

## 8.4 Exercícios da Aula

Parte dos exercícios desta lista foram Adaptados de [Tenenbaum, Langsam e Augenstein \(1995, p. 143-178\)](#) e [Edelweiss e Livi \(2014, p. 433-436\)](#).

1. Faça um programa em C que calcule, por meio de uma função recursiva,  $a \times b$  usando a adição, em que **a** e **b** são inteiros não-negativos.
2. Crie uma função recursiva que receba um número inteiro positivo **N** e calcule o somatório dos números de 1 a **N**.
3. Considere um vetor **vet** de tamanho 20. Construa um programa com algoritmos recursivos para calcular:
  - o elemento máximo do vetor;
  - o elemento mínimo do vetor;
  - a soma dos elementos do vetor;
  - o produto dos elementos do vetor;
  - a média dos elementos do vetor.
4. A sequência de Fibonacci é a sequência de inteiros: 0, 1, 1, 2, 3, 5, 8, 13, 21, 34, .... Implemente uma função recursiva que calcule e imprima todos os elementos da série Fibonacci de 0 até **n**. Em que, **n** deve ser informado pelo usuário do programa.
5. Escreva uma função recursiva em C para calcular o máximo divisor comum de dois números,  $\text{mdc}(x, y)$ .
6. Escreva um programa recursivo em linguagem C para converter um número da sua forma decimal para a forma binária. Dica: dividir o número sucessivamente por 2, sendo que o resto da *i*-ésima divisão vai ser o dígito **i** do número binário (da direita para a esquerda).
7. Escreva uma função recursiva em linguagem C para calcular o valor de  $x^n$
8. Escreva um programa em C recursivo que inverta a ordem dos elementos, números inteiros, de uma lista armazenada em um vetor. Ao final da execução, o conteúdo do primeiro elemento deverá estar no último, o do segundo no penúltimo, e assim por diante. Dica: troque os conteúdos das duas extremidades do vetor e chame uma função recursivamente para fazer o mesmo no subvetor interno.
9. Escreva uma função recursiva para calcular a função de Ackermann **A(m,n)**, sendo **m** e **n** valores inteiros não negativos, dada por:

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

10. Imagine que  $\text{comm}(n, k)$  representa o número de diferentes comitês de **k** pessoas, que podem ser formados, dadas **n** pessoas a partir das quais escolher. Por exemplo,  $\text{comm}(4, 3) = 4$ , porque dadas quatro pessoas, **A**, **B**, **C** e **D** existem quatro possíveis comitês de três pessoas: **ABC**, **ABD**, **ACD** e **BCD**. Escreva e teste um programa

recursivo em C para calcular  $\text{comm}(n,k)$  para  $n, k \geq 1$ . Para tal, considere a seguinte identidade:

$$\text{comm}(n,k) = n \quad \text{se } k = 1$$

$$\text{comm}(n,k) = 1 \quad \text{se } k = n$$

$$\text{comm}(n,k) = \text{comm}(n-1, k) + \text{comm}(n-1, k-1) \quad \text{se } 1 < k < n$$