Problem 1 - ADA Adventurer (Programming) (11 points)

Problem Description

ADA Road, a prosperous country, was established by the ancestors' wisdom long time ago. This country consists of N cities indexed from 1 to N and $(N-1)$ roads. Each road connects two different cities, and all cities are connected. More formally, for any two different cities, a path exists from one to another along the given roads, and the distance between two cities is defined as the number of roads on this path.

To become a valiant ADA adventurer, you decided to visit this country. After arriving at the ADA Road, you discovered that the distance of some city pairs is too large, so villages need to pay lots of effort to reach one from another. To help the villages and get the experience points, you would like to help them resolve this problem. The only stuff allowed is to rebuild at most one road. That is, you can only remove at most one road from this country, and find a new pair of cities to build a road between them. The target is to minimize the farthest distance after rebuilding the road. Notice that you should avoid making some pairs of cities disconnected. Can you find the best way to complete it to win villages' reliance?

Input

The first line only contains one integer N, describing the number of cities in ADA Road.

In the next $N-1$ lines, each line contains two different integers a, b $(1 \leq a, b \leq N)$, describing a road connecting city a and b.

It is guaranteed that there exists a path from one city to another one for any pair of cities.

Test Group 0 (0 %)

- Sample Input.

Test Group 1 (20 %)

- $2 \leq N \leq 50$.

Test Group 2 (30 %)

- $2 \leq N \leq 1000$.

Test Group 3 (50 %)

- $2 \leq N \leq 2 \times 10^5$.

Output

Print one integer in a line, indicating the farthest distance after rebuilding the road.

Sample Input 1

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

Sample Output 1

```
3
```

Sample Input 2

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

Sample Output 2

```
3
```

Sample Input 3

```
10
8 3
2 8
7 9
6 5
4 6
9 1
3 10
3 5
2 9
```

Sample Output 3

```
4
```