

Prova de Programação de Computadores

Prof.: Carlos Camarão

01 de Dezembro de 2009

1. Escreva um programa que leia um número inteiro positivo n , em seguida n valores inteiros positivos v_1, \dots, v_n , depois um valor inteiro m e um valor inteiro positivo k , e imprima todos os valores dentre v_1, \dots, v_n que são maiores que m .

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

A saída deve imprimir os valores separados por um espaço, e contendo no máximo k valores por linha: a cada k valores impressos deve ser impresso um caractere de terminação de linha.

Por exemplo, para a entrada:

10 1 3 5 7 9 6 7 8 9 10 5 2

a saída deve ser:

7 9
6 7
8 9
10

(7,9,6,7,8,9,10 são os valores maiores que 5 dentre os 10 valores lidos).

2. Escreva um programa que leia um valor inteiro positivo n , em seguida n valores inteiros positivos v_1, v_2, \dots, v_n , depois um valor positivo m , e imprima o valor resultante do cálculo de:

$$((m-1) \times v_1) + ((m-2) \times v_2) + \dots + ((m-n) \times v_n)$$

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:

4 1 2 3 4 10

a saída deve ser: 70 (pois esse é o resultado de $9 \times 1 + 8 \times 2 + 7 \times 3 + 6 \times 4$).

3. Escreva um programa que leia um inteiro positivo n , depois n valores inteiros a_1, \dots, a_n , e imprima uma mensagem indicando se a sequência de valores a_1, \dots, a_n é não-decrescente (ou seja, cada valor a_{i+1} é maior ou igual a a_i , para $i = 1, \dots, n - 1$), não-crescente (ou seja, cada valor a_{i+1} é menor ou igual a a_i , para $i = 1, \dots, n - 1$), constante (todos os valores a_i são iguais entre si, para $i = 1, \dots, n$), ou nenhuma dessas opções (nem não-decrescente, nem não-decrescente, nem constante).

Por exemplo, para a entrada:

4 1 2 3 3

a saída deve ser: "Não-decrescente" (pois a sequência dos 4 valores 1,2,3,3 é não-decrescente).

Para a entrada:

4 1 2 5 3

a saída deve ser: "Nem não-decrescente, nem não-crescente, nem constante" (pois a sequência dos 4 valores 1,2,5,3 não é não-decrescente, nem não-decrescente, nem constante).

Dica: Use três booleanos `naoCrescente`, `naoDecrescente`, `constante`, inicialize-os com `true` e atualize-os ao percorrer (a partir do segundo valor da sequência) a sequência de valores (armazenada em um arranjo) examinando valores adjacentes da sequência.