

E. Little Girl and Problem on Trees

time limit per test: 2 seconds

memory limit per test: 256 megabytes

input: standard input

output: standard output

A little girl loves problems on trees very much. Here's one of them.

A tree is an undirected connected graph, not containing cycles. The degree of node x in the tree is the number of nodes y of the tree, such that each of them is connected with node x by some edge of the tree.

Let's consider a tree that consists of n nodes. We'll consider the tree's nodes indexed from 1 to n . The considered tree has the following property: each node except for node number 1 has the degree of at most 2.

Initially, each node of the tree contains number 0. Your task is to quickly process the requests of two types:

- Request of form: $0 \ v \ x \ d$. In reply to the request you should add x to all numbers that are written in the nodes that are located at the distance of at most d from node v . The distance between two nodes is the number of edges on the shortest path between them.
- Request of form: $1 \ v$. In reply to the request you should print the current number that is written in node v .

Input

The first line contains integers n ($2 \leq n \leq 10^5$) and q ($1 \leq q \leq 10^5$) — the number of tree nodes and the number of requests, correspondingly.

Each of the next $n - 1$ lines contains two integers u_i and v_i ($1 \leq u_i, v_i \leq n, u_i \neq v_i$), that show that there is an edge between nodes u_i and v_i . Each edge's description occurs in the input exactly once. It is guaranteed that the given graph is a tree that has the property that is described in the statement.

Next q lines describe the requests.

- The request to add has the following format: $0 \ v \ x \ d$ ($1 \leq v \leq n, 1 \leq x \leq 10^4, 1 \leq d < n$).
- The request to print the node value has the following format: $1 \ v$ ($1 \leq v \leq n$).

The numbers in the lines are separated by single spaces.

Output

For each request to print the node value print an integer — the reply to the request.

Examples

input
3 6 1 2 1 3 0 3 1 2 0 2 3 1 0 1 5 2 1 1 1 2 1 3
output
9 9 6

input

```
6 11
1 2
2 5
5 4
1 6
1 3
0 3 1 3
0 3 4 5
0 2 1 4
0 1 5 5
0 4 6 2
1 1
1 2
1 3
1 4
1 5
1 6
```

output

```
11
17
11
16
17
11
```