

Prova 05

Prof. Dr. Gustavo Teodoro Laureano

Profa. Dra. Luciana Berretta

Prof. Dr. Thierson Rosa Couto

Sumário

1	Comparação de textos	2
2	União e intersecção de conjuntos	4

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

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

A distância entre A e B é dada pela equação:

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

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:

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

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 diferenciar maiúsculas de minúsculas. Também 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 União e intersecção de conjuntos



(++++)

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

Entrada

Durante a entrada de dados, o programa deve ler tamanhos válidos para os conjuntos A e B . Caso o tamanho informado seja inválido, o programa deve refazer a leitura do tamanho do conjunto. A leitura deve ser feita separadamente. Uma vez definido os tamanhos dos vetores, o programa deve ler os elementos de cada vetor. Em seguida, o programa deve apresentar na tela uma linha com o conjunto união, ou seja, todos os elementos que ocorrem em A e B , e outra linha contendo o conjunto intersecção, ou seja, todos os elementos que ocorrem em A e B .

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 uma linha com o conjunto união, ou seja, todos os elementos que ocorrem em A e B , e outra linha contendo o conjunto intersecção, ou seja, todos os elementos que ocorrem em A e 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
3
2
1 2 3
1 2
Saída
(1, 2, 3)
(1, 2)

Entrada
0
0
1001
2
-1
4
5 9
0 5 7 2
Saída
(5, 9, 0, 7, 2)
(5)

Entrada
0
0
1001
2
-1
4
5 9 9 5
0 0 0 7 7 2 5 7 2
Saída
(5, 9, 0, 7, 2)
(5)

Entrada
1
1
5
9
Saída
(5, 9)
()