

Prova resgate

Prof.: Carlos Camarão

15 de Dezembro de 2009

1. Escreva um programa que leia um número inteiro positivo n , em seguida n valores inteiros v_1, \dots, v_n , depois um número positivo m e imprima:

$$(v_1 + v_2 + \dots + v_m) \times (v_{m+1} + v_{m+2} + \dots + v_{m+m}) \times \dots$$

até que se tenha m ou menos valores restantes, e nesse último passo deve-se multiplicar a soma desses m ou menos valores restantes.

Os valores devem ser lidos do dispositivo de entrada padrão e podem estar separados entre si por um ou mais espaços ou linhas.

Por exemplo, para a entrada:

5 1 2 3 4 5 2

a saída deve ser: 105.

Isso ocorre porque $105 = (1 + 2) \times (3 + 4) \times 5$.

2. Escreva um programa que leia um texto em português, e imprima uma lista que indique, para cada tamanho de palavra, o número de vezes que uma palavra desse tamanho ocorre no texto.

Tamanhos que nunca ocorrem no texto não devem fazer parte da lista.

Você pode considerar que caracteres de pontuação sempre ocorrem apenas no final de uma palavra, e não devem ser considerados como parte da palavra.

Por exemplo, para a entrada:

Hoje estou fazendo uma prova de PC; está legal.

A saída deve ser:

Tamanho 2: 2 ocorrências
Tamanho 3: 1 ocorrência
Tamanho 4: 2 ocorrências
Tamanho 5: 3 ocorrências
Tamanho 7: 1 ocorrência

No exemplo, temos:

As ocorrências de palavras de tamanho 2 são: **de, PC**
A única ocorrência de palavra de tamanho 3 é: **uma**
As ocorrências de palavras de tamanho 4 são: **Hoje, está**
As ocorrências de palavras de tamanho 5 são: **estou, prova, legal**
A única ocorrência de palavra de tamanho 7 é: **fazendo**

O texto deve ser lido do dispositivo de entrada padrão, usando o método **next** da classe **Scanner** para ler uma palavra (considere que **next** retorna a próxima palavra do texto, incluindo possível caractere de pontuação no final da palavra, e avança a posição de leitura para a próxima palavra). Por exemplo, ao chamar o método **next** para ler "PC;", a cadeia de caracteres "PC;" é retornada, incluindo o caractere ";" como último caractere da cadeia.

O fim do texto deve ser testado por meio do uso do método **hasNext** da classe **Scanner**.

Considere que palavras têm no máximo 30 caracteres.