

## FUNÇÕES. RECURSIVIDADE.

1) Construa um programa em C++ que, dado um número inteiro  $n$ , calcule  $n!$ , utilizando uma função não recursiva.

2) Construa um programa em C++ que ordene por ordem crescente uma sequência de  $n$  números reais dados, pelo algoritmo de procura directa. Estruture o programa utilizando funções.

3) Construa um programa em C++ que, dado um número inteiro  $n$ , calcule  $n!$ , utilizando uma função recursiva.

4) Construa um programa em C++ que ordene por ordem crescente uma sequência de  $n$  números reais dados, utilizando o algoritmo *quicksort*. Apresente uma simulação da execução do programa e prepare o código para obter como output os passos intermédios dessa simulação.

5) Escreva um programa em C++ com estrutura recursiva que calcule  $n$  termos da sucessão de *Fibonacci* e verifique as duas propriedades conhecidas desta sucessão (ver o exercício 12 da Folha IV).

6) O triângulo de Pascal, cujos elementos representam os coeficientes do binómio de Newton, baseia-se na seguinte propriedade das combinações :

$$C_n^m = C_{n-1}^m + C_{n-1}^{m-1}.$$

Escreva um programa em C++ com estrutura recursiva que, para um número natural  $n$  dado, calcule os coeficientes do binómio de Newton  $(a + b)^n$ . Prepare o programa para fornecer como output o desenvolvimento do binómio de Newton e num ficheiro de output o respectivo triângulo de Pascal associado.

7) Construa um programa em C++ que calcule o maior divisor comum de dois números naturais dados.

8) Construa um programa em C++ que, dada uma sequência de nomes (*strings*) os ordena alfabeticamente.

9) Construa um programa em C++ para determinar como se devem colocar oito rainhas na tábua de xadrez, sem se atacarem uma a outra. Prepare o programa para fornecer todas as soluções possíveis.