

Tema(s):

- Funções.
- Biblioteca standard.
- Programação modular.

Parte 1

1. Implemente um programa que leia um valor inteiro do utilizador e crie uma função que imprima numa linha um número de asteriscos igual ao valor inserido pelo utilizador. O valor inteiro deve ser lido através de uma função.
2. Implemente um programa que solicite dois números inteiros ao utilizador e que permite depois ao utilizador escolher entre 4 operações aritméticas: adição, subtração, multiplicação e divisão. Cada uma dessas operações deve ser implementada através de uma função específica para esse propósito.
3. Implemente e use um módulo (**myIntLib**) com as funções aritméticas do exercício anterior.
4. Implemente funções que, dadas 2 matrizes de inteiros com a mesma dimensão bem como a sua dimensão, apresente (numa função) em forma de matriz o resultado da soma das 2 matrizes e (noutra função) o somatório dos seus elementos.
5. Faça a traçagem da função apresentada na página 6 dos slides teóricos sobre recursividade.

Parte 2

1. Altere a função do exercício 1 de modo a validar o intervalo de valores permitido. A função deve receber dois parâmetros que identifiquem o intervalo de valores permitidos. Acrescente a função ao módulo (**myIntLib**). Teste a implementação.
2. Implemente um programa que leia as notas de um número aleatório de alunos. Considere que as notas têm uma gama de valores de 0-20 utilizando o módulo **myIntLib**. Apresente a média das notas lidas.
3. Acrescente ao módulo **myIntLib** duas funções que retornem o valor máximo e mínimo de dois valores inteiros passos como argumentos. Teste a implementação como valores lidos do utilizador.
4. Considerando a seguinte definição matemática da função recursiva para encontrar a potência de um número. A função aceita dois números **x** e **y** e calcula **x^y**. Acrescente esta função ao módulo.

$$f(x, y) = \begin{cases} 1, & y = 0 \\ x \times f(x, y - 1), & y > 0 \\ \frac{1}{f(x, -y)}, & y < 0 \end{cases}$$

5. Implemente um menu para as funções do módulo que implementou, criando desta forma uma calculadora com diversas funcionalidades. Os valores lidos deve usar a função implementada no exercício 1.

Parte 3 (opcional)

1. Pretende-se que altere a aplicação criada para resolver o problema da Parte 3 da Ficha Prática 4 e que adote princípios de modularidade no código com recurso à implementação de funções e procedimentos.

Sugestões:

- a) Crie uma função para a resolução de cada uma das seguintes alíneas: b) a e)
- b) Crie uma função que tenha como parâmetros um texto a apresentar ao utilizador, bem como um limite numérico inferior e superior e valide se o valor introduzido está dentro desse limite. Use essa função na leitura do código do funcionário e na identificação do número de dias trabalhados.