## Problem 1 – Maximal Path

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

The input data should be read from the console.

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 (p1 <- p2) each meaning that node **p1** is parent of the node **p2**. 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

The output data should be printed on the console.

At the only output line you should print the maximal sum of nodes found.

### Constraints

* **N** will be between 2 and 3000, inclusive.
* Allowed working time for your program: 0.80 seconds.
* Allowed memory: 16 MB.

### Example

|  |  |  |
| --- | --- | --- |
| Input example | Output example | Explanation |
| 10  (5 <- 11)  (1 <- 8)  (11 <- 3)  (8 <- 7)  (1 <- 5)  (11 <- 2)  (8 <- 6)  (2 <- 15)  (8 <- 4) | 49 | The maximal path is:  7 -> 8 -> 1 -> 5 -> 11 -> 2 -> 15  which is same as:  15 -> 2 -> 11 -> 5 -> 1 -> 8 -> 7    7+8+1+5+11+2+15=49 |