Programação I

Folha exercícios 3 - Recursividade

Exercício 1:

Desenvolver um programa que, dando o valor de *x* e *y* calcule x^y recorrendo ao uso da recursividade. Desrecursive a função implementada anteriormente.

Exercício 2:

Implementar uma função recursiva que calcule o tamanho de uma string. Desrecursive a função implementada anteriormente.

Exercício 3:

Implementar uma função recursiva que calcule o somatório de um array de inteiros. Os números do array deverão ser introduzidos pelo utilizador.

Exercício 4:

Implementar um programa que leia do teclado um número inteiro positivo e escreva para o ecrã o respectivo número triangular. Os números triangulares são definidos pela seguinte relação de recorrência:

$$\begin{cases} \text{Se } n=1,\ T(n)=1 \\ \text{Se } n>1,\ T(n)=T(n-1)+n \end{cases}$$

Desrecursive a função implementada anteriormente.

Exercício 5:

Implementar um programa que leia do teclado um número inteiro positivo e escreva para o ecrã o respectivo número quadrático. Os números quadráticos são definidos pela seguinte relação de recorrência:

$$\begin{cases} \text{Se } n=1,\ Q(n)=1 \\ \text{Se } n>1,\ Q(n)=Q(n-1)+2n-1 \end{cases}$$

Desrecursive a função implementada anteriormente.

Exercício 6:

Implementar um programa que leia do teclado um número inteiro positivo e escreva para o ecrã o número definido pela seguinte relação de recorrência:

$$\begin{cases} \text{Se } n <= 10 \rightarrow Seq(n)=10 \\ \text{Se } n > 10 \rightarrow Seq(n)=2n - Seq(n-1) + 4 \end{cases}$$

Desrecursive a função implementada anteriormente.

Exercício 7:

O **máximo divisor comum** de dois números inteiros positivos pode ser calculado, utilizando o método de Euclides, cujo algoritmo é dado pela seguinte relação de recorrência:

$$\text{mdc}(m, n) = \begin{cases} m, \text{ se } n = 0 \\ \text{mdc}(n, m\%n), \text{ se } n \neq 0 \end{cases}$$

Escrever um programa que leia do teclado dois números inteiros positivos, calcule e escreva para o ecrã o seu máximo divisor comum. Implemente uma versão do programa usando um algoritmo iterativo e outra versão usando um algoritmo recursivo.

Exercício 8 :

Escreva uma função recursiva para calcular o resultado da fórmula para n:

$$\sum_{i=1}^{n} (2i^2 + 3)$$

Desrecursive a função anterior.

Exercício 9:

Desenvolva um algoritmo recursivo para a geração do output de uma sequência de valores inteiros, iniciada por um valor positivo, introduzido pelo utilizador, e que varia sequencialmente até ao valor zero. A sequência numérica deve ser apresentada verticalmente.

Exercício 10:

Dada a definição da função de ***Ackermann***

$$A(m,n) = \begin{cases} n+1 \; se \; m = 0 \\ A(m-1,1) \; se \; m > 0 \; e \; n = 0 \\ A(m-1, A(m, n-1)) \; se \; m > 0 \; e \; n > 0 \end{cases}$$

A função é válida para inteiros não negativos de m e n, implemente uma versão recursiva do algoritmo.

Exercício 11:

Implemente uma função recursiva que recebe um número inteiro e escreva para o ecrã o correspondente número binário. O número deverá ser introduzido pelo utilizador.

Exercício 12:

Implemente uma função recursiva que receba dois números inteiros: x e y. A função deve calcular o resto da divisão inteira de x por y.

MOD(x,y): se x=y ->0; se x <y -> x senão MOD(x-y,y)