Submit:	railway.c / railway.cpp / railway.pas	Time limit:	on the order of seconds
Input:	railway.in	Memory limit:	32 MB
Output:	railway.out	Points:	100

Several years ago, the government of the well-known country Byteland decided to build a railway network to make public transport in the country a little bit easier. However, due to insufficient funds, the minister of the transport wanted the "network" to be as simple as possible. Hence, the following requirements needed to be met:

- Every train connects directly two cities, there are no stops in between.
- Every train operates in both directions (i.e., the train going from A to B goes from B to A as well) and the prices for the two directions are the same.
- It is possible to go from every city to every other city (possibly with some changes).
- The number of trains is the minimum possible.

Now, the railway network has been finished and the first customers can enjoy the speed and the reliability of the Byteland railway. The only problem is the issuing of tickets: When the customer travels from A to B using 42 trains, the assistant has to sum up the prices of the trains, which is really anoying. What is more, the assistant can make a mistake. To solve the problem, the director of the Byteland Railway Corporation hired you. Your task is to write a program which calculates the price of the tickets.

First, the program reads the description of the railway and the prices of the trains. Then, the program is given several queries to answer: each query is a pair of cities x_i and y_i and the answer is the price of the ticket from x_i to y_i.

Input: The first line of the input file `railway.in` contains the integers N and Q (in this order) separated by a single space. N is the number of cities, $1 \le N \le 100\,000$, and Q is the number of queries, $1 \le Q \le 2\,000\,000\,000$. For simplicity, the cities are numbered by the integers $1, \ldots, N$. The following $N-1$ lines describe the trains: each line contains three space-separated integers a_i, b_i and c_i ($1 \le a_i, b_i \le N$, $0 \le c_i \le 2\,000\,000\,000$) meaning that there is a train from the city a_i to the city b_i and its cost is c_i. You may assume that the railway is correct, i.e., it really forms a tree. The following Q lines describe the queries: each line contains two integers x_i and y_i ($1 \le x_i, y_i \le N$, $x_i \ne y_i$) – the cities for which we want to calculate the price of the ticket.

Output: The output file `railway.out` should contain exactly Q lines. The i-th line should contain a single integer – the price of the ticket from x_i to y_i. You may assume that the price of a ticket from any city to any other city will fit into 32-bit signed integer, i.e., it will fit into `int` in C/C++ and into `longint` in Pascal.

Example:

`railway.in`
```
4 3
1 2 5
2 3 6
2 4 1
1 2
2 3
1 3
```

`railway.out`
```
5
6
11
```