Problem C
Conveyor Systems Engineer
Problem ID: carousels
Time limit: 1 second

A *baggage carousel* is the device at the airport that delivers checked luggage to the passengers at the baggage-reclaim area. Luggage is automatically moved to-and-from the baggage carousels using a *baggage conveyor network*.

The network can be described as follows. There are N baggage carousel's in the airport, numbered 1 to N, and conveyor belts are used to move luggage between them. Each conveyor belt connects two carousels, and can move luggage in both directions. Not all pairs of carousel's are connected by a single conveyor belt, but every pair of carousels is *connected*. That is, it is possible to move luggage between any pair of carousels by moving luggage through a series of one-or-more conveyor belts and carousels.

Zed is the Conveyor Systems Engineer at the airport, and designed the baggage conveyor network. Because Zed values simplicity and elegance above everything else, his network has the minimum number of conveyor belts such that every pair of carousels is connected.

However, Zed's manager has given him a new requirement: the network needs to be more robust. Sometimes conveyor belts fail, and the network needs to be able to handle a single failure of any conveyor belt. That is, if any single conveyor belt fails, every pair of carousels must remain connected.

So, Zed needs to add some extra conveyor belts. He wants to keep his network as simple as possible, so he wants to know: what is the minimum number of extra conveyor belts needed to meet the new requirement?

Input

The first line contains the integer N $(1 \leq N \leq 5 \times 10^5)$: the number of baggage carousels.

The next $N-1$ lines describe the conveyor network. The i^{th} of which contains two space-seperated integers: a_i and b_i, indicating that there is a conveyor belt between carousels a_i and b_i $(1 \leq a_i, b_i \leq N)$.

Output

Display the minimum number of conveyor belts that Zed needs to add to his network.

Sample Input 1	Sample Output 1

3
1 2
2 3
	1

Sample Input 2

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

Sample Output 2

```
2
```