



Lista de Exercícios 06 – Estruturas de Dados Homogêneas - Matrizes

- 1) Criar um programa que leia os elementos de uma matriz inteira 10 x 10 e escreva os elementos da diagonal principal.

```
N = 10;
for I = 1 : N
    for J = 1 : N
        M(I,J) = input(sprintf('Digite o elemento (%dx%d) da matriz: ',I,J));
    end
end
for I = 1 : N
    for J = 1 : N
        if ( I == J )
            fprintf(1, '%d\t', M(I,J));
        else
            fprintf(1, '\t');
        end
    end
    fprintf(1, '\n');
end
```

- 2) Criar um programa que leia os elementos de uma matriz inteira 10 x 10 e escreva todos os elementos, exceto os elementos da diagonal principal.

```
N = 10;
for I = 1 : N
    for J = 1 : N
        MAT(I,J) = input(sprintf('Digite o elemento (%dx%d ) da matriz: ',I,J));
    end
end
disp('Matriz sem elementos da diagonal principal: ');
for I = 1 : N
    for J = 1 : N
        if ( I ~= J )
            fprintf(1, '%d\t ', MAT(I,J));
        else
            fprintf(1, '\t');
        end
    end
    fprintf(1, '\n');
end
```



- 3) Criar um programa que leia os elementos de uma matriz inteira 10 x 10 e escreva somente os elementos acima da diagonal principal.

```
N = 10;
for I = 1 : N
    for J = 1 : N
        M(I,J) = input(sprintf('Digite o elemento (%dx%d) da matriz: ',I,J));
    end
end
disp('Elementos acima da diagonal principal:');
for I = 1 : N
    for J = 1 : N
        if ( I < J )
            fprintf(1, '%d\t', M(I,J));
        else
            fprintf(1, '\t');
        end
    end
    fprintf(1, '\n');
end
```

- 4) Criar um programa que leia os elementos de uma matriz inteira 10 x 10 e imprima a soma dos elementos que estão acima da diagonal principal:

```
N = 10;
for I = 1 : N
    for J = 1 : N
        M(I,J) = input(sprintf('Digite o elemento (%dx%d) da matriz: ',I,J));
    end
end
S = 0;
for I = 1 : N
    for J = 1 : N
        if ( I < J )
            S = S + M(I,J);
        end
    end
end
fprintf(1, 'A soma eh: %d\n', S);
```

- 5) Criar um programa que leia os elementos de uma matriz inteira 10 x 10 e escreva somente os elementos abaixo da diagonal principal.

```
N = 10;
for I = 1 : N
    for J = 1 : N
        M(I,J) = input(sprintf('Digite o elemento (%dx%d) da matriz: ',I,J));
    end
end
disp('Elementos abaixo da Diagonal Principal');
for I = 2 : N
    for J = 1 : I-1
        fprintf(1, '%d\t', M(I,J));
    end
    fprintf(1, '\n');
end
```



- 6) Criar um programa que leia os elementos de uma matriz inteira 10 x 10 e imprima o produto dos elementos que estão abaixo da diagonal principal.

```
N = 10;
for I = 1 : N
    for J = 1 : N
        M(I,J) = input(sprintf('Digite o elemnto (%dx%d): ',I,J));
    end
end
P = 1;
for I = 1 : N
    for J = 1 : N
        if ( I > J )
            P = P * M(I,J);
        end
    end
end
disp(P);
```

- 7) Criar um programa que leia os elementos de uma matriz inteira 10 x 10 e escreva os elementos da diagonal secundária.

```
N = 10;
for I = 1 : N
    for J = 1 : N
        M(I,J)=input(sprintf('Digite o elemento (%dx%d) da matriz: ',I,J));
    end
end
disp('Elementos da Diagonal Secundaria:');
for I = 1 : N
    for J = 1 : N
        if ( J == (N - I + 1) )
            fprintf(1, '%d\t', M(I,J));
        else
            fprintf(1, '\t');
        end
    end
    fprintf(1, '\n');
end
```

