Prova Substitutiva - Vetores e Strings

Prof. Dr. Gustavo Teodoro Laureano
Profa. Dra. Luciana Berretta
Prof. Dr. Thierson Rosa Couto

Sumário

1 Comparação de textos

 (++++)

Um sistema inteligente de reconhecimento de textos precisa de um algoritmo que seja capaz de comparar frases. Você, um excelente projetista de sistemas de reconhecimento de padrões, sugeriu o seguinte método de comparação: dados duas *strings* A e B, a distância entre as A e B pode ser calculada usando a distância euclidiana entre os vetores de frequência das vogais que compõem cada *string*. Um vetor de frequências das vogais "a, e, i, o, u" , ou suas maiúsculas, é um vetor com 5 posições, onde cada posição armazena a quantidade de vezes que as vogais aparecem na *string*. Por exemplo:

Seja A ="ola, meu nome e maria", possui um vetor de frequências de vogais $F_A = (3,3,1,2,1)$, ou seja, há 3 vogais "a", 3 vogais "e", 1 vogal "i", "2 vogais "o"e 1 vogal "u".

Para a *string* B ="era uma vez um lobo mal...", $F_B = (3,2,0,2,2)$.

A distância entre A e B é dada pela equação:

$$d(A,B) = \sqrt{\sum_{i=0}^{4}(F_A(i) - F_B(i))^2} \tag{1}$$

onde, $F_A(i)$ e $F_B(i)$ é a quantidade de vezes que a vogal i aparece nas *strings* A e B respectivamente.

Para o exemplo dado, o resultado da distância seria:

$$d(A,B) = \sqrt{(3-3)^2 + (3-2)^2 + (1-0)^2 + (2-2)^2 + (1-2)^2} = 1.732050808 \tag{2}$$

Faça um programa que leia duas *strings*, calcule e apresente a distância entre elas usando o método descrito.

Entrada

O programa deve ler uma linha contendo 2 *strings*, cada uma de no máximo 1000 caracteres, separadas pelo caracter ';'.

Saída

Se o texto informado não conter o caracter separador ';' ou mais de um caracter ';', o programa deve imprimir a mensagem "FORMATO INVALIDO!". Caso contrário, o programa deve apresentar 3 linhas. As duas primeiras devem conter os vetores de frequências de cada *string*, com os valores entre parênteses e separados por vírgulas, e a última linha deve conter o valor da distância entre as *strings* com 2 casas decimais.

Observações

O programa não deve diferencias maiúsculas de minúsculas. Também não não são admitidos acentos no texto de entrada.

Exemplo

Entrada
`Ola mundo, meu nome e Maria; Era uma vez um lobo mal...`
Saída

`(3,3,1,3,2)`
`(3,2,0,2,2)`
`1.73`

Entrada
Eu serei um grande Cientista da Computacao.
Saída
FORMATO INVALIDO!

2 Sentença Dançante

 (+++)

Uma sentença é chamada de dançante se sua primeira letra for maiúscula e cada letra subsequente for o oposto da letra anterior. Espaços devem ser ignorados ao determinar o case (minúsculo/maiúsculo) de uma letra. Por exemplo, "A b Cd"é uma sentença dançante porque a primeira letra (’A’) é maiúscula, a próxima letra (’b’) é minúscula, a próxima letra (’C’) é maiúscula, e a próxima letra (’d’) é minúscula.

Entrada

A entrada contém vários casos de teste. Cada caso de teste é composto por uma linha que contém uma sentença, que é uma string que contém entre 1 e 50 caracteres (’A’-’Z’,’a’-’z’ ou espaço ’ ’), inclusive, ou no mínimo uma letra (’A’-’Z’,’a’-’z’). A entrada termina por fim de arquivo.

Saída

Transforme a sentença de entrada em uma sentença dançante (conforme o exemplo abaixo) trocando as letras para minúscula ou maiúscula onde for necessário. Todos os espaços da sentença original deverão ser preservados, ou seja, "sentence "deverá ser convertido para "SeNtEnCe ".

Exemplo

Entrada	Saída

`This is a dancing sentence`
`  This is a dancing sentence`
`aaaaaaaaaaa`
`z`
	`ThIs Is A dAnCiNg SeNtEnCe`
	`  ThIs Is A dAnCiNg SeNtEnCe`
	`AaAaAaAaAaA`
	`Z`

3 Subtração e produto de conjuntos

 (++)

Faça um programa que leia 2 conujuntos (A e B) válidos, sem elementos repetidos, cada um com no mínimo 1 e no máximo 100 elementos, e imprima A, B e $A - B$.

Entrada

O programa deve ler um número inteiro T_A, correspondente ao tamanho do conjunto A, até que T_A seja válido, em seguida outro número inteiro T_B, correspondente ao tamanho do conjunto B até que T_B seja válido. Uma vez definido os tamanhos dos vetores, o programa deve ler $T_A + T_B$ elementos, correspondentes aos elementos de A e B. Durante a leitura dos elementos de um conjunto, o programa deve permitir somente a leitura de elementos diferentes aos já presentes no conjunto. Caso um elemento lido já esteja presente no conjunto, o programa deve ignorá-lo e realizar uma nova leitura do elemento.

Saída

O programa deve apresentar na tela quatro linhas. A primeira com o conjunto A, a segunda com o conjunto B, a tereceira com o conjunto $A - B$ e a quarta com o conjunto $A \times B$. O conjunto $A - B$ é formado por todos os elementos que ocorrem em A e que não ocorrem em B. O conjunto $A \times B$ é formado por todas as combinações em pares dos conjuntos de A com B no formato $(a_i x b_j)$ onde i é o i-ésimo elemento de A e j é o j-ésimo elemento de B. Os elementos dos conjuntos devem ser apresentados entre parênteses, separados por vírgula e sem espaços.

Observações

Não se esqueça que um conjunto válido não permite a existência de elementos repetidos.

Exemplo

Entrada	Saída
3 2 1 2 3 1 2	(1,2,3) (1,2) (3) ((1x1),(1x2),(2x1),(2x2),(3x1),(3x2))

Entrada	Saída
0 0 1001 2 -1 4 5 9 0 5 7 2	(5,9) (0,5,7,2) (9) ((5x0),(5x5),(5x7),(5x2),(9x0),(9x5),(9x7),(9x2))

Entrada	Saída
0 0 1001 2 -1 4 5 9 9 5 0 0 0 7	(5,9) (9,5,0,7) () ((5x9),(5x5),(5x0),(5x7),(9x9),(9x5),(9x0),(9x7))

Entrada	Saída
1 1 5 9	(5) (9) (5) ((5x9))