 is subtracted from the strength of its child, node 4. So the strengths of the nodes are now [1, 4, 1, −1].

Then 3 is added to the strength of node 1. This propagates and −3 is added to the strength of its children, nodes 2 and 3. From node 2, it propagates again, adding 3 to the strength of its child, node 4. Node 3 has no children and so cannot propagate. The strength of the nodes are [4, 1, −2 , 2].