Garden Party

Spring is in the air: a season much loved by the citizens of Silhouettown. Every year the question is asked, "Where is it best to enjoy the sunshine?", and every year the shady Shadow Garden is overrun by joggers and picnickers.

This year will be hotter than most, and the garden is expected to be heaving. The most popular attraction in the garden is the trail of inky outlines. Many visitors have argued over the meanings of each of the shapes, but all can agree on their beauty.

In the town hall, a map of the garden can be bought, showing the **n** outlines (numbered from 1 to **n**). Every outline is connected to at least one other outline by a set of **n**-1 paths, each of which directly connects two outlines.

Due to the expected crowds, the mayor wishes to build some more paths in the garden. It is desired that the visitors are able walk around the garden so they visit each outline exactly once and finish at the outline they started at.

Ever concerned about the city budget, the mayor wishes to construct the fewest number of paths possible to achieve this.

INPUT   You will be given an integer **n**, denoting the number of outlines in the garden. The next **n**-1 lines each contain 2 integers, **a** and **b**, indicating that outline **a** is directly connected to outline **b** by a path.
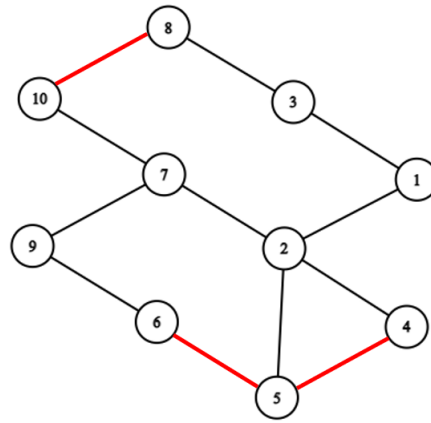
$1 \leq$ **n** $\leq 1,000,000$

OUTPUT   Output a single integer, the minimum number of paths that need to be constructed so that visitors can walk in a loop around the garden, visiting each node exactly once.

SAMPLE   For example, consider the garden shown below. Three paths have been added in red, to allow visitors to visit each node exactly once in a loop. One such path is 1 - 2 - 4 - 5 - 6 - 9 - 7 - 10 - 8 - 3 - 1. This is the minimum number of paths that need to be added to achieve this.
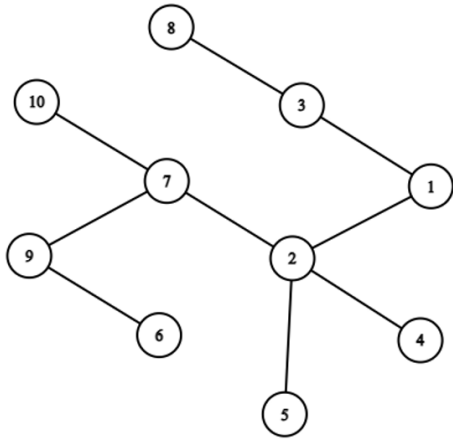
INPUT

6
1 2
1 3
1 4
1 5
1 6

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

OUTPUT

4

3