After seeing that the bussiness with printing presses didn't bring much profit, Ivan decided to become a thief. The city in which Ivan lives consists of houses connected with streets. On every street there is some number of policemen and starting from one house, one can't return to the same house without passing through a street two or more times. Ivan can sneak through some number of policemen who stay on patrol on a given street without getting caught.

Formally said, the city is an unrooted tree with N nodes, where the nodes are houses and the edges are streets. Every house is characterized by the total price of all items in it, which Ivan can steal, and every street is characterized with the number of policemen on it. Ivan wants to start from his house and steal as much money as possible, paasing only through streets with less policemen than he can outsmart. Help him by finding the maximum amount of money that he can steal for each of the queries given to you. Other than that, because Ivan isn't sure of the inventories of the houses, he will ask you to change the values of the total price of items in some houses.

**Input**

On the first row of the file thief.in there are two natural numbers **N** and **Q**. On the second row there are **N** numbers – the total value of items that Ivan can steal from every house. The next N-1 rows consist of three numbers **A, B, C**. Every such row defines a two-way street from **A** to **B**, on which there are **C** policemen. On the last **Q** rows will be the queries, for which Ivan wants answers. He describes them by 3 numbers **t X L**:

1. If **t**=1, then he wants to ask how much money he can steal if his home is vertex X and can pass unnoticed through streets with no more than L policemen.

2) If **t**=2, then he tells you that from now on vertex **X** has **L** money on it, i.e. **аx** = **L**.

**Output**

For every query of type 1) in the file thief.out output the maximal amount of money that Ivan can steal without getting noticed.

**Limits**

The given graph is a tree.

for every edge

For every query:

**Time limit: 2sec.**

**Memory limit: 256 MB.**

**Example test**

|  |  |
| --- | --- |
| **Input (thief.in)** | **Output (thief.out)** |
| 7 5  1 3 2 3 10 1 1  1 2 4  2 3 7  1 4 6  2 5 10  5 6 3  1 7 15  1 1 7  2 2 100  1 1 7  1 2 10  1 7 14 | 9  106  117  1 |

**Explanation**

The queries are as follows:

1) Ivan can rob the houses with numbers 1, 2, 3, 4; notice that for simplicity we say that Ivan can also rob his own house. He gets 1+3+2+3=9 money.

2) In house 2 there is now 100 money.

3) Ivan can rob houses with numbers 1, 2, 3, 4. He gets 1+100+2+3=106 money.

4) Ivan can rob houses with numbers 1, 2, 3, 4, 5, 6. He gets 1+100+2+3+10+1=117 money.

5) Ivan can rob only the house with number 7. He receives 1 money.