- 8) Criar um programa que leia os elementos de uma matriz inteira 10 x 10 e escreva todos os elementos exceto os elementos da diagonal secundária.

```
N = 10;
for I = 1 : N
    for J = 1 : N
        M(I,J) = input(sprintf('Digite o elemento (%dx%d) da matriz: ',I,J));
    end
end
disp('Elementos acima da diagonal principal');
for I = 1 : N
    for J = 1 : N
        if ( I ~= N - J + 1 )
            fprintf(1, '%d\t', M(I,J));
        else
            fprintf(1, '\t');
        end
    end
    fprintf(1, '\n');
end
```

- 9) Criar um programa que leia os elementos de uma matriz inteira 10 x 10 e escreva somente os elementos acima da diagonal secundária.

```
N = 10;
for I = 1 : N
    for J = 1 : N
        M(I,J) = input(sprintf('Digite o elemento (%dx%d) da matriz: ',I,J));
    end
end
disp('Elementos acima da diagonal secundaria');
for I = 1 : N
    for J = 1 : N
        if ( I < N - J + 1)
            fprintf(1, '%d\t', M(I,J));
        else
            fprintf(1, '\t');
        end
    end
    fprintf(1, '\n');
end
```

- 10) Criar um programa que leia os elementos de uma matriz inteira 10 x 10 e escreva somente os elementos abaixo da diagonal secundária.

```
N = 10;
for I = 1 : N
    for J = 1 : N
        M(I,J) = input(sprintf('Digite o elemento (%dx%d) da matriz: ',I,J));
    end
end
disp('Elementos abaixo da Diagonal Secundaria');
for I = 2 : N
    for J = 1 : N-I+1
        fprintf(1, '\t');
    end
    for J = N-I+2 : N
        fprintf(1, '%d\t', M(I,J));
    end
    fprintf(1, '\n');
end
```

- 11) Entrar com valores para uma matriz $A_{3 \times 4}$. Gerar e imprimir uma matriz B que é o triplo da matriz A.

```
for I = 1 : 3
    for J = 1 : 4
        MA(I,J) = input(sprintf('Digite o elemento (%dx%d) da matriz: ',I,J));
    end
end
for I = 1 : 3
    for J = 1 : 4
        MB(I,J) = 3 * MA(I,J);
    end
end
for I = 1 : 3
    for J = 1 : 4
        fprintf(1, '%f\t', MB(I,J));
    end
    fprintf(1, '\n');
end
```



12) Entrar com valores inteiros para uma matriz $A_{4 \times 4}$ e para uma matriz $B_{4 \times 4}$. Gerar e imprimir a SOMA ($A+B$).

```
N = 4;
for I = 1 : N
    for J = 1 : N
        MA(I,J) = input(sprintf('Digite o elemento (%dx%d) da matriz A: ', I,J));
    end
end
for I = 1 : N
    for J = 1 : N
        MB(I,J) = input(sprintf('Digite o elemento (%dx%d) da matriz B: ', I,J));
    end
end
for I = 1 : N
    for J = 1 : N
        MS(I,J) = MA(I,J) + MB(I,J);
    end
end
disp('Matriz Soma: ');
for I = 1 : N
    for J = 1 : N
        fprintf(1, '%d\t', MS(I,J));
    end
    fprintf(1, '\n');
end
```

13) Entrar com valores para duas matrizes inteiras de ordem cinco. Gerar e imprimir a matriz diferença.

```
N = 5;
for I = 1 : N
    for J = 1 : N
        MA = input(sprintf('Digite o elemento (%dx%d) da matriz A: ', I,J));
    end
end
for I = 1 : N
    for J = 1 : N
        MB = input(sprintf('Digite o elemento (%dx%d) da matriz B: ', I,J));
    end
end
for I = 1 : N
    for J = 1 : N
        MD(I,J) = MA(I,J) - MB(I,J);
    end
end
disp('A matriz diferença é:');
for I = 1 : N
    for J = 1 : N
        fprintf(1, '%d\t', MD(I,J));
    end
    fprintf(1, '\n');
end
```



