Exercícios – Lista 2

ATENÇÃO	**a)** Todas as matrizes e vetores utilizados nos exercícios têm que ser declarados na função *main*. **b)** Não é permitido utilizar variáveis globais. O **#define** pode ser utilizado.

1) Faça um algoritmo para ler pelo teclado uma matriz 4 x 3 com valores reais e imprimir o maior elemento da matriz e a sua localização (linha e coluna). Crie e utilize dois **procedimentos**: um para a leitura da matriz e outro para imprimir as informações.

2) Faça um algoritmo para ler pelo teclado uma matriz 3 x 5 com valores reais e imprimir quantos valores maiores do que 20 ela possui. Crie e utilize um **procedimento** para a leitura da matriz e uma **função** para fazer o cálculo. A impressão da informação (quantos valores maiores do que 20) deve ser na função *main*.

3) Faça um algoritmo para ler pelo teclado uma matriz 4 x 4 com valores inteiros e também um número inteiro *x*. Ao final, o algoritmo tem que imprimir se o número *x* pertence ou não à matriz. O número *x* tem que ser lido na função *main*. Crie e utilize um **procedimento** para a leitura da matriz e uma **função** para realizar a busca. A impressão da informação (se *x* está presente ou não na matriz) deve ser na função *main*.

4) Faça um algoritmo para ler pelo teclado duas matrizes de reais 5 × 3 e imprimir a soma dessas matrizes. A matriz que irá receber a soma das outras duas deve ser criada na função *main*. Crie e utilize três **procedimentos**: um para a leitura das matrizes, outro para calcular a soma e um terceiro para imprimir a matriz com o resultado da soma.

5) Faça um algoritmo para ler pelo teclado duas matrizes de reais 3 × 4 e imprimir a subtração dessas matrizes. A matriz que irá receber a subtração das outras duas deve ser criada na função *main*. Crie e utilize três **procedimentos**: um para a leitura das matrizes, outro para calcular a subtração e um terceiro para imprimir a matriz com o resultado da subtração.

6) Faça um algoritmo para imprimir o resultado da multiplicação de um número lido pelo teclado (lido na função *main*) por uma matriz de inteiros 2 × 8 gerada aleatoriamente com números de 0 até 29. A matriz que irá receber o resultado da multiplicação do número pela matriz deve ser criada na função *main*. Crie e utilize três **procedimentos**: um para a geração da matriz, outro para calcular a multiplicação e um terceiro para imprimir a matriz com o resultado da multiplicação.

7) Faça um algoritmo para verificar se uma matriz quadrada 6 x 6 gerada aleatoriamente com números de 1 até 50 é **simétrica**. Crie e utilize um **procedimento** para a geração da matriz e uma **função** para a verificação. De acordo com o retorno da função deve-se imprimir na função *main*: MATRIZ SIMETRICA ou MATRIZ NAO SIMETRICA.

8) Faça um algoritmo para calcular e imprimir a matriz **transposta** de uma matriz 7 x 3 gerada aleatoriamente com números de 1 até 25. A matriz que irá receber o resultado da transposta deve ser criada na função *main*. Crie e utilize três **procedimentos**: um para a geração da matriz, outro para calcular a transposta e um terceiro para imprimir a matriz transposta.

9) Faça um algoritmo para verificar se uma dada matriz quadrada 10 x 10 gerada aleatoriamente com números de 0 até 19 é uma **matriz identidade**. Em uma matriz identidade, todos os elementos da **diagonal principal** são iguais a 1 e os demais são iguais a 0. Crie e utilize um **procedimento** para a geração da matriz e uma **função** para a verificação. De acordo com o retorno da função deve-se imprimir na função *main*: MATRIZ IDENTIDADE ou MATRIZ NÃO E IDENTIDADE.

10) Faça um algoritmo que, dadas duas matrizes quadradas A e B de tamanho 6 x 6 geradas aleatoriamente com números de 1 até 10, verifique se B é a **inversa** de A, isto é, se B é igual a A^{-1}. Se B for a inversa, a multiplicação de A por B resulta em uma matriz identidade. Crie e utilize uma **função** para verificar se a matriz resultante de A x B é uma matriz identidade e dois **procedimentos**: um para a geração das matrizes e outro para realizar a multiplicação delas. De acordo com o retorno da função deve-se imprimir na função *main*: B E INVERSA DE A ou B NAO E INVERSA DE A.

