



UNIVERSIDADE FEDERAL DO ABC
CENTRO DE MATEMÁTICA, COMPUTAÇÃO E COGNIÇÃO

Prof. Monael Pinheiro Ribeiro

Prof. Paulo Henrique Pisani

INSERTION SORT

InsertionSort.[c | cpp | java | cs]

No exercício Inserção apenas um elemento foi inserido na lista ordenada. Usando a mesma abordagem repetidas vezes, você seria capaz de ordenar uma lista inteira que estivesse inicialmente desordenada.

Você pode imaginar que a lista ordenada inicialmente possui apenas um elemento, o que obviamente já a torna uma lista ordenada; depois disso, evolua a lista ordenada, elemento por elemento inserindo-os nas posições correspondentes.

Neste exercício você não precisa imprimir a lista a cada deslocamento de elementos, ao invés disso, imprima toda lista após cada inserção de elementos em suas posições corretas da lista ordenada.

Dica: inicie a ordenação por inserção a partir do segundo elemento da lista, pois o primeiro elemento naturalmente já será inserido na lista ordenada.

Entrada

A entrada é composta de apenas um caso de teste com 2 linhas.

Na primeira linha há um inteiro **N**, que consiste no comprimento da lista **L**.

A segunda linha terá uma sucessão de **N** valores inteiros separados por um espaço em branco cada, representando os **N** elementos da lista **L**.

Restrições

- $1 \leq N \leq 1000$
- $-10000 \leq L_i \leq 1000$

Saída

Seu programa deve gerar **N-1** linhas com lista **L** completa a cada vez que um novo elemento é inserido na posição correta da lista **L** ordenada. Após imprimir toda a lista **L**, salte uma linha.

Exemplos

Entrada	Saída
6	1 4 3 5 6 2
1 4 3 5 6 2	1 3 4 5 6 2
	1 3 4 5 6 2
	1 3 4 5 6 2
	1 2 3 4 5 6