- 14) Ler uma matriz 4x5 de inteiros, calcular e imprimir a soma de todos os seus elementos.

```
for I = 1 : 4
    for J = 1 : 5
        M(I,J) = input(sprintf('(%dx%d): ',I,J));
    end
end
for I = 1 : 4
    for J = 1 : 5
        S = S + M(I,J);
    end
end
disp(S);
```

- 15) Ler valores inteiros para a matriz $A_{3 \times 5}$. Gerar e imprimir a matriz (vetor) SL (soma das 3 linhas), onde cada elemento é a soma dos elementos de uma linha da matriz A. Faça o trecho que gera a matriz SL separado (laços de repetição) da entrada e da saída de dados.

```
% entrada de dados
for I = 1 : 3
    for J = 1 : 5
        MA(I,J) = input(sprintf('Digite Elemento (%dx%d) da matriz: ',I,J));
    end;
end
% calculo de SL
for I = 1 : 3
    SL(I) = 0;
    for J = 1 : 5
        SL(I) = SL(I) + MA(I,J);
    end
end
% saida de dados
for I = 1 : 3
    fprintf(1,'Soma da %da. linha: %d\n',I,SL(I));
end
```



- 16) Uma floricultura conhecedora de sua clientela gostaria de fazer um programa que pudesse controlar sempre um estoque mínimo de determinadas plantas, pois todo dias, pela manhã, o dono faz novas aquisições. Criar um programa que deixe cadastrar 50 tipos de plantas e nunca deixar o estoque ficar abaixo do ideal. Para cada planta, o dono gostaria de cadastrar o nome, o estoque ideal e a quantidade em estoque. Dessa forma o algoritmo pode calcular a quantidade que o dono da loja precisa comprar no próximo dia. Essa quantidade a ser comprada deve ser impressa (quando maior que zero) como uma lista para o dono da floricultura.

```
for I = 1 : 50
    TP{I} = input('Planta: ', 's');
    M(I,1) = input('Quantidade ideal para estoque: ');
    M(I,2) = input('Quantidade atual no estoque: ');
    M(I,3) = M(I,2) - M(I,1);
end
for I = 1 : 50
    if ( M(I,3) < 0 )
        fprintf(1, '%s comprar %d\n', TP{I}, -M(I,3));
    end
end
```



- 17) A gerente do cabeleireiro Sempre Bela tem uma tabela em que registra os “pés” as “mãos” e o serviço de podologia das cinco manicures. Sabendo-se que cada uma ganha 50% do que faturou ao mês, criar um programa que possa calcular e imprimir quanto cada um vai receber, uma vez que não têm carteiras assinadas; os valores, respectivamente, são R\$ 10,00; R\$ 15,00 e R\$ 30,00.

```
for M = 1 : 5
    fprintf(1,'Manicure: %d\n',M);
    for S = 1 : 3
        if ( S == 1 )
            MAT(M,S) = input('Digite o numero de pes feitos: ');
        end
        if ( S == 2 )
            MAT(M,S) = input('Digite o numero de maos feitas: ');
        end
        if ( S == 3 )
            MAT(M,S) = input('Digite o numero de servicos de podologia feitos: ');
        end
    end
end
for M = 1 : 5
    for S = 1 : 3
        if ( S == 1 )
            MAT(M,S) = MAT(M,S)*10;
        end
        if ( S == 2 )
            MAT(M,S) = MAT(M,S)*15;
        end
        if ( S == 3 )
            MAT(M,S) = MAT(M,S)*30;
        end
    end
end
for M = 1 : 5
    SAL(M) = 0;
end
for M = 1 : 5
    for S = 1 : 3
        SAL(M) = MAT(M,S) / 2 + SAL(M);
    end
end
disp('Salarios das Manicures:');
for M = 1 : 5
    fprintf(1,'Manicure %d: %d\n',M,SAL(M));
end
```


18) A matriz dados contém na 1ª coluna a matrícula do aluno no curso; na 2ª, o sexo (0 para feminino e 1 para masculino); na 3ª, o código do curso, e na 4ª, o CR (Coeficiente de Rendimento). Suponha 10 alunos e que o CR é um número inteiro.

