

ORDENAÇÃO BUBBLE CHAR

OrdenaBubbleChar.[c | cpp | java | cs | py]

Faça um programa que receba um vetor de caracteres de tamanho **N** e ordene os seus elementos lexicograficamente crescente usando o algoritmo bolha (Bubble Sort).

Entrada

O programa terá apenas um caso de teste.

A primeira linha do caso de teste contém um inteiro **N**, $0 < N \leq 200$, representando o tamanho do vetor.

A segunda linha do caso de teste contém **N** caracteres separados por um espaço em branco cada.

Restrições:

- As **N** letras serão todas de 'A' até 'Z' ou de 'a' até 'z'.
- Cada entrada será composta por todas as letras maiúsculas ou minúsculas; ou seja, não haverá misturas de capitulação de letras.

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

A última linha trata-se de palavra "Trocas: X" seguida da quantidade de trocas que o algoritmo realizou. Observe que a letra T está em maiúsculo e após os dois pontos de há um espaço em branco antes do número inteiro **X**.

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 q d y c a u g o h b	q d y c a u g o h b d q y c a u g o h b d q c y a u g o h b d q c a y u g o h b d q c a u y g o h b d q c a u g y o h b d q c a u g o y h b d q c a u g o h y b d q c a u g o h b y d c q a u g o h b y d c a q u g o h b y d c a q g u o h b y d c a q g o u h b y d c a q g o h u b y d c a q g o h b u y c d a q g o h b u y c a d q g o h b u y c a d g q o h b u y c a d g o q h b u y c a d g o h q b u y c a d g o h b q u y a c d g o h b q u y a c d g h o b q u y a c d g h b o q u y a c d g b h o q u y a c d b g h o q u y a c b d g h o q u y a b c d g h o q u y a b c d g h o q u y Trocas: 27

Entrada	Saída
10 d d f i l n o p v y	d d f i l n o p v y d d f i l n o p v y Trocas: 0

Entrada	Saída
10 J I H G F E D C B A	J I H G F E D C B A I J H G F E D C B A I H J G F E D C B A I H G J F E D C B A I H G F J E D C B A I H G F E J D C B A I H G F E D J C B A I H G F E D C J B A I H G F E D C B J A I H G F E D C B A J H I G F E D C B A J H G I F E D C B A J H G F I E D C B A J H G F E I D C B A J H G F E D I C B A J H G F E D C I B A J H G F E D C B I A J H G F E D C B A I J G H F E D C B A I J G F H E D C B A I J G F E H D C B A I J G F E D H C B A I J G F E D C H B A I J G F E D C B H A I J G F E D C B A H I J F G E D C B A H I J F E G D C B A H I J F E D G C B A H I J F E D C G B A H I J F E D C B G A H I J

	F E D C B A G H I J
	E F D C B A G H I J
	E D F C B A G H I J
	E D C F B A G H I J
	E D C B F A G H I J
	E D C B A F G H I J
	D E C B A F G H I J
	D C E B A F G H I J
	D C B E A F G H I J
	D C B A E F G H I J
	C D B A E F G H I J
	C B D A E F G H I J
	C B A D E F G H I J
	B C A D E F G H I J
	B A C D E F G H I J
	A B C D E F G H I J
	A B C D E F G H I J
	Trocas: 45