



Lista 09 - Vetores e Matrizes

1. Crie um programa que preencha um vetor com 20 valores inteiros entre 1 e 100, gerados automaticamente. Em seguida, implemente uma função que receba esse vetor e um número a ser buscado. A função deve imprimir todas as posições onde o número ocorre.
2. Implemente uma função que preencha um vetor de tamanho 100 com os 100 primeiros números naturais que não são múltiplos de 6 e que não terminam com 6. O vetor preenchido deve ser retornado e impresso na `main`.
3. Escreva um programa que contenha três funções:
 - uma função que preencha um vetor de 5 elementos inteiros com valores fornecidos pelo usuário;
 - uma função que encontre e retorne a posição do menor elemento do vetor;
 - uma função que troque o menor elemento com o primeiro e imprima o vetor resultante.
4. Crie uma função que receba dois vetores preenchidos A e B de 10 números inteiros cada e calcule um terceiro vetor C tal que $C = A - B$. A função deve retornar o vetor C e a `main` deve exibi-lo.
5. Implemente dois procedimentos:
 - `preencheValores`, que preenche um vetor X de 10 elementos com valores inteiros fornecidos pelo usuário;
 - `copiaNegativos`, que recebe esse vetor preenchido e copia os valores negativos para outro vetor de até 10 elementos. Os valores copiados devem ser organizados sem espaços e o vetor deve conter o número 0 após o último negativo.

A função `main` deve chamar esses procedimentos e imprimir os valores negativos, desconsiderando os zeros.

6. Manipulando uma matriz simples

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:

- a) Mostrar a matriz original no formato de tabela.
- b) Criar uma nova matriz chamada `matrizTransposta` e exibir a transposta (ou seja, uma matriz 3×2).
Dica: a transposta de uma matriz transforma linhas em colunas.
- c) Calcular e exibir a soma de todos os elementos da matriz original.

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. Funções e passagem de matrizes

Escreva um programa principal em C que realize as seguintes etapas:

- Crie e armazene uma matriz quadrada de tamanho $n \times n$, onde n deve ser fornecido pelo usuário. Pode-se definir um tamanho máximo (`#define MAX 100`) e trabalhar com a parte útil da matriz.
- Chame uma função `preencheMatriz` que preencha a matriz com valores aleatórios entre 1 e 100. A função deve receber a matriz e seu tamanho.
- Chame uma função `somaDiagonalPrincipal` que receba a matriz e seu tamanho, no formato `int matriz[][MAX]`, `int n`, e calcule e exiba a soma dos elementos da diagonal principal.
- Chame uma função `somaAbaixoDiagonal` que receba a matriz e calcule e exiba a soma dos elementos abaixo da diagonal principal. A função pode receber a matriz no formato `int *matriz` considerando acesso linearizado, ou `int matriz[][MAX]` para acesso bidimensional.
- Implemente as funções utilizadas no programa principal:
 - `void preencheMatriz(int matriz[][MAX], int n);`
 - `void somaDiagonalPrincipal(int matriz[][MAX], int n);`
 - `void somaAbaixoDiagonal(int *matriz, int n);`

Dica: embora a sintaxe pareça indicar passagem por valor, em C as matrizes são sempre passadas por referência.