Faça um programa que armazene esses dados sabendo-se que:

- O código do curso é uma parte de um número de matrícula: aascccnnn (aa ano, s semestre, ccc código do curso e nnn matrícula no curso), que deve ser lido; Além, disso, o sexo e o CR devem ser lidos também.

Um grupo empresarial resolveu premiar a aluna com CR mais alto de um curso cujo código deverá ser digitado.

```
N = 10;
for C = 1 : N
    COD = input('Digite o codigo do aluno: ');
    MAT(C,1) = mod(COD,1000); % matricula no curso
    MAT(C,3) = mod(floor(COD/1000),10000); % codigo do curso
    MAT(C,2) = input('Digite o sexo do aluno (0=F,1=M): '); % sexo
    MAT(C,4) = input('Digite o Coeficiente de Rendimento do Aluno: '); % CR
end
% ordenando
for I = 1 : N-1
    K = I;
    for J = I+1 : N
        if ( MAT(K,4) < MAT(J,4) )
            K = J;
        end
    end
    AUX = MAT(I,1); MAT(I,1) = MAT(K,1); MAT(K,1) = AUX;
    AUX = MAT(I,2); MAT(I,2) = MAT(K,2); MAT(K,2) = AUX;
    AUX = MAT(I,3); MAT(I,3) = MAT(K,3); MAT(K,3) = AUX;
    AUX = MAT(I,4); MAT(I,4) = MAT(K,4); MAT(K,4) = AUX;
end
COD = input('Qual o codigo do curso: ');
for C = 1 : N
    if ( COD == MAT(C,3) ) & ( MAT(C,2) == 0 )
        fprintf(1, '%d %d\n', MAT(C,1), MAT(C,4));
        break;
    end
end
```



19) Criar um programa que possa armazenar as alturas de dez atletas de cinco delegações que participarão dos jogos de verão. Imprimir a maior altura de cada delegação.

```
for I = 1 : 5
    for J = 1 : 10
        M(I,J) = input(sprintf('Digite a altura do %do. atleta da %da. delegacao: ',J,I));
    end
end
for I = 1 : 5
    MAIOR(I) = M(I,1);
    for J = 2 : 10
        if ( M(I,J) > MAIOR )
            MAIOR(I) = M(I,J);
        end
    end
end
for I = 1 : 5
    fprintf('A maior altura da %da. delegacao eh: %f\n',I,MAIOR(I));
end
```

20) Criar um programa que carregue uma matriz 12 x 4 com os valores das vendas de uma loja, em que cada linha represente um mês do ano, e cada coluna, uma semana do mês. Para fins de simplificação considere que cada mês possui somente 4 semanas. Calcule e imprima:

- Total vendido em cada mês do ano;
- Total vendido em cada semana durante todo o ano;
- Total vendido no ano.

```
for I = 1 : 12
    for J = 1 : 4
        MAT(I,J) = input(sprintf('Mes - %d, Semana - %d: ',I,J));
    end
end
TOTANO = 0;
for I = 1 : 12
    TOTMES = 0;
    for J = 1 : 4
        TOTMES = TOTMES + MAT(I,J);
    end
    fprintf(1,'Total do mes %d: %3.2f\n',I,TOTMES);
    TOTANO = TOTANO + TOTMES;
end
for J = 1 : 4
    TOTSEM = 0;
    for I = 1 : 12
        TOTSEM = TOTSEM + MAT(I,J);
    end
    fprintf(1,'Total da semana %d: %3.2f\n',J,TOTSEM);
end
fprintf(1,'Total do ano: %3.2f\n',TOTANO);
```



- 21) Criar um programa que entre com valores inteiros para uma matriz m 3 x 3 e imprima a matriz final, conforme mostrado a seguir:

$$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix} \text{ a matriz gira } 90^\circ \begin{bmatrix} 7 & 4 & 1 \\ 8 & 5 & 2 \\ 9 & 6 & 3 \end{bmatrix}$$

```
N = 3;
for I = 1 : N
    for J = 1 : N
        M(I,J) = input(sprintf('Digite o o elemento (%dx%d) da matriz: ',I,J));
    end
end
for I = 1 : N
    for J = 1 : N
        F(J,N-I+1) = M(I,J);
    end
end
for I = 1 : N
    for J = 1 : N
        fprintf(1, '%d\t', F(I,J));
    end
    fprintf(1, '\n');
end
```

