



### Lista 10 - Vetores e Matrizes com ponteiros

**Instruções:** Lembre-se de testar suas implementações. Em cada exercício que envolver alocação dinâmica, certifique-se de liberar a memória ao final.

1. Considere um vetor de 10 números inteiros positivos maiores que zero e um único número  $X$  inteiro, também positivo e maior que zero. Faça um programa para:
  - (a) ler pelo teclado o vetor;
  - (b) ler pelo teclado o número  $X$ ;
  - (c) encontrar e imprimir o par de posições consecutivas cujas componentes possuem a maior distância entre elas;
  - (d) verificar se o vetor está em ordem crescente, decrescente ou não ordenado;
  - (e) dizer quantos números no vetor são maiores que  $X$ , menores que  $X$  e iguais a  $X$ .
2. Uma agência de publicidade pediu à agência de modelos Luz & Beleza para encontrar um ator que tenha idade entre 18 e 20 anos para participar de uma campanha publicitária. Foram inscritas 20 candidatos e, ao se inscreverem, forneceram nome e idade. Tais informações foram armazenadas em 2 vetores distintos. O funcionário responsável pelas inscrições se distraiu e inscreveu um candidato fora dos limites de idade. Faça um programa para retirar do vetor de nomes o candidato que não está nas especificações da agência. Imprima o vetor que contém os nomes dos candidatos aptas a concorrer a uma vaga para a campanha.
3. Desenvolva uma função que receba como parâmetro os ponteiros de dois vetores de 5 posições. O procedimento deverá imprimir na tela os valores contidos nos dois vetores de forma crescente (Utilize ponteiros)

Exemplo:

Vetor 1 = 2, 5, 9, 8, 3

Vetor 2 = 7, 4, 1, 10, 6

Saída: 1, 2, 3, 4, 5, 6, 7, 8, 9, 10

4. Escreva uma função que recebe uma matriz representada como um ponteiro para o primeiro elemento, o número de linhas e colunas, e retorna o maior valor encontrado na matriz. A matriz deve ser percorrida usando aritmética de ponteiros.
5. Implemente uma função que recebe uma matriz quadrada ( $N \times N$ ) como ponteiro para o primeiro elemento e sua ordem  $N$ , e transpõe a matriz (troca linhas por colunas) diretamente na memória, usando ponteiros.

## 6. Criando uma matriz

Escreva um programa em C que declare uma matriz de inteiros com 2 linhas e 3 colunas. O programa deve solicitar que o usuário insira os valores inteiros para preencher a matriz.

Em seguida, o programa deve:

- Mostrar a matriz original.
- Criar uma matriz chamada `matrizTransposta` e exibir a matriz transposta (ou seja, uma matriz  $3 \times 2$ ).
- Calcular e exibir a soma de todos os elementos da matriz.

### Exemplo de saída:

```
Digite o valor para a posição [0][0]: 1
Digite o valor para a posição [0][1]: 2
Digite o valor para a posição [0][2]: 3
Digite o valor para a posição [1][0]: 4
Digite o valor para a posição [1][1]: 5
Digite o valor para a posição [1][2]: 6
```

Matriz original:

```
1  2  3
4  5  6
```

Matriz transposta:

```
1  4
2  5
3  6
```

A soma dos elementos da matriz é: 21

## 7. Usando matrizes em funções e procedimentos

- a) Escreva um programa principal em C que crie e armazene uma matriz quadrada de tamanho  $n \times n$ , onde  $n$  deve ser lido do teclado e fornecido pelo usuário.
- b) Chame uma função `preencheMatriz` que preencha a matriz criada no programa principal com valores aleatórios entre 1 e 100.
- c) Chame um procedimento `somaDiagonalPrincipal` que calcule e exiba a soma dos elementos da diagonal principal da matriz. Esse procedimento deve receber uma **cópia da matriz**. Note que essa abordagem não é eficiente em termos de memória, mas será utilizada para fins didáticos.
- d) Chame um procedimento `somaAbaixoDiagonal` que calcule e exiba a soma dos elementos abaixo da diagonal principal da matriz. Esse procedimento deve receber o **ponteiro que aponta para a matriz** e o tamanho da matriz, tornando-o mais eficiente, pois economiza memória ao evitar a criação de uma cópia desnecessária da matriz.
- e) Implemente as funções e procedimentos utilizados no programa principal:
  - `preencheMatriz`
  - `somaDiagonalPrincipal`
  - `somaAbaixoDiagonal`