

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 calcular e mostrar a soma dos 100 primeiros números naturais.
4. Construa um programa para calcular a soma e o produto dos números naturais ímpares até 1000 e, depois, mostrar os resultados obtidos.
5. Construa um programa para ler uma sequência de inteiros positivos (ou seja, termina com a introdução de um valor negativo) e calcular a sua soma.
6. Construa um programa para determinar a soma de N números. O programa deverá mostrar uma mensagem de ERRO se N for inferior a 2 e voltar a pedir o seu valor até ser inserido um valor válido.
7. Construa um programa para 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).
8. 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).
9. Construa um programa para determinar o produto e a soma dos números inteiros positivos não nulos (> 0) pares entre N1 e N2. A aplicação deverá dar erro se $N2 \leq N1$ e voltar a pedir estes valores. Deverá no final apresentar o produto e a soma calculados..
10. Construa um programa para determinar o maior e o menor número de uma sequência de N (com $N \geq 2$) 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).
11. Construa um programa para calcular o factorial de um número inteiro positivo não nulo (> 0).
12. Construa um algoritmo que permita determinar o maior e o menor número de uma sequência de números inteiros dados pelo utilizador. A introdução de números deverá terminar quando for inserido o valor zero e nessa altura a aplicação deverá mostrar o valor máximo e mínimo de entre os valores introduzidos.
13. Construa um programa para determinar a média aritmética duma sequência de N ($N \geq 2$) números inteiros. Se $N < 2$ então deverá ser pedido novamente até obter um valor válido. A

introdução de números deverá terminar quando forem inseridos N números e nessa altura a aplicação deverá mostrar o valor da média.

14. Construa um programa para determinar a média aritmética de uma sequência de números inteiros. A introdução de números deverá terminar quando for inserido o valor zero. Nessa altura a aplicação deverá mostrar o valor da média. Deverão ter sido introduzidos pelo menos dois números, caso contrário deverá ser mostrado um erro.
15. Construa um programa para determinar a soma dos dígitos de um número inteiro positivo.
16. 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).
17. Construa um programa para converter um número binário em número decimal. Por exemplo, $10010_2 = 1 \times 2^4 + 0 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 0 \times 2^0 = 18_{10}$.
18. Construa um programa para converter um número em formato decimal num número em formato binário.
19. Construa um programa para determinar se um dado número inteiro positivo não nulo (> 0) é primo. Um número é primo se for **apenas** divisível por ele próprio e pela unidade (por exemplo: 13). Caso o número introduzido seja negativo, a aplicação deverá pedir outro número até que seja introduzido um número válido.
20. Construa um programa para determinar se um número inteiro positivo é capicua. Um número é capicua se for o mesmo número quando lido da esquerda para a direita ou vice-versa (por exemplo: 202).
21. Construa um programa que sabendo que o valor de e^x pode ser calculado através da fórmula que se segue

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

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

22. 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.
23. 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€, e 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€.
24. 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).