

### **Instruções de repetição**

**Construir programas em linguagem C para resolver os problemas que se seguem.**

- 1.** Construa um programa para mostrar 20 vezes o seu nome no écran.
- 2.** Construa um programa para mostrar os 100 primeiros números naturais (1, ..., 100).
- 3.** Construa um programa para mostrar os números naturais ímpares até 100.
- 4.** Construa um programa para mostrar os 100 primeiros números naturais ímpares.
- 5.** Construa um programa para calcular a soma dos N primeiros números naturais.
- 6.** Construa um programa para calcular a soma dos N primeiros números naturais pares.
- 7.** Construa um programa para calcular o produto dos números naturais ímpares e o produto dos números naturais pares até 1000.
- 8.** Construa um programa para determinar a soma de N números reais, em que  $N \geq 2$ . Se  $N < 2$  será mostrada uma mensagem de ERRO e pedido um novo valor para N.
- 9.** Construa um programa que permita determinar o maior número de uma sequência de N ( $N \geq 1$ ) números inteiros dados pelo utilizador. Se  $N < 1$  deverá ser pedido novamente até obter um valor válido. A introdução de números deverá terminar quando forem inseridos N números. Nessa altura a aplicação deverá mostrar o resultados obtido (maior valor introduzido).
- 10.** Construa um programa para determinar a soma de uma sequência de inteiros positivos (a sequência termina quando for inserido um número negativo, que não é somado).
- 11.** Construa um programa para determinar a média de N números reais, dados um de cada vez. O valor de N deverá ser lido antes dos N reais (primeiro valor a ser inserido).
- 12.** Construa um programa para determinar o produto dos números inteiros entre N1 e N2, com  $N1 < N2$ .
- 13.** Construa um programa para calcular o produto e a soma dos números inteiros pares entre N1 e N2, com  $N1 < N2$ .
- 14.** Construa um programa para determinar o maior e o menor números de uma sequência de N números reais, com  $N > 1$ .
- 15.** Construa um programa para determinar a média aritmética de uma sequência de N números reais positivos, em que  $N > 1$ . A introdução de números terminar com um número menor ou igual a 0.

- 16.** Construa um programa para determinar o maior e o menor número de uma sequência de  $N$  ( $N \geq 2$ ) números inteiros dados pelo utilizador. Se  $N < 2$  deverá ser pedido novamente até obter um valor válido. A introdução de números deverá terminar quando forem inseridos  $N$  números. Nessa altura a aplicação deverá mostrar os resultados obtidos (maior e menor valores introduzidos).
- 17.** Construa um programa para determinar o maior e o menor números de uma sequência (com pelo menos 1 elemento) de números inteiros dados pelo utilizador. A introdução de números deverá terminar quando for inserido o valor zero.
- 18.** Construa um programa para calcular a soma dos dígitos de um número inteiro positivo.
- 19.** Construa um programa para determinar o maior, em valor absoluto, entre  $N$  números reais fornecidos um de cada vez.
- 20.** Construa um programa para determinar o maior e o menor, em valor absoluto, entre  $N$  números reais, fornecidos um de cada vez.
- 21.** Construa um programa para determinar o maior e o menor, em valor relativo, entre  $N$  números reais, fornecidos um de cada vez.
- 22.** Construa um programa para calcular o factorial de um número inteiro positivo. (Nota: caso seja introduzido um número negativo, este deverá ser rejeitado e introduzido um outro; este procedimento deverá ser repetido até ser introduzido um número positivo).
- 23.** Construa um programa para calcular o valor da seguinte expressão (com  $N > 0$ ):

$$\text{soma} = \sum_{k=1}^N \frac{1}{k}.$$

- 24.** Calcular o valor da seguinte expressão (com  $N > 0$ ):

$$\text{soma} = \sum_{k=1}^{10} \frac{1}{k^N}.$$

- 25.** Construa um programa que sabendo que o valor de  $e^x$  pode ser calculado através da fórmula que se segue

$$e^x = \sum_{n=0}^{\infty} \frac{x^n}{n!} = 1 + \frac{x^2}{2!} + \frac{x^3}{3!} + \dots, \quad \text{em que, } \frac{x^k}{k!} = \frac{x^{k-1}}{(k-1)!} \cdot \frac{x}{k}$$

calcular  $e^x$  para um dado valor  $x$ , com erro inferior a  $10^{-5}$ , e mostrar o resultado e o número de iterações necessário.

- 26.** Construa um programa para escrever uma tabela de senos e de cosenos de valores de  $x \in [0, \pi/2]$  com intervalos de variação de 0.001.

- 27.** Construa um programa para determinar o máximo divisor comum entre 2 números inteiros positivos, usando o algoritmo de Euclides, que se enuncia da seguinte forma:  $m.d.c.(a, b) = a$ , se  $b = 0$ , ou  $m.d.c.(a, b) = m.d.c.(b, a \% b)$ , caso contrário.
- 28.** Construa um programa para mostrar os primeiros N números da sequência de Fibonacci. A sequência de Fibonacci é a seguinte: 1, 1, 2, 3, 5, 8, 13, 21, ... (a seguir ao primeiro 1 cada número na sequência é a soma dos dois números anteriores).
- 29.** O dono de um hotel resolveu organizar os seus preços de uma forma bastante original:
- no 1º andar cobrava 500€ (euros) ,
  - no 2º andar cobrava 125€,
  - de uma maneira geral, no n-ésimo andar cobrava  $(500/n^2)$ €.
- Construa um programa para determinar quantos andares tem o hotel para que no último andar o preço seja 5€.
- 30.** Construa um programa para converter um número binário em número decimal.  
Por exemplo,  $10010_2 = 1 \times 2^4 + 0 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 0 \times 2^0 = 18_{10}$ .
- 31.** Construa um programa para converter um número decimal num número binário.
- 32.** Construa um programa para determinar se um dado número inteiro positivo é primo. Um número é primo se for divisível apenas por ele próprio e pela unidade (por exemplo: 13).
- 33.** Construa um programa para verificar se um número inteiro positivo é capicua. Um número é capicua se for o mesmo quando lido da esquerda para a direita e vice-versa (por ex: 232).
- 34.** Construa um programa que mostre a tabuada de N. Para isso deverá pedir ao utilizador o valor de N que deverá ser obrigatoriamente maior ou igual 1 e menor ou igual a 10. A tabuada apresentada é o exemplo para  $N = 2$ .

```
2 x 1 = 2
2 x 2 = 4
2 x 3 = 6
2 x 4 = 8
2 x 5 = 10
2 x 6 = 12
2 x 7 = 14
2 x 8 = 16
2 x 9 = 18
2 x 10 = 20
```