

4 Funções

22 Escreva uma função que faça a troca do valor de duas variáveis, mas sem usar variáveis auxiliares, isto é, utilizando somente as operações aritméticas elementares.

23 Implemente as seguintes funções:

- | | |
|---|--|
| <p>1. $\log : \mathbb{R}^+ \times A \rightarrow \mathbb{R}$
 $(x, a) \mapsto \log_a(x) = \frac{\ln(x)}{\ln(a)}$
 com $A = \{x \in \mathbb{R} : x > 0 \text{ e } x \neq 1\}$</p> | <p>3. $f : \mathbb{R} \rightarrow \mathbb{R}$
 $x \mapsto \begin{cases} \sqrt{x} & \text{se } x \geq 0 \\ \sqrt{-x} & \text{se } x < 0 \end{cases}$</p> |
| <p>2. $\sinh : \mathbb{R} \rightarrow \mathbb{R}$
 $x \mapsto \sinh(x) = \frac{e^x - e^{-x}}{2}$</p> | <p>4. $\operatorname{argsech} :]0, 1] \rightarrow \mathbb{R}$
 $x \mapsto \operatorname{argsech}(x) = \log\left(\frac{1 + \sqrt{1 - x^2}}{x}\right)$</p> |

24 Números Amigos.

1. Dados dois números inteiros positivos escreva uma função que verifique se os dois números são ou não amigos, isto é, se cada um deles é igual à soma dos divisores próprios do outro.
2. Usando a alínea anterior construa um programa que escreva todos os pares de números amigos existentes entre 1 e 1000. Tente otimizar este programa.

5 Recursão

25 Elabore um programa que leia uma palavra e a escreva invertida, por exemplo com a palavra “programa”, ter-se-ia o resultado “amargorp”.

26 Elabore um programa que calcule os n primeiros termos da série harmónica:

$$1 + \frac{1}{2} + \frac{1}{3} + \frac{1}{4} + \dots + \frac{1}{n} + \dots$$

27 Escreva uma função que calcule o factorial de um número. Usando essa função escreva um programa que calcule o número de combinações de m , n a n sabendo que:

$$C_n^m = \frac{m!}{n!(m-n)!}$$

28 Escreva um programa que calcule o m.m.c de dois números inteiros positivos, baseando-se nas seguintes propriedades:

$$\text{mmc}(a, b) = \frac{a \times b}{\text{mdc}(a, b)} \quad \text{e} \quad \text{mdc}(a, b) = \begin{cases} a & \text{se } b = 0 \\ \text{mdc}(b, a \bmod b) & \text{se } b \neq 0 \end{cases}$$

29 O sub-factorial ($!n$) de um número inteiro não negativo é definido por:

$$!n = \begin{cases} 1 & \text{se } n = 0 \\ !(n-1)n + (-1)^n & \text{se } n > 0 \end{cases}$$

Escreva uma função recursiva para calcular $!n$.

30 Implemente uma função recursiva que permita calcular a potência:

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

para n inteiro e x real.

31 Seja $\text{Comissões}(m, n)$ o número de diferentes comissões de n pessoas que se podem formar por eleição entre m pessoas. Por exemplo, $\text{Comissões}(4, 3) = 4$, porque dadas quatro pessoas, **A**, **B**, **C** e **D**, há quatro possíveis comissões de três membros, **ABC**, **ABD**, **ACD** e **BCD**. É sabido que:

- $\text{Comissões}(m, n) = 1$ quando $m = n$;
- $\text{Comissões}(m, n) = m$ quando $n = 1$;
- $\text{Comissões}(m, n) = \text{Comissões}(m-1, n-1) + \text{Comissões}(m-1, n)$ quando $m > n > 1$.

Escreva uma função recursiva em C que calcule $\text{Comissões}(m, n)$ para $m \geq n \geq 1$.

32 Implemente uma função recursiva que permita escrever, por ordem inversa, os dígitos de um número inteiro.

33 Para uso dos alunos do Ensino Primário, pretende-se imprimir uma tabuada de multiplicar. Por exemplo, para $n = 5$, tal tabuada terá o seguinte formato:

```

1
2 4
3 6 9
4 8 12 16
5 10 15 20 25

```

Elabore uma função recursiva para imprimir uma tabuada de tamanho n , de cabeçalho:

```
void imprimirTabuada (int n);
```

34 Considere a seguinte função, definida em \mathbb{N} :

$$P(n) = \prod_{i=1}^n (2i - 1)$$

1. Elabore uma função recursiva para o seu cálculo.
2. Construa uma versão iterativa da mesma função.