

## **Lista de Exercícios 6**

### **Informações sobre cópias**

- As questões são individuais. Em caso de cópias de trabalho a pontuação será zero para os autores originais e copiadores. Não serão aceitas justificativas como: “Fizemos o trabalho juntos, por isso estão idênticos”.

### **Parte A – Matrizes, Alocação Dinâmica e Estruturas - Exercícios para serem entregues no Canvas**

*Para as questões 1 a 3 considere a declaração das seguintes constantes globais:*

**const int MAX\_LIN=10;**

**const int MAX\_COL=5;**

1. Construa uma função que calcule e retorne a soma dos valores de uma linha  $i$  de uma matriz.

A função deverá ter os seguintes argumentos:

- A passagem por referência da matriz;
- O número da linha a ser calculada.

2. Considere uma matriz quadrada MAX\_LIN x MAX\_LIN. Construa uma função que calcule a soma dos valores posicionados acima da diagonal principal.

3. Considere uma matriz quadrada MAX\_LIN x MAX\_LIN. Construa um procedimento que escreva os elementos da diagonal secundária.

4. Escreva um programa que preencha dois vetores de inteiros A e B com 4 números ordenados (em cada vetor). Em seguida, implemente um procedimento *Merge*, que receba os vetores A e B e retorne um vetor C ordenado, combinando os resultados dos vetores A e B. Imprima o vetor C na função *main*. A, B e C devem ser alocados usando malloc ou calloc.

Obs. 1: O vetor C deve ser instanciado (“alocado”) no procedimento *Merge*.

Obs. 2: Após a impressão dos valores, o comando *free* deve ser executado para liberação de memória dos vetores A, B e C.

Ex.:  $A = \{5, 8, 12, 16\}$ ,  $B = \{3, 7, 14, 15\}$ ,  $C = \{3, 5, 7, 8, 12, 14, 15, 16\}$

5. Desenvolva um programa que instancie um vetor de inteiros A utilizando *malloc*. O programa deve ler um valor *k*, onde *k* é positivo e menor que o tamanho do vetor (rejeite entradas inválidas). Implemente um procedimento que receba como parâmetros o vetor A, o tamanho do vetor  $|A|$  e o valor *k*. Retorne o *k*-ésimo menor elemento do vetor.

Ex.:  $A = \{3, 5, 7, 8, 12, 14, 15, 16\}$ ,  $k = 3$ , *retorno* = 7.

6. Escreva um programa que leia do teclado um valor *n*. Instancie, usando o comando *malloc*, uma matriz de números em ponto flutuante  $M_{n \times n}$ . Encontre (a) o maior elemento, (b) o menor elemento, (c) a média dos elementos da matriz.

*Para as questões de 7 a 10 considere a criação de uma struct chamada Veiculo que contenha os seguintes campos:*

- id (número de frota) - int
- quilometragem (quilometragem total do veículo) – float

7. Escreva um programa que preencha um vetor de 20 veículos usando a struct para armazenar as identificações e quilometragem dos veículos de uma empresa de transporte. Depois, o programa deve solicitar ao usuário um número de identificação para pesquisar no vetor e mostrar todas as posições onde essa identificação aparece.

8. Escreva um programa que utilize alocação dinâmica para criar um vetor de 100 números de série de modelos de veículos cujos números são múltiplos de 4, mas que não terminam com o dígito 4. Certifique-se de que o vetor tenha exatamente 100 modelos e, ao final, libere a memória alocada.

9. Escreva um algoritmo que crie um vetor para armazenar 5 veículos (struct) e ofereça ao usuário funcionalidades para:

- Procurar o veículo com a maior quilometragem.
- Troque o veículo com a maior quilometragem com o da primeira posição.
- Imprima a lista de quilometragens após a troca.

Cada uma dessas etapas deve ser implementada em funções ou procedimentos separados.

10. Amplie a struct Veículo adicionando o campo capacidade, que irá representar o número de passageiros. Implemente um procedimento preencheCapacidade que preencha um vetor com 10 veículos, armazenando suas capacidades. Em seguida, faça um procedimento copiaGrandesCapacidades que copie veículos com capacidade acima de 30 passageiros para um novo vetor de até 10 elementos. Ao final, mostre o vetor resultante.

