

Algoritmos com Matrizes

- 5.1) Escreva uma função ***void preenche_aleatorio(int x[LIN][COL], int nl, int nc)*** que preencha uma matriz de dimensão $nl \times nc$ elementos com números inteiros aleatórios com valores entre 1 e 100
- 5.2) Escreva um algoritmo que mostre no écran, devidamente formatada a matriz com $nl \times nc$ elementos ***void mostra_matriz(int x[LIN][COL], int nl, int nc)***
- 5.3) Construa o programa principal que lhe permita testar as duas funções anteriores (criando uma matriz com 5×10 elementos e mostrando o seu conteúdo no écran). Vá actualizando o programa principal com testes adequados às restantes alíneas do exercício. Utilize as instruções: `#define LIN 5` e `#define COL 10` para definir (e poder alterar facilmente o tamanho da matriz).
- 5.4) Escreva uma função que calcule a soma dos elementos uma matriz com $nl \times nc$ elementos inteiros ***int calcula_soma(int x[LIN][COL], int nl, int nc)***.
- 5.5) Escreva uma função que calcule a média dos elementos de uma matriz com $nl \times nc$ elementos inteiros ***double calcula_media(int x[LIN][COL], int nl, int nc)***.
- 5.6) Escreva uma função ***void preenche_borda(int x[LIN][COL], int nl, int nc)*** que preencha o rebordo da matriz (elementos na coluna 0 ou coluna $N-1$ ou na linha 0 ou linha $N-1$) com zeros.
- 5.7) Escreva uma função ***int pesquisa_matriz(int x[LIN][COL], int nl, int nc, int elem, int &co, int &li)*** que dado um vetor com n elementos inteiros permita pesquisar a existência de um dado valor (fornecido pelo utilizador) nesse vetor retornando nas variáveis `co` e `li` a posição (`co/li`) respetiva do elemento no vetor. A função deve retornar 1 caso encontre o elemento ou -1 caso esse elemento não exista na matriz.
- 5.8) Escreva uma função que dadas duas matrizes com as mesmas dimensões some os seus elementos, um a um, gerando uma nova matriz ***void soma_matrizes(int x1[LIN][COL], int x2[LIN][COL], int x3[LIN][COL], int nl, int nc)***
- 5.9) Escreva uma função ***int ocorrencias(int x[LIN][COL], int nl, int nc, int elem)*** que dada uma matriz com $nl \times nc$ elementos e um número inteiro, calcule o número de ocorrências desse número na matriz.
- 5.10) Escreva uma função ***void substitui_elemento(int x[LIN][COL], int nl, int nc, int elem1, int elem2)*** que dada uma matriz com $nl \times nc$ elementos substitua todas as ocorrências na matriz no número `elem1` por `elem2`.
- 5.11) Escreva uma função que dada uma matriz quadrada com $nl \times nc$ elementos calcule a sua transposta e coloque numa matriz `x2`: ***void transposta_matriz(int x[LIN][COL], int x2[COL][LIN], int nl, int nc)***
- 5.12) Escreva uma função semelhante à do exercício anterior mas que calcule a transposta de uma matriz quadrada $n \times n$ na própria matriz: ***void transposta_matriz_quad(int x[LIN][COL], int n)***
- 5.13) Escreva uma função que calcule a multiplicação de duas matrizes: ***void multiplica_matriz(int x[LIN][COL], int x2[COL][LIN], int x3[LIN][LIN], int nl, int nc)***