



Aula Prática P-11

- * Todos os exercícios que envolvem programas devem ser resolvidos através de programas em C/C++.
 - * A entrega será feita até às 23h55 do dia da aula prática no Moodle, sem zipar (entregue apenas o código fonte)
 - * Inclua seu número de matrícula, nome e turma em um comentário no início de cada arquivo com código fonte.
 - * Você só pode utilizar conhecimento prévios à aula para resolver o exercício. Caso use uma matéria que ainda não foi dada sua nota será penalizada.
 - * Códigos que não compilam serão zerados.
-

Questão 01

Escreva um programa que leia os elementos de um vetor (*vet1*) de inteiros, dado o seu tamanho n .

Em seguida, o programa irá calcular a média desses números e gerar dois outros vetores. Um com os valores abaixo da média e outro com os valores iguais ou superiores a média.

Por exemplo:

Para $n = 5$ e $\text{vet1} = \{5, 2, 8, 9, 1\}$ a média é igual a 5.

Portanto, teremos $\text{vet2} = \{1, 2\}$ e $\text{vet3} = \{5, 8, 9\}$.

Imprima a quantidade de valores em cada um dos vetores gerados e seus respectivos valores.

```
1 Digite o valor de n: 5
2 Digite os valores do vetor: 5 2 8 9 1
3 A média é: 5
4 Vetor com os valores abaixo da média: [ 1 2 ]
5 Vetor com os valores iguais ou acima da média: [ 5 8 9 ]
```

Observação: Em todos os exercícios desta lista, utilize alocação dinâmica para criar os vetores. Isto quer dizer que:

- Você não pode desperdiçar memória com os vetores vazios. Os vetores devem ser alocados de acordo com a quantidade de valores que irá armazenar.
- Você não pode implementar soluções que imponham limite de tamanho suportado ao usuário.

Questão 02

Escreva um programa para ler uma matriz ($A_{m \times n}$) e calcular a sua matriz transposta (A^t).

Para determinar a transposta A^t de uma matriz A , basta reescrevê-la de forma que suas linhas e colunas troquem de posições ordenadamente, isto é, a primeira linha é reescrita como a primeira coluna, a segunda linha é reescrita como a segunda coluna e assim por diante, até que se termine de reescrever todas as linhas na forma de coluna, por exemplo:

$$A = \begin{vmatrix} 1 & 3 \\ 2 & 6 \\ -5 & 0 \end{vmatrix} \Rightarrow A^t = \begin{vmatrix} 1 & 2 & -5 \\ 3 & 6 & 0 \end{vmatrix}$$

O programa deve ler os índices m , n e matriz A e imprimir a matriz original e transposta. As matrizes devem ser armazenadas utilizando-se alocação dinâmica.

Utilize as seguintes funções:

```

1 // Recebe uma matriz e seus índices e retorna a matriz transposta
2 double** cria_transposta (double **A, int m, int n);
3
4 // Aloca espaço para uma matriz m x n. Se não for possível criar a matriz, retorna NULL
5 double** cria_matriz(int m, int n);
6
7 // Libera o espaço utilizado por uma matriz.
8 void desaloca_matriz(double** A, int m)

```

Questão 03

Crie uma função que multiplica duas matrizes $A_{n \times m}$ e $B_{p \times q}$. Utilize o protótipo a seguir:

```

1 int multiplica_matrizes(int ***R, int **A, int n, int m, int **B, int p, int q);

```

A função deve alocar memória para a matriz de resultados R e armazenar o endereço desta nova matriz no endereço `int ***R`, recebido como parâmetro. Note que a memória para a matriz R é alocada pela função.

Esta nova matriz R conterá o resultado da multiplicação, tal que $R = A \times B$. Por fim, a função deve retornar 1 em caso de sucesso e 0 caso seja impossível multiplicar as matrizes.

Em seguida, crie um programa que lê as matrizes A e B do usuário e imprime o resultado de $A \times B$.

Exemplos de execução (dados digitados pelo usuário em azul):

```

1 Digite os tamanhos da matriz A: 2 3
2 Digite os dados da matriz A:
3 2 5 9
4 3 6 8
5
6 Digite os tamanhos da matriz B: 3 2
7 Digite os dados da matriz B:
8 2 7
9 4 3
10 5 2
11
12 Resultado de A x B:
13 69 47
14 70 55

```

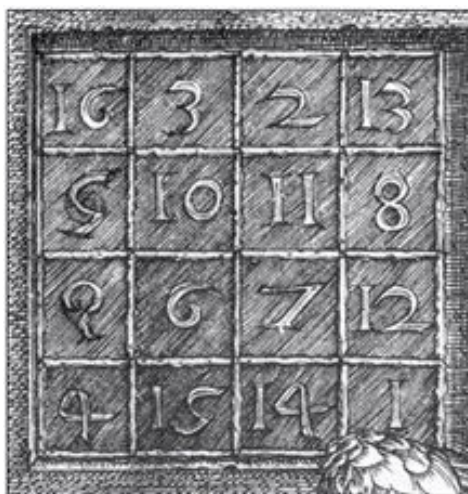
```

1 Digite os tamanhos da matriz A: 1 2
2 Digite os dados da matriz A:
3 2 5
4
5 Digite os tamanhos da matriz B: 3 2
6 Digite os dados da matriz B:
7 2 7
8 4 3
9 5 2
10
11 Não é possível multiplicar as matrizes A e B.

```

Questão 04

Quadrado Mágico. Um Quadrado Mágico é uma tabela quadrada de lado n , onde a soma dos números das linhas, das colunas e das diagonais principal e secundária é constante, sendo que nenhum destes números se repete.



(Fonte: Wikipédia/A Melancolia, de Albrecht Dürer, 1514).

Crie uma **função** para verificar (não confunda com resolver) se uma matriz realmente é um quadrado mágico.

Utilize alocação dinâmica para criar a matriz e não se esqueça de liberar a memória ao final do programa.

Crie também a função **main**, responsável por gerar saída semelhante ao exemplo abaixo (note que os dados digitados pelo usuário estão destacados em azul):

```

1 Digite o valor de n: 4
2 Digite os valores da matriz n x n:
3 16 3 2 13
4 5 10 11 8
5 9 6 7 12
6 4 15 14 1
7
8 Esta matriz é um quadrado mágico!

```