

# Programação 2022/23

## LEI, LEI-PL, LEI-CE

### Guião laboratorial n.º 0

#### Diagnóstico do conhecimento atual da linguagem C

##### Tópicos da matéria:

Desenvolvimento de programas simples em linguagem C: Funções, Tabelas, Matrizes e Strings

##### Bibliografia:

Luís Damas. *Linguagem C*. FCA.

K. N. King. *C programming: A Modern Approach* (2<sup>nd</sup> Edition). W. W. Norton: capítulos 1 a 9 e 13.

---

#### A – Tabelas Unidimensionais / Estratégias de Resolução de Problemas

1. Escreva uma função em C que encontre o maior elemento armazenado numa tabela de inteiros. A função recebe como argumentos o nome e a dimensão da tabela. Devolve como resultado o maior valor existente na tabela.

2. Escreva uma função em C que encontre o maior elemento armazenado numa tabela de inteiros. A função recebe como argumentos o nome e a dimensão da tabela. Devolve como resultado a posição em que se encontra o maior valor existente na tabela. Se o maior valor surgir várias vezes na tabela, a função devolve a primeira posição em que ele ocorre.

3. Escreva uma função em C que conte quantas vezes surge o maior elemento numa tabela de inteiros. A função recebe como argumentos o nome e a dimensão da tabela. Devolve como resultado o número de vezes que o maior elemento surge na tabela.

4. Escreva uma função em C que descubra qual o elemento que surge mais vezes numa tabela de inteiros. A função recebe como argumentos o nome e a dimensão da tabela. Devolve como resultado o elemento mais comum da tabela (i.e., o que surge mais vezes). Se existirem vários elementos mais comuns, a função deve devolver o maior deles.

## Programação 2022/23

### LEI, LEI-PL, LEI-CE

5. Escreva uma função em C que verifique se um determinado valor se encontra numa tabela de inteiros. A tabela está ordenada de forma crescente e não contém elementos repetidos. A função recebe como argumentos o nome e a dimensão da tabela e o valor a pesquisar. Devolve 1 se o valor estiver na tabela, ou 0, no caso contrário.

6. Escreva uma função em C que receba uma tabela de inteiros com dimensão superior ou igual a 3 e escreva na consola a combinação de todos os conjuntos de 3 elementos da tabela cuja soma seja igual a um determinado valor. Este valor também é passado por argumento. O protótipo da função é o seguinte:

```
void calculaSoma(int tab[], int dim, int valor);
```

**Exemplo:** Dada a tabela {1,-2, 3, 4, -5, 6} com 6 elementos e o valor 8

A função deve apresentar na consola: 1 3 4        -2 4 6

7. Escreva uma função em C que calcule e devolva o desvio padrão de um conjunto de valores inteiros indicados pelo utilizador. A função recebe como argumento o número de valores a considerar. Pode assumir que não serão recebidos mais de 10 valores. O desvio padrão de um conjunto de N valores ( $x_1, x_2, \dots, x_N$ ) é calculado através da seguinte fórmula, em que  $\bar{X}$  representa a média dos valores:

$$StdDev = \sqrt{\frac{1}{N} \times \sum_{i=1}^N (\bar{X} - x_i)^2}$$

### B – Matrizes

8. Escreva uma função em C que preencha uma matriz  $N \times 3$  de inteiros da seguinte forma: em cada uma das linhas, a primeira coluna deve ficar com um inteiro entre 1 e 100 introduzido pelo utilizador, a segunda coluna com o quadrado deste valor e a terceira com o cubo. Durante o preenchimento deve garantir que não existem repetições nos valores da primeira coluna. A função recebe como argumentos o nome da matriz e o número de linhas que esta tem. Pode assumir que a matriz não tem mais de 100 linhas.

## Programação 2022/23

### LEI, LEI-PL, LEI-CE

9. Escreva uma função em C que apresente o conteúdo de uma matriz  $N \times 3$  de inteiros no monitor. A função recebe como argumentos o nome da matriz e o número de linhas que esta tem.

10. Um quadrado mágico é uma matriz  $3 \times 3$  contendo valores entre 1 e 9 (sem repetições), de tal forma que a soma dos elementos de qualquer uma das linhas, colunas e diagonais é sempre igual. A figura ao lado mostra um exemplo de um quadrado mágico.

6	1	8
7	5	3
2	9	4

Escreva uma função em C que verifique se uma matriz  $3 \times 3$  passada como argumento é um quadrado mágico. A função devolve 1 se isso se verificar ou 0 no caso contrário.

### C - Strings

11. Escreva uma função em C que receba uma *string* como argumento (i.e., um vetor contendo uma sequência de caracteres terminada com `'\0'`) e a escreva invertida no monitor.

Se a *string* passada por argumento for: **Hoje e Domingo!**

No monitor deve surgir: **!ognimoD e ejoH**

12. Escreva uma função em C que, após receber uma *string* como argumento, escreva cada uma das palavras que a constituem numa linha separada. Considere que as palavras estão separadas por um ou mais espaços em branco, podendo também existir espaços no início e fim da frase.

Se a *string* passada por argumento for: **Hoje e Domingo!**

No monitor deve surgir:

**Hoje**

**e**

**Domingo!**

13. Escreva uma função em C que receba 3 *strings* como argumento e faça o seguinte:

- Se o conteúdo das duas primeiras *strings* recebidas for exatamente igual, a função coloca a frase “Conteúdo Igual!” na terceira *string* e termina;
- Se as duas primeiras *strings* tiverem conteúdo diferente, mas forem do mesmo tamanho, então a função coloca a frase “Tamanho Igual!” na terceira *string* e termina;
- Caso contrário, a função coloca na terceira *string* o resultado da concatenação das duas primeiras. A *string* alfabeticamente menor deve surgir primeiro na concatenação.

## Programação 2022/23

### LEI, LEI-PL, LEI-CE

Ao responder a esta questão, pode assumir que a terceira *string* tem tamanho suficiente para a informação que é lá colocada.

**14.** Escreva uma função em C que verifique se uma *string* recebida como argumento armazena uma capicua. A função devolve 1 se isso se verificar, ou 0, no caso contrário.

**15.** Escreva uma função em C que conte o número de palavras que existem num *string* recebida como argumento. Considere que as palavras estão separadas por um ou mais espaços em branco. A função devolve o número de palavras contabilizado.

**16.** Escreva uma função em C que elimine os espaços em branco supérfluos que poderão existir entre palavras de uma *string* recebida como argumento.