

ORDENAÇÃO INSERTION

OrdenaInsertion.[c | cpp | java]

Faça um programa que receba um vetor de inteiros de tamanho N e ordene os seus elementos usando o algoritmo de inserção (Insertion Sort).

Entrada

O programa terá apenas um caso de teste.

A primeira linha do caso de teste contém um inteiro N , $0 < N \leq 10000$, representando o tamanho do vetor.

A segunda linha do caso de teste contém N números inteiros separados por um espaço em branco cada. É garantido que não há repetição dos valores do vetor.

Saída

O programa gerará várias linhas de saída.

A primeira linha, trata-se do vetor na sua ordem original, com os valores separados por espaço em branco.

A cada troca que o algoritmo realizar, você deve imprimir todo o vetor, com seus valores separados por espaço em branco.

A antepenúltima linha trata-se do vetor após passar pelo algoritmo de ordenação, ou seja, ordenado com seus valores separados por espaço em branco.

A penúltima linha trata-se de um inteiro representando a quantidade de vezes que o menor valor foi alterado.

A última linha deverá conter uma das seguintes frases: “PIOR CASO”, “MELHOR CASO” ou “CASO ALEATORIO” de acordo com o caso que foi tratado pelo algoritmo.

Vale lembrar que antes do primeiro elemento do vetor não há nada impresso, assim como não há nada impresso depois do último elemento do vetor apenas uma quebra de linha.

Após a impressão da última linha da saída, salte uma linha.

Exemplos

Entrada	Saída
10	54 12 2 90 4 78 55 21 1 48
54 12 2 90 4 78 55 21 1 48	12 54 2 90 4 78 55 21 1 48
	12 2 54 90 4 78 55 21 1 48
	2 12 54 90 4 78 55 21 1 48
	2 12 54 4 90 78 55 21 1 48
	2 12 4 54 90 78 55 21 1 48
	2 4 12 54 90 78 55 21 1 48
	2 4 12 54 78 90 55 21 1 48
	2 4 12 54 78 55 90 21 1 48
	2 4 12 54 55 78 90 21 1 48
	2 4 12 54 55 78 21 90 1 48
	2 4 12 54 55 21 78 90 1 48
	2 4 12 54 21 55 78 90 1 48
	2 4 12 21 54 55 78 90 1 48
	2 4 12 21 54 55 78 1 90 48
	2 4 12 21 54 55 1 78 90 48
	2 4 12 21 54 1 55 78 90 48
	2 4 12 21 1 54 55 78 90 48
	2 4 12 1 21 54 55 78 90 48
	2 4 1 12 21 54 55 78 90 48
	2 1 4 12 21 54 55 78 90 48
	1 2 4 12 21 54 55 78 90 48
	1 2 4 12 21 54 55 78 48 90
	1 2 4 12 21 54 55 48 78 90
	1 2 4 12 21 54 48 55 78 90
	1 2 4 12 21 48 54 55 78 90
	1 2 4 12 21 48 54 55 78 90
	Trocas:25
	CASO ALEATORIO

Entrada	Saída
10	1 2 4 12 21 48 54 55 78 90
1 2 4 12 21 48 54 55 78 90	1 2 4 12 21 48 54 55 78 90
	Trocas:0
	MELHOR CASO

Entrada	Saída
10	90 78 55 54 48 21 12 4 2 1
90 78 55 54 48 21 12 4 2 1	78 90 55 54 48 21 12 4 2 1
	78 55 90 54 48 21 12 4 2 1
	55 78 90 54 48 21 12 4 2 1
	55 78 54 90 48 21 12 4 2 1
	55 54 78 90 48 21 12 4 2 1
	54 55 78 90 48 21 12 4 2 1
	54 55 78 48 90 21 12 4 2 1
	54 55 48 78 90 21 12 4 2 1
	54 48 55 78 90 21 12 4 2 1
	48 54 55 78 90 21 12 4 2 1
	48 54 55 78 21 90 12 4 2 1
	48 54 55 21 78 90 12 4 2 1
	48 54 21 55 78 90 12 4 2 1
	48 21 54 55 78 90 12 4 2 1
	21 48 54 55 78 90 12 4 2 1
	21 48 54 55 78 12 90 4 2 1
	21 48 54 55 12 78 90 4 2 1
	21 48 54 12 55 78 90 4 2 1
	21 48 12 54 55 78 90 4 2 1
	21 12 48 54 55 78 90 4 2 1
	12 21 48 54 55 78 90 4 2 1
	12 21 48 54 55 78 4 90 2 1
	12 21 48 54 55 4 78 90 2 1
	12 21 48 54 4 55 78 90 2 1
	12 21 48 4 54 55 78 90 2 1
	12 21 4 48 54 55 78 90 2 1
	12 4 21 48 54 55 78 90 2 1
	4 12 21 48 54 55 78 90 2 1

	4 12 21 48 54 55 78 2 90 1
	4 12 21 48 54 55 2 78 90 1
	4 12 21 48 54 2 55 78 90 1
	4 12 21 48 2 54 55 78 90 1
	4 12 21 2 48 54 55 78 90 1
	4 12 2 21 48 54 55 78 90 1
	4 2 12 21 48 54 55 78 90 1
	2 4 12 21 48 54 55 78 90 1
	2 4 12 21 48 54 55 78 1 90
	2 4 12 21 48 54 55 1 78 90
	2 4 12 21 48 54 1 55 78 90
	2 4 12 21 48 1 54 55 78 90
	2 4 12 21 1 48 54 55 78 90
	2 4 12 1 21 48 54 55 78 90
	2 4 1 12 21 48 54 55 78 90
	2 1 4 12 21 48 54 55 78 90
	1 2 4 12 21 48 54 55 78 90
	1 2 4 12 21 48 54 55 78 90

Trocas:45

PIOR CASO