A seguir a sugestão de assinatura dos procedimentos que deverão ser criados.

```
void preencheCapacidade(Veiculo vetor[], int length);  
void copiaGrandesCapacidades(Veiculo vetor[], int length, Veiculo vetorGrandes[]);
```

## Parte B – Exercícios para Treino – Revisão (não precisam ser entregues)

B1. Escreva um programa que leia um nome e o imprima de trás para frente.

B2. Crie um programa que receba uma string representando a placa de um veículo e mostre o número de letras e dígitos numéricos que ela contém. Ignore espaços e caracteres especiais.

B3. Desenvolva um programa que preencha dois vetores de inteiros A e B com 5 valores inteiros aleatórios entre 0 e 10. Crie uma função que gere um vetor C, correspondente a união dos vetores A e B ( $A \cup B$ ). Obs: C não deve conter valores repetidos. Utilize o comando *malloc* para instanciar os vetores de inteiros com 5 elementos cada. Caso necessário, utilize o comando *realloc* para redimensionar o vetor C. O vetor C não deve conter posições não preenchidas no vetor, correspondendo exatamente a  $|A \cup B|$ .

Para os próximos exercícios construa um procedimento que preenche uma matriz de números reais de tamanho M x N. Os valores de M e N poderão ser informados pelo usuário ou definidos com constantes no programa. O usuário poderá escolher se a matriz deverá ser preenchida por ele ou se será preenchida utilizando números aleatórios. Depois, faça o que se pede nas questões a seguir.

B4. Construa uma função que calcule a amplitude em uma matriz: a diferença entre o maior e o menor valor nela presente.

B5. Construa um algoritmo que solicite ao usuário que digite um número real k, calcule o mostre quantas vezes o número k ocorre dentro da matriz.

B6. Construa um procedimento que cria uma segunda matriz de números reais de tamanho M x N e solicita ao usuário o preenchimento dessa segunda matriz. A seguir exiba para o usuário a terceira matriz com o resultado da soma de ambas.

B7. Construa uma função que recebe uma matriz e verifica se a matriz é triangular, DICA: existe dois casos para a matriz ser triangular:

- Os valores acima da diagonal principal deve ser todos zeros.
- Os valores abaixo da diagonal principal deve ser todos zeros.

Exemplos:

```
12 14 8
0  4 65 É uma matriz triangular superior;
0  0 8
```

```
9  0 0
-9 5 0 É uma matriz triangular inferior;
89 13 5
```

```
7 0 0
0 8 0 É uma matriz triangular superior. e inferior.
0 0 9
```

-A função deverá retornar um booleano.

B8. Escreva um procedimento que leia dois inteiros *linha* e *coluna* que são índices de linha e coluna respectivamente de uma matriz. Seu procedimento deverá imprimir a soma total dos elementos da linha *linha* com os elementos da coluna *coluna* da matriz.

B9. Construa uma função para verificar se uma matriz é quadrática, ou seja, M deve ser igual a N. A função deve ser booleana.

B10. Construa uma função que calcule o determinante de uma matriz. Primeiramente, deve-se utilizar a função construída no item anterior e se caso for quadrática a função deve calcular o seu determinante.

Dica:

$$A = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$$
$$\det(A) = (a * d) - (b * c)$$

B11. Construa um algoritmo que determina e imprime a transposta da matriz.

B12. Construa uma função que verifique se uma matriz é simétrica.

Dica:

$$A = A^T$$

$$\begin{bmatrix} 1 & 2 & 3 \\ 2 & 4 & 5 \\ 3 & 5 & 8 \end{bmatrix} = \begin{bmatrix} 1 & 2 & 3 \\ 2 & 4 & 5 \\ 3 & 5 & 8 \end{bmatrix}^T$$

B13. Construa uma função que recebe uma matriz, calcule e retorne quantos números são maiores que o valor médio da matriz.

B14. Construa uma função que recebe uma matriz e verifica se a primeira linha de uma matriz de reais é igual à sua diagonal principal. Trate as situações que impeçam a execução dessa verificação. A função deverá retornar um booleano.

B15. Crie uma struct chamada Transporte que contenha os seguintes campos:

- `lotacao_maxima` (lotação máxima do veículo)
- `ocupacao_atual` (ocupação atual do veículo)

Escreva um programa que preenche vetor do tipo Transporte, com as lotações máximas e ocupações atuais de 10 veículos. Na sequência, o programa deve calcular a diferença (lotação máxima - ocupação atual) para cada veículo e armazenar em um novo vetor de inteiros. Mostre na tela os valores deste novo vetor.