Digital Pakistan Speed Programing Competition Online Qualifier Round

Instructions

- Do not open the booklet unless you are explicitly told to do so. You can only read these instructions below.
- If you have any question regarding the problems, seek a clarification from the judges using DOMJudge.
- Before submitting a run, make sure that it is executable via command line. For Java, it must be executable via "javac" and for GNU C++ via "g++". Java programmers need to remove any "package" statements and source code's file name must be the same as of main class. C++ programmers need to remove any getch() / system("pause") like statements.
- Do not attach input files while submitting a run, only submit/attach source code files, i.e., *.java or *.cpp or *.py.
- Language supported: C/C++, Java and Python3
- Source code file name should not contain white space or special characters.
- You must take input from Console i.e.: Standard Input Stream (stdin in C, cin in C++, System.in in Java, stdin in Python)
- You must print your output to Console i.e.: Standard Output Stream (stdout in C, cout in C++, System.out in Java)
- Please, don't create/open any file for input or output.
- Please strictly meet the output format requirements as described in problem statements, because your program will be auto judged by computer. Your output will be compared with judge's output byte-by-byte and not tolerate even a difference of single byte. So, be aware! **Pay special attention to spaces, commas, dots, newlines, decimal places, case sensitivity etc.**
- All your programs must meet the time constraint specified.
- The decision of judges will be absolutely final.

Problem 09: Raju and Rani

Time limit: 1 seconds

Raju and Rani decide to play a game in their Data Structures class. They have a **tree** available for the play. First, Raju chooses a subset of nodes such that there are **no two adjacent nodes** in the subset. Then, Rani chooses another subset of nodes (nodes in Raju's subset cannot be chosen by Rani) and destroys them. Rani always makes her choice such that after destroying the nodes in her subset, there are no two nodes in Raju's subset that are still connected. In addition, Rani always makes sure her subset contains as few nodes as possible. You should help Raju choose his subset such that the number of nodes Rani chooses is maximized.

Input Format

The first line contains a single integer ***n***, representing the number of nodes of the tree.
Each of the next ***n*-1** lines contains two integer values, representing two nodes that share an edge.

Output Format

The output should contain a single integer representing the maximum number of nodes Rani can be forced to choose by Raju.

Constraints

- $1 \le n \le 100000$
- Rani always plays optimally.

Sample

Input	Output

5
1 2
1 3
3 4
3 5
	1

3
1 2
2 3
	1

- In Sample 1: Raju chooses nodes 2 and 4 (for instance), and Rani must destroy either 1 or 3 to disconnect them.
- In Sample 2: Raju selects from node 1 and 3; Rani must destroy 1 node, which is node 2.