



## INSTITUTO DE ENGENHARIAS E CIÊNCIAS DO MAR

### Ficha de Trabalho #1

▷ **Tópicos:** Recursividade

1. Elabore uma função que calcule por recorrência potências de 2.
2. Elabore um procedimento recursivo que conte por ordem decrescente de  $N$  para 0.
3. Elabore um procedimento recursivo que imprima um número por ordem inversa. Por exemplo, 123456 deverá ser impresso como 654321.
4. Elabore um procedimento recursivo que imprima um determinado vetor por ordem inversa. Por exemplo:  $v = \{35, 45, 23, 78, 90, 65, 78, 90, 76, 34, 25, 100\}$  será impresso 100 25 34 76 90 78 65 90 78 23 45 35
5. Elabore uma função recursiva que conte os elementos de um vetor de inteiros que sejam inferiores a um dado número.
6. Elabore uma função recursiva que calcule o somatório dos elementos de um vetor.
7. Escreva um programa que calcule o máximo divisor comum de dois números inteiros positivos, baseando-se nas seguintes propriedades:

$$\text{mdc}(m, n) = \begin{cases} m & \text{se } n = 0 \\ \text{mdc}(n, m \bmod n) & \text{se } n > 0 \end{cases}$$

8. Implemente uma função recursiva que permita calcular a potência:

$$x^n = \begin{cases} 1 & \text{se } n = 0 \\ x^{n-1} * x & \text{se } n > 0 \\ x^{n+1}/x & \text{se } n < 0 \end{cases}$$

9. Pretende-se implementar uma função recursiva que recebe dois números inteiros positivos calcule o valor da função de Ackermann. A função de Ackermann é definida pela seguinte relação de recorrência:

$$Cn = \begin{cases} n + 1 & \text{se } m = 0 \\ \text{Ackermann}(m - 1, 1), & \text{se } n = 0 \\ \text{Ackermann}(m - 1, \text{Ackermann}(m, n - 1)), & \text{com } n > 0 \text{ e } m > 0 \end{cases}$$

10. Escreva uma função que devolva o número de Fibonacci associado a um determinado inteiro positivo. A sequência de Fibonacci, é definida da seguinte forma:

$$F(n) = \begin{cases} 0 & \text{se } n = 0, \\ 1 & \text{se } n = 1, \\ F(n-1) + F(n-2) & \text{se } n > 1 \end{cases}$$

11. O puzzle Torres de Hanoi com 3 colunas resolve-se, para um número  $n$  de peças (discos), em  $h(n) = 2 \times h(n-1) + 1$  movimentos (para  $n > 1$ ) e  $h(1) = 1$ . Desenvolva uma função recursiva para determinar  $h(n)$  e caso  $n < 1$  a função deve devolver -1. O protótipo da função a desenvolver é: `int hanoi(int n)`