

# Chords

Input file:            **standard input**  
Output file:           **standard output**  
Time limit:            3 seconds  
Memory limit:         1024 megabytes

On the circle,  $2n$  points are marked, arranged at equal intervals in a clockwise direction. These points have been **randomly** (more on this in the **Input** section) connected in  $n$  pairs to form  $n$  chords. Your task is to find the largest possible set of chords in which no two intersect, and output the size of this set.

## Input

The first line of the standard input contains a single integer  $n$  ( $1 \leq n \leq 100\,000$ ), indicating the number of chords.

In the next  $n$  lines, there are two integers  $a_i$  and  $b_i$  ( $1 \leq a_i < b_i \leq 2n$ ) each, indicating that the chord connects the  $a_i$ -th and  $b_i$ -th points. All  $2n$  values of  $a_i$  and  $b_i$  are pairwise distinct.

**NOTE:** ~~Out of laziness of the jury~~ To make the task more interesting, all tests for this task (except for the sample test) were randomly generated. For each test, a value of  $n$  and a seed for the random number generator were chosen. Then a random permutation of numbers from 1 to  $2n$  was generated, which was divided into  $n$  pairs. Then, in each pair, the numbers could be swapped to ensure the condition  $a_i < b_i$ .

The sample test was created manually, but in order for the solution to be accepted, it must also be solved correctly.

## Output

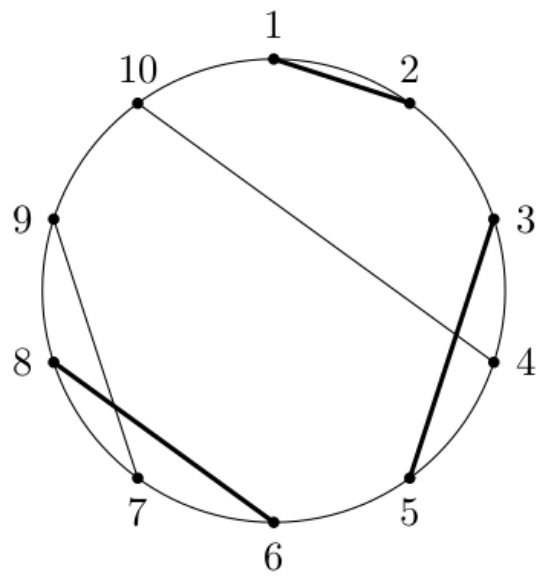
The output should contain a single integer, indicating the size of the largest set of non-intersecting chords.

## Example

standard input	standard output
5	3
1 2	
4 10	
7 9	
3 5	
6 8	

## Note

The chords in the sample test look as follows:



One of the possible largest correct sets of chords contains the chords connecting points 1 and 2, 3 and 5, and 6 and 8.