11) Faça um algoritmo que verifique se uma dada matriz quadrada 8 x 8 gerada aleatoriamente com números de 0 até 14 é uma matriz **triangular inferior**. Em uma matriz triangular inferior, todos os elementos acima da diagonal principal são iguais a 0. Os elementos da diagonal principal ou abaixo dela podem assumir valores quaisquer. Crie e utilize um **procedimento** para a geração da matriz e uma **função** para a verificação. De acordo com o retorno da função deve-se imprimir na função *main*: MATRIZ TRIANGULAR INFERIOR ou MATRIZ NAO TRIANGULAR INFERIOR.

12) Faça um algoritmo para gerar e imprimir uma matriz quadrada A de tamanho 10 x 10, onde seus elementos são:

$$A_{ij} = \begin{cases} 2.i + 7.j + 2, \text{ se } i < j; \\ 3.i^2 - 1, \text{ se } i = j; \\ 4.i^3 - 5.j^2 + 1, \text{ se } i > j. \end{cases}$$

Crie e utilize dois **procedimentos**: um para gerar a matriz e outro para imprimi-la.

13) Faça um algoritmo para calcular a **soma** dos elementos que estão **acima** da diagonal principal de uma matriz quadrada 5 x 5 gerada aleatoriamente com números de 1 até 40. Crie e utilize um **procedimento** para a geração da matriz e uma **função** para realizar a soma. O resultado da soma tem que ser impresso na função *main*.

14) Faça um algoritmo para calcular a **multiplicação** dos elementos que estão **abaixo** da diagonal principal de uma matriz quadrada 6 x 6 gerada aleatoriamente com números de 1 até 10. Crie e utilize um **procedimento** para a geração da matriz e uma **função** para realizar a multiplicação. O resultado da multiplicação tem que ser impresso na função *main*.

15) Faça um algoritmo para calcular a **soma** dos elementos que estão em uma coluna *x* de uma matriz 10 x 4 gerada aleatoriamente com números de 1 até 15. A coluna a ser somada (coluna *x*) deve ser lida pelo teclado na função *main*. Crie e utilize um **procedimento** para a geração da matriz e uma **função** para realizar a soma. O resultado da soma tem que ser impresso na função *main*.

16) Faça um algoritmo que leia uma matriz de ordem 4 com valores reais e imprima a soma dos valores contidos em sua diagonal secundária. Crie e utilize um **procedimento** para ler a matriz e uma **função** para realizar a soma. O resultado da soma tem que ser impresso na função *main*.

17) Na teoria dos sistemas, define-se como elemento ***minimax*** de uma matriz o menor elemento de uma linha onde se encontra o maior elemento da matriz. Faça um algoritmo para gerar uma matriz quadrada 10 x 10 com números aleatórios de 1 até 50 e imprimir o seu elemento *minimax*. Crie e utilize um **procedimento** para a geração da matriz e uma **função** para retornar o elemento *minimax*. A impressão desse elemento tem que ser na função *main*.

18) Faça um algoritmo que leia uma matriz 3 x 3 de números inteiros. Depois, calcule e armazene na posição *i* de um vetor de tamanho 3 a soma dos valores da coluna *i* da matriz ($0 \leq i \leq 2$), imprimindo em seguida o vetor. Crie e utilize dois **procedimentos**: um para ler a matriz e outro para preencher e imprimir o vetor. O vetor também deve ser declarado na função *main*. Ex.:

5	9	6
12	4	8
18	0	2

35	13	16

19) Considere uma matriz quadrada de ordem 6 gerada aleatoriamente com números de 1 até 30. Faça um algoritmo que calcule e imprima a soma dos elementos dessa matriz que **não** pertençam à diagonal principal nem à diagonal secundária. Crie e utilize um **procedimento** para a geração da matriz e uma **função** para retornar a soma. A impressão da soma tem que ser na função *main*.

20) Faça um algoritmo que leia uma matriz 10 x 3, onde as linhas representam os alunos e as colunas as 3 provas de cada aluno. Em seguida, imprima a maior nota e a menor nota da prova 1, da prova 2 e da prova 3. Crie e utilize dois **procedimentos**: um para ler a matriz e outro para imprimir as informações.

21) Faça um algoritmo que leia o nome completo (com no máximo 20 letras e considerando espaços) de 50 pessoas. Após a leitura de todos os nomes, o algoritmo deve imprimir os mesmos em ordem inversa de entrada. Crie e utilize dois **procedimentos**: um para ler os nomes e outro para imprimir os mesmos em ordem inversa de entrada.