- 22) Criar um programa que entre com valores inteiros para uma matriz m 3 x 3 e imprima a matriz final, conforme mostrado a seguir:

$$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix} \text{ a matriz gira } 180^\circ \begin{bmatrix} 9 & 8 & 7 \\ 6 & 5 & 4 \\ 3 & 2 & 1 \end{bmatrix}$$

```
N = 3;
for I = 1 : N
    for J = 1 : N
        M(I,J) = input(sprintf('Digite o elemento (%dx%d) da matriz: ',I,J));
    end
end
for I = 1 : N
    for J = 1 : N
        F(N-I+1,N-J+1) = M(I,J);
    end
end
for I = 1 : N
    for J = 1 : N
        fprintf(1, '%d\t', F(I,J));
    end
    fprintf(1, '\n');
end
```

- 23) Criar um programa que entre com valores inteiros para uma matriz m 3 x 3 e imprima a matriz final, conforme mostrado a seguir:

$$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix} \text{ a matriz gira } 270^\circ \begin{bmatrix} 3 & 6 & 9 \\ 2 & 5 & 8 \\ 1 & 4 & 7 \end{bmatrix}$$

```
N = 3;
for I = 1 : N
    for J = 1 : N
        M(I,J) = input(sprintf('Digite o elemento (%d,%d) da matriz: ',I,J));
    end
end
for I = 1 : N
    for J = 1 : N
        F(N-J+1,I) = M(I,J);
    end
end
for I = 1 : N
    for J = 1 : N
        fprintf(1, '%d\t', F(I,J));
    end
    fprintf(1, '\n');
end
```



24) Criar um programa que leia e armazene os elementos de uma matriz inteira $M_{10 \times 10}$ e imprimi-la. Troque, na ordem a seguir:

- a segunda linha pela oitava linha;
- a quarta coluna pela décima coluna;
- a diagonal principal pela diagonal secundária.

```
N = 10;
for I = 1 : N
    for J = 1 : N
        M(I,J) = input(sprintf('(%dx%d): ', I, J));
    end
end
for J = 1 : N
    AUX = M(8, J);
    M(8, J) = M(2, J);
    M(2, J) = AUX;
end
for I = 1 : N
    AUX = M(I, 10);
    M(I, 10) = M(I, 4);
    M(I, 4) = AUX;
end
for I = 1 : N
    AUX = M(I, I);
    M(I, I) = M(I, N-I+1);
    M(I, N-I+1) = AUX;
end
for I = 1 : N
    for J = 1 : N
        fprintf(1, '%d\t', M(I, J));
    end
    fprintf(1, '\n');
end
```

25) Criar um programa que leia valores para uma matriz $M_{2 \times 2}$. Calcular e imprimir o determinante. Para cálculo do determinante de uma matriz de ordem 2, é simplesmente computar a diferença entre os produtos das diagonais principal e secundária, respectivamente.

```
for I = 1 : 2
    for J = 1 : 2
        M(I,J) = input(sprintf('Elemento (%d,%d): ', I, J));
    end
end
DET = M(1,1) * M(2,2) - M(1,2) * M(2,1);
fprintf(1, 'Determinante: %5.4f\n', DET);
```



26) Criar um programa que leia uma matriz $A_{N \times N}$ ($N \leq 10$) e calcule a respectiva matriz transposta A^t .

```
N=input('Entre com a ordem da matriz: ');
for I = 1 : N
    for J = 1 : N
        A(I,J) = input(sprintf('Entre com o elemento %dx%d: \n',I,J));
    end
end
for I = 1 : N
    for J = 1 : N
        AT(J,I) = A(I,J);
        fprintf(1, '%.2f\t', AT(I,J));
    end
    fprintf(1, '\n');
end
```

