Maximal Path

All submissions
Best submissions

✔ **Points:** 100 (partial)
Time limit: 0.1s
Java 9: 0.5s
Kotlin: 0.5s
Memory limit: 32M
Java 9: 32M
Kotlin: 32M
Author:
doncho

Tags
Graphs
Difficulty
Easy

We are given a tree of **N** nodes, each containing a distinct integer number (between 1 and 2147483640, inclusive) and optionally a set of descendent nodes. Write a program that finds a path from some leaf of the tree to another (different) leaf of the tree with maximal sum of its nodes and prints this sum.

Input

- Read from the standard input
- The first input line contains **N** - the number of nodes in the tree.
- At the next **N-1** lines there are pairs of numbers in format (p_1 <- p_2) each meaning that node $\mathbf{p_1}$ is parent of the node $\mathbf{p_2}$. See the example bellow.

The input data will always be valid and in the described format. There is no need to check it explicitly.

Output

- Print on the standard output
- At the only output line you should print the maximal sum of nodes found

Constraints

- **N** will be between 2 and 3000, inclusive.

Sample tests

Input

```
10
(5 <- 11)
(1 <- 8)
(11 <- 3)
(8 <- 7)
(1 <- 5)
(11 <- 2)
(8 <- 6)
(2 <- 15)
(8 <- 4)
```

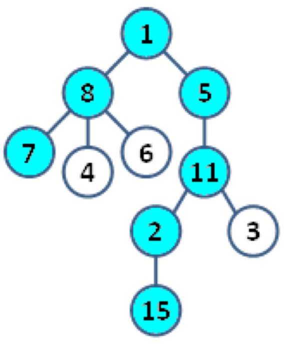
Copy

Output

```
49
```

Explanation

```
The maximal path is:
7 -> 8 -> 1 -> 5 -> 11 -> 2 -> 15
which is same as:
15 -> 2 -> 11 -> 5 -> 1 -> 8 -> 7
```



Comments

There are no comments at the moment.