22) Faça um algoritmo que leia o nome completo (com no máximo 20 letras e considerando espaços) de 30 pessoas. Após a leitura de todos os nomes, o algoritmo tem que ler (pelo teclado na função *main*) um nome completo e informar se este pertence ou não à lista dos nomes digitados. Crie e utilize dois **procedimentos**: um para ler os nomes e outro para informar se o nome digitado pertence ou não à lista.
Obs.: Considere que todos os nomes digitados seguem o mesmo padrão para as letras (minúsculas e/ou maiúsculas).

23) Faça um algoritmo que simule uma pilha de, no máximo, 60 livros. Um livro somente poderá ser incluído e removido no topo da lista. O nome da cada livro pode ter, no máximo, 15 letras (considerando os espaços). Crie um menu para o usuário selecionar de cada vez uma das seguintes funcionalidades: (1) Inserir Livro; (2) Remover Livro; e (3) Listar Livros. Crie e utilize um **procedimento** para cada funcionalidade. Caso o usuário não queira mais inserir, remover e/ou listar os livros inseridos, o mesmo deve digitar a opção: (4) Sair.

24) Faça um algoritmo para, primeiro, ler pelo teclado todos os 30 números naturais (maiores do que 1) da matriz M_{4x5} e, em seguida, imprimir na tela a quantidade de números primos que M possui. Crie e utilize dois **procedimentos**: um para ler a matriz e outro para calcular e imprimir a quantidade de números primos. Crie e utilize também uma **função** que retorna 1, se um número natural for primo, ou retorna 0, caso contrário.

Um **grafo** é uma estrutura matemática/computacional muito utilizada em variadas aplicações para representar um **conjunto de objetos** (simbolizado por **vértices**) e as **relações existentes entre eles** (simbolizadas por **arestas** ou **arcos** – dependendo do grafo). Diversos sistemas complexos como redes sociais, redes biológicas, redes de tráfego aéreo, dentre tantas outras aplicações, são modeladas matematicamente/computacionalmente por grafos.

A Figura 1 apresenta um exemplo de um grafo simples não-direcionado G = (V, E), onde o conjunto de vértices V contém 7 vértices e o conjunto de arestas E contém 11 arestas:

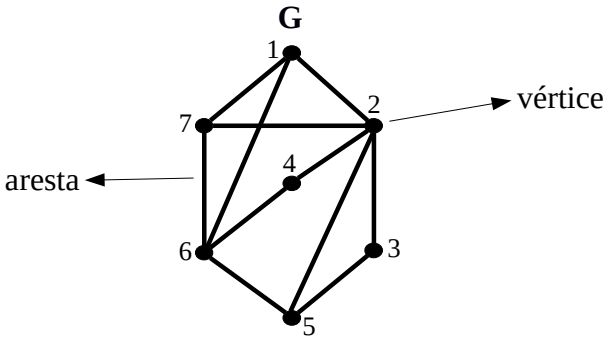

Figura 1: Exemplo de um grafo simples não-direcionado G com 7 vértices e 11 arestas.

O **número de vértices** de um grafo é representado por *n* e o **número de arestas** (ou arcos, dependendo do grafo) por *m*. No exemplo do grafo G da Figura 1, *n* = 7 e *m* = 11.

Um grafo G pode ser modelado no computador por uma matriz quadrada de ordem *n*, chamada **Matriz de Adjacência** de G (representada por A(G)), onde as linhas e as colunas de A(G) representam os vértices de G e A_{ij} = 1, se o vértice *i* está ligado ao vértice *j* (ou seja, se a aresta $(i, j) \in E$), e A_{ij} = 0, caso contrário. A matriz abaixo é a Matriz de Adjacência A(G) do grafo G da Figura 1:

A(G) =

0	1	0	0	0	1	1
1	0	1	1	1	0	1
0	1	0	0	1	0	0
0	1	0	0	0	1	0
0	1	1	0	0	1	0
1	0	0	1	1	0	1
1	1	0	0	0	1	0

25) O **grau** de um vértice *v*, representado por *d*(*v*), em um grafo simples não-orientado é o número de arestas incidentes (ligadas) a ele. Por exemplo, o grau do vértice 5 do grafo G da Figura 1 é igual a 3 (*d*(5) = 3). Faça um algoritmo para ler pelo teclado a matriz de adjacência A(G) de um grafo simples não-direcionado G com 10 vértices e imprimir na tela o grau de cada vértice. Crie e utilize dois **procedimentos**: um para ler a matriz pelo teclado e outro para imprimir o que se pede.

