**Uncle Grandpa and inversion**

Uncle Grandpa has an array. He wants to calculate the minimum number of swaps between adjacent elements to sort the array in non-decreasing order.

## Input

The first line contains a single integer – the length of the array

The second line contains integers – the element of the array a

## Output

One integer is the answer to the problem

## Examples

|  |  |
| --- | --- |
| Input (inversion1.in) | Output (inversion1.out) |
| 4  5 3 4 1 | 5 |

## Explanation:

One of the possible way to sort the array in 5 operation is as follows:

* Swap the 3rd element and 4th element
* Swap the 2nd element and 3rd element
* Swap the 1st element and 2nd element
* Swap the 2nd element and 3rd element
* Swap the 3rd element and 4th element

## Note:

1. A skeleton file has been given to help you. You should not create a new file or rename the file provided. You should develop your program using this skeleton file.
2. You are free to define your own helper methods and classes (or remove existing ones) if it is suitable but you must put all the new classes, if any, in the same skeleton file provided

## Skeleton File

You are given the skeleton file Inversion.java. You should see the following contents when you open the file:

|  |
| --- |
| /\*\*  \* Name :  \* Matric. No :  \*/  import java.util.\*;  public class Inversion {  private void run() {  }  public static void main(String args[]) {  Inversion runner = new Inversion();  runner.run();  }  } |