

Programação 2023/24

LEI, LEI-PL, LEI-CE

Aula Laboratorial 2

Bibliografia:

K. N. King. *C programming: A Modern Approach* (2nd Edition). W. W. Norton: capítulos 8 e 13.

Código de apoio para a aula:

<https://gist.github.com/FranciscoBPereira>

Matrizes

Exercícios Obrigatórios

1. Escreva uma função em C que apresente no monitor o conteúdo de uma matriz de inteiros com dimensão $N \times 3$. A função recebe como argumentos o nome da matriz e o número de linhas que esta tem.
2. Escreva uma função em C que preencha uma matriz de inteiros com dimensão $N \times 3$ da seguinte forma: em cada uma das linhas, a posição da primeira coluna deve ficar com um valor inteiro introduzido pelo utilizador, a posição da segunda coluna deve armazenar o quadrado do valor especificado e a terceira posição deve armazenar o seu cubo. A função recebe como argumentos o nome da matriz e o número de linhas que esta tem.
3. Efetue as seguintes alterações na função implementada no ponto anterior:
 - Os valores indicados pelo utilizador devem pertencer ao intervalo $[1, 100]$.
 - Durante o preenchimento, a função deve garantir que não existem repetições nos valores que são armazenados na primeira coluna.

Ao responder a esta questão pode assumir que a matriz não tem mais de 100 linhas.

Exercícios Complementares

4. Escreva uma função em C que verifique se existem elementos repetidos numa matriz de inteiros com dimensão $N \times 3$. A função recebe como argumentos o nome da matriz e o número de linhas que esta tem. Devolve 1 se todos os elementos armazenados na matriz forem únicos, ou 0, no caso contrário.

Programação 2023/24

LEI, LEI-PL, LEI-CE

5. Um quadrado mágico é uma matriz 3×3 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 3×3 passada como argumento é um quadrado mágico. A função devolve 1 se isso se verificar ou 0 no caso contrário.

Strings

Exercícios Obrigatórios

6. 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

7. Escreva uma função em C que receba uma *string* como argumento (i.e., um vetor contendo uma sequência de caracteres terminada com `'\0'`) e escreva no monitor cada uma das palavras que a constituem numa linha separada. Considere que as palavras estão separadas por **um único espaço em branco** e que não existem espaços no início ou final da frase.

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

No monitor deve surgir:

Hoje

e

Domingo!

8. 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.

Ao responder a esta questão, pode assumir que a terceira *string* tem tamanho suficiente para a informação que é lá colocada. A resolução desta função deve recorrer a algumas das funções existentes na biblioteca standard *string.h*.

Programação 2023/24

LEI, LEI-PL, LEI-CE

Exercícios Complementares

9. Altere a função implementada no exercício 7, de modo que consiga lidar corretamente com vários espaços em branco entre as palavras e com espaços em branco que eventualmente existam no início e/ou final da frase.
10. 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.
11. Escreva uma função em C que elimine os espaços em branco supérfluos que poderão existir entre palavras de uma *string* recebida como argumento.

Trabalho Autónomo

A biblioteca standard *ctype.h* contém funções de classificação e alteração de caracteres. São pequenas funções, úteis para auxiliar a resolver problemas envolvendo tarefas de manipulação de caracteres. Consulte a lista de funções desta biblioteca (por exemplo, páginas 612-615 do livro *C Programming: A Modern Approach*) e resolva os desafios seguintes.

12. Escreva uma função em C que receba uma string como argumento e altere todas as vogais que surjam na frase para a sua correspondente maiúscula.
13. Escreva uma função em C que receba uma string como argumento e verifique se esta contém mais dígitos ou mais sinais de pontuação. A função devolve 1 se existirem mais dígitos, -1 se existirem mais sinais de pontuação, ou 0, em caso de empate.
14. Escreva uma função em C que contabilize quantos caracteres alfabéticos distintos é que existem numa string. A função recebe a string como argumento e devolve o valor contabilizado.