Double Sorting

Here we describe a typical problem. There are n balls and n boxes. Each ball is labeled by a unique number from 1 to n. Initially each box contains one of these balls. We can swap two balls in adjacent boxes. We are to sort these balls in increasing order by swaps, i.e. move the ball labeled by 1 to the first box, labeled by 2 to the second box, and so forth. The question is how many swaps are needed.

Now let us consider the situation where the balls are doubled, that is, there are $2n$ balls and n boxes, exactly two balls are labeled by k for each $1 \leq k \leq n$, and the boxes contain two balls each. We can swap two balls in adjacent boxes, one ball from each box. We are to move the both balls labeled by 1 to the first box, labeled by 2 to the second box, and so forth. The question is again how many swaps are needed.

Here is one interesting fact. We need 10 swaps to sort [5; 4; 3; 2; 1] (the state with 5 in the first box, 4 in the second box, and so forth): swapping 5 and 4, then 5 and 3, 5 and 2, 5 and 1, 4 and 3, 4 and 2, 4 and 1, 3 and 2, 3 and 1,and finally 2 and 1. Then how many swaps we need to sort [5, 5; 4, 4; 3, 3; 2, 2; 1, 1] (the state with two 5's in the first box, two 4's in the second box, and so forth)? Some of you might think 20 swaps — this is not true, but the actual number is 15.

Write a program that calculates the number of swaps for the two-ball version and verify the above fact.

Input

The input has the following format:

n
$ball_{1,1}$ $ball_{1,2}$
$ball_{2,1}$ $ball_{2,2}$
...
$ball_{n,1}$ $ball_{n,2}$

n is the number of boxes ($1 \leq n \leq 8$). $ball_{i,1}$ and balli,2 , for $1 \leq i \leq n$, are the labels of two balls initially contained by the i-th box.

Output

Print the minumum possible number of swaps.

Example

Input:
5
5 5
4 4
3 3
2 2
1 1

Output:
15