## Lista de Exercícios 05 – Estruturas de Dados Homogêneas - Matrizes

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

```
algoritmo L5P05;
constante
  N = 10;
var
  inteiro: I, J, MAT[1..N,1..N];
início
  para I de 1 até N faça
    para J de 1 até N faça
      imprima("Digite Elemento (",I,"x",J,") da matriz: ");
      leia(MAT[I,J]);
    fim-para
  fim-para
  imprima("Elementos abaixo da Diagonal Principal");
  para I de 2 até N faça
    para J de 1 até I-1 faça
      imprima("Elemento (",I,"x",J,"): ",MAT[I,J]);
    fim-para
  fim-para
fim
```

```
program l5p05;
const
  N = 10;
var
  I, J: integer;
  MAT: array [1..N,1..N] of integer;
begin
  for I := 1 to N do
    for J := 1 to N do
    begin
      write('Digite Elemento (',I,'x',J,') da matriz: ');
      readLn(MAT[I,J]);
    end;
  writeLn('Elementos abaixo da Diagonal Principal');
  for I := 2 to N do
  begin
    for J := 1 to I-1 do
      write(MAT[I,J],chr(9));
    writeLn('');
  end;
end.
```

```
N = 10;
for I = 1 : N
  for J = 1 : N
    MAT(I,J) = input(sprintf('Digite 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',MAT(I,J));
  end
  fprintf(1,'\n');
end
```

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

```
algoritmo L5P10;
constante
  N = 10;
var
  inteiro: I, J, MAT[1..N,1..N];
início
  para I de 1 até N faça
    para J de 1 até N faça
      imprima("Digite Elemento (",I,"x",J,") da matriz: ");
      leia(MAT[I,J]);
    fim-para
  fim-para
  imprima("Elementos abaixo da Diagonal Secundaria");
  para I de 2 até N faça
    para J de N-I+2 até N faça
      imprima("Elemento (",I,"x",J,"): ",MAT[I,J]);
    fim-para
  fim-para
fim
```

```
program l5p10;
const
  N = 10;
var
  I, J: integer;
  MAT: array [1..N,1..N] of integer;
begin
  for I := 1 to N do
    for J := 1 to N do
    begin
      write('Digite Elemento (',I,'x',J,') da matriz: ');
      readLn(MAT[I,J]);
    end;
  writeLn('Elementos abaixo da Diagonal Secundaria');
  for I := 2 to N do
  begin
    for J := 1 to N-I+1 do
      write(chr(9));
    for J := N-I+2 to N do
      write(MAT[I,J],chr(9));
    writeLn('');
  end;
end.
```

```
N = 10;
for I = 1 : N
  for J = 1 : N
    MAT(I,J) = input(sprintf('Digite 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',MAT(I,J));
  end
  fprintf(1,'\n');
end
```

15) Ler valores inteiros para a matriz $A_{3x5}$. 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.

```
algoritmo L5P15;
var
  inteiro: I, J, A[1..3,1..5],SL[1..3];
início
  { entrada de dados }
  para I de 1 até 3 faça
    para J de 1 até 5 faça
      imprima("Digite Elemento (",I,"x",J,") da matriz: ");
      leia(A[I,J]);
    fim-para
  fim-para
  { cálculo de SL }
  para I de 1 até 3 faça
    SL[I] <- 0;
    para J de 1 até 5 faça
      SL[I] <- SL[I] + A[I,J];
    fim-para
  fim-para
  { saída de dados }
  para I de 1 até 3 faça
    imprima("Soma da ",I,"a. linha: ",SL[I]);
  fim-para
fim
```

```
program l5p15;
var
  I, J: integer;
  A: array [1..3,1..5] of integer;
  SL: array [1..3] of integer;
begin
  { entrada de dados }
  for I := 1 to 3 do
    for J := 1 to 5 do
    begin
      write('Digite Elemento (',I,'x',J,') da matriz: ');
      readLn(A[I,J]);
    end;
  { calculo de SL }
  for I := 1 to 3 do
  begin
    SL[I] := 0;
    for J := 1 to 5 do
      SL[I] := SL[I] + A[I,J];
  end;
  { saida de dados }
  for I := 1 to 3 do
    writeLn('Soma da ',I,'a. linha: ',SL[I]);
end.
```

```
% entrada de dados
for I = 1 : 3
  for J = 1 : 5
    A(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) + A(I,J);
  end
end
% saida de dados
for I = 1 : 3
  fprintf(1,'Soma da %da. linha: %d\n',I,SL(I));
end
```

20) Criar um algoritmo 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.

```
algoritmo L5P20;
var
  inteiro: I, J;
  real:    TOTANO, TOTMES, TOTSEM, MAT[1..12,1..4];
início
  para I de 1 até 12 faça
    para J de 1 até 4 faça
      imprima("Mes - ",I,", Semana - ",J,": ");
      leia(MAT[I,J]);
    fim-para
  fim-para
  TOTANO <- 0;
  para I de 1 até 12 faça
    TOTMES <- 0;
    para J de 1 até 4 faça
      TOTMES <- TOTMES + MAT[I,J];
    fim-para
    imprima("Total do mes ",I,": ",TOTMES);
    TOTANO <- TOTANO + TOTMES;
  fim-para
  para J de 1 até 4 faça
    TOTSEM <- 0;
    para I de 1 até 12 faça
      TOTSEM <- TOTSEM + MAT[I,J];
    fim-para
    imprima("Total da semana ",J,": ",TOTSEM);
  fim-para
  imprima("Total do ano: ",TOTANO);
fim
```

```
program l5p20;
var
  I, J: integer;
  TOTANO, TOTMES, TOTSEM: real;
  MAT: array [1..12,1..4] of real;
begin
  for I := 1 to 12 do
    for J := 1 to 4 do
    begin
      write('Mes - ',I,', Semana - ',J,': ');
      readLn(MAT[I,J]);
    end;
  TOTANO := 0;