27) Criar um programa que leia uma matriz $A_{N \times N}$ ($N \leq 10$) e verifique (informe) se tal matriz é ou não simétrica ($A^t = A$).

```
N = input('Digite o numero de linhas da matriz: ');
for I = 1 : N
    for J = 1 : N
        M(I,J) = input(sprintf('Digite o elemento (%dx%d) da matriz: ',I,J));
    end
end
SIT = 1;
for I = 1 : N
    for J = I : N
        if ( M(I,J) ~= M(J,I) )
            SIT = 0;
        end
    end
end
if ( SIT = 1 )
    disp('A matriz eh simetrica!');
else
    disp('A matriz nao eh simetrica!');
end
```



28) Criar um programa que leia uma matriz $A_{N \times N}$ ($N \leq 10$) e verifique (informe) se tal matriz é ou não anti-simétrica ($A^t = -A$).

```
N = input('Qual o número de colunas da Matriz A (NxN): ');
for I = 1 : N
    for J = 1 : N
        A(I,J) = input(sprintf('Digite o elemento (%d,%d) da matriz: ',I,J));
    end
end
for I = 1 : N
    for J = 1 : N
        AT(I,J) = A(J,I);
        AS(I,J) = -A(I,J);
    end
end
SIT = 1;
for I = 1 : N
    for J = 1 : N
        if ( AT(I,J) ~= AS(I,J) )
            SIT = 0;
        end
    end
end
if ( SIT == 1 )
    disp('A matriz A é anti-simétrica!');
else
    disp('A matriz A não é anti-simétrica!');
end
```



29) Criar um programa que leia uma matriz $A_{2 \times 2}$ e calcule a respectiva inversa A^{-1} .

```
for I = 1 : 2
    for J = 1 : 2
        A(I,J) = input(sprintf('(%dx%d): ',I,J));
    end
end
DET = A(1,1) * A(2,2) - A(1,2) * A(2,1);
fprintf(1,'O determinante eh: %f\n',DET);
if ( DET ~= 0 )
    B(1,1) = A(2,2);
    B(2,2) = A(1,1);
    B(1,2) = (-1) * A(1,2);
    B(2,1) = (-1) * A(2,1);
    % calculo
    for I = 1 : 2
        for J = 1 : 2
            B(I,J) = (1/DET) * B(I,J);
        end
    end
    disp('A inversa eh');
    for I = 1 : 2
        for J = 1 : 2
            fprintf(1, '%.2f\t', B(I,J));
        end
        fprintf(1, '\n');
    end
else
    disp('A matriz nao eh inversivel');
end
```


- 30) Criar um programa que receba duas matrizes $A_{C \times D}$ e $B_{E \times F}$ (C, D, E e $F \leq 6$). Esse algoritmo deve verificar se o produto matricial de A por B é possível ($D = E$). Caso seja possível, calcular o tal produto, imprimindo a matriz $G_{C \times F}$ resultado.

```
C = input('Qual o número de linhas da Matriz A (c): ');
D = input('Qual o número de colunas da Matriz A (d): ');
E = input('Qual o número de linhas da Matriz B (e): ');
F = input('Qual o número de colunas da Matriz B (f): ');
% entrada de dados - matriz A
for I = 1 : C
    for J = 1 : D
        A(I,J) = input(sprintf('Elemento (%d,%d): ',I,J));
    end
end
% entrada de dados - matriz B
for I = 1 : E
    for J = 1 : F
        B(I,J) = input(sprintf('Elemento (%d,%d): ',I,J));
    end
end
if ( D == E )
    % calculo do produto matricial
    for I = 1 : C
        for J = 1 : F
            ELEM = 0;
            for K = 1 : D
                ELEM = ELEM + A(I,K) * B(K,J);
            end
            G(I,J) = ELEM;
        end
    end
    % saida de dados - matriz G
    for I = 1 : C
        for J = 1 : F
            fprintf(1, '%7.4f\t', G(I,J));
        end
        fprintf(1, '\n');
    end
else
    disp('Impossivel calcular o produto matricial entre A e B!');
    disp('O numero de colunas de A (d) deve ser igual ao numero de linhas de B (e)!');
end
```