26) Um grafo simples não-direcionado G é ***k*-regular** se todos os vértices têm grau *k*. Faça um algoritmo para ler pelo teclado a matriz de adjacência A(G) de um grafo simples não-direcionado G com 15 vértices e imprimir na tela se G é ou não um grafo *k*-regular. Em caso verdadeiro, imprima também o valor de *k*. Crie e utilize dois **procedimentos**: um para ler a matriz pelo teclado e outro para imprimir o que se pede.

27) Faça um algoritmo para ler pelo teclado uma matriz de um grafo G com 20 vértices e imprimir na tela se a matriz é ou não a matriz de adjacência A(G) de um **grafo simples não-direcionado** G. Crie e utilize um **procedimento** para ler a matriz pelo teclado e uma **função** para retornar 1, se a matriz for a matriz de adjacência de um grafo simples não-direcionado, ou retornar 0, caso contrário.

Obs.: A matriz de adjacência A(G) de um **grafo simples não-direcionado** G tem as seguintes características: a diagonal igual a zero; somente uma aresta entre qualquer par de vértices; e $A_{ij} = A_{ji}$.

Um **dígrafo** é um grafo direcionado, onde as arestas são chamadas de **arcos**, pois possuem direção, diferentemente das arestas. A Figura 2 apresenta um exemplo de um dígrafo simples D = (V, A), onde o conjunto de vértices V contém 5 vértices e o conjunto de arcos A contém 7 arcos:

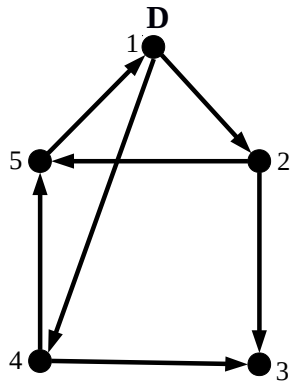

Figura 2: Exemplo de um dígrafo simples D com 5 vértices e 7 arestas.

Um dígrafo D também pode ser modelado no computador por uma **Matriz de Adjacência** A(D), onde as linhas e as colunas de A(D) representam os vértices de D e $A_{ij} = 1$, se o vértice i está ligado ao vértice j (ou seja, se o arco $(i, j) \in A$), e $A_{ij} = 0$, caso contrário. A Matriz de Adjacência A(D) do grafo D da Figura 2 é apresentada abaixo:

A(D) =

0	1	0	1	0
0	0	1	0	1
0	0	0	0	0
0	0	1	0	1
1	0	0	0	0

28) Faça um algoritmo para ler pelo teclado a matriz de adjacência A(D) de um dígrafo simples D com 20 vértices e preencher a matriz de adjacência R(D') de um dígrafo simples D', onde D' possui os **arcos reversos** de D. Ou seja, D' possui o arco (*i*,*j*) se, e somente se, o arco (*j*,*i*) pertence a D. Crie e utilize dois **procedimentos**: um para ler a matriz de adjacência A(D) pelo teclado e outro para preencher a matriz de adjacência R(D').

29) Faça um algoritmo para ler pelo teclado a matriz de adjacência A(D) de um dígrafo simples D com 10 vértices e imprimir na tela **todos** os **ciclos elementares** de comprimento 3 presentes em D. Crie e utilize dois **procedimentos**: um para ler a matriz de adjacência A(D) pelo teclado e outro para imprimir na tela o que se pede.
Obs.: Exemplos de ciclos elementares de comprimento 3 (representados pela sequência de vértices) presentes no grafo D da Figura 2: 1-2-5-1; 2-5-1-2; 5-1-2-5.

30) Considere as matrizes A $=\left[a_{ij}\right]_{n\times m}$ e B $=\left[b_{ij}\right]_{m\times n}$, onde n = 4 e m = 5, ambas com números inteiros gerados aleatoriamente de 1 até 30. Faça um algoritmo para gerar as matrizes A e B, fazendo, em seguida, a verificação da seguinte condição:

$$\min_{1\le j\le m}\sum_{i=1}^{n}|a_{ij}| \quad \le \quad \max_{1\le i\le m}\prod_{j=1}^{n} b_{ij}$$

Crie e utilize um **procedimento** para gerar a matriz e uma **função** para retornar 1, se a condição for satisfeita, ou retornar 0, caso contrário. De acordo com o retorno da função de verificação, deve-se imprimir na função *main*: "Condicao Satisfeita" ou "Condicao Nao Satisfeita".