Assignment 3

Problem A

Problem Description

Given a rooted binary tree from 1 to n, the root node of the tree is node 1. You are asked to print the parent node of every node in the binary tree.

Input

The first line contains one integer $n\ (1 \leq n \leq 500000)$, indicating the number of tree nodes.

Then $n-1$ lines follow. Each line contains two integers $u, v\ (1 \leq u, v \leq n)$ are the indices of nodes that form an edge. u is the parent of v.

Output

An array containing n integers, the i^{th} element of which represents the parent node of nodes from 1 to n in order in the binary tree. Note that for node 1, which is the root node of the tree, the parent should output -1

Sample input

```
6
1 2
1 3
3 4
3 5
2 6
```

Sample output

```
-1 1 1 3 3 2
```

Problem B

Problem Description

Given a rooted tree numbered from 1 to n, each edge has a weight w. The root node of the tree is node 1. You are asked to calculate the number of paths that start from the root, terminate in a leaf node, and satisfy the sum of edge weights in the path equals to num.

Input

The first line contains two integers n and num $(1 \leq n \leq 500000, 1 \leq num \leq 2000000000)$, indicating the number of tree nodes and the target number.

Then $n-1$ lines follow. Each line contains three integers u, v, w $(1 \leq u, v \leq n, 1 \leq w \leq 100)$ describing an edge. The first two integers are the indices of nodes that form an edge and the last integer indicates the weight of the edge.

Output

Output an integer which means how many paths satisfying the sum of edge weights in the path equals to num.

Sample input

```
6 6
1 2 2
1 3 3
3 4 4
3 5 3
2 6 4
```

Sample output

```
2
```

Requirement as homework

1. You need to accomplish the `TODO` in `Parents.java`, `Paths.java` . Please **don't modify** the function name or return value. But you need to define the parameters that are passed to the function.
2. Please **explain your algorithm** and **analyze its time complexity correctly** in a file, which must be `.pdf` file. (It's OK to write in Chinese.)
   If the complexity analysis does not correspond to the algorithm you wrote, you will lose some points.
3. In this assignment, two sets of data are provided for each problem, but we will use other data to verify the correctness of your code.
4. Please submit `Parents.java`, `Paths.java`, explanation file as a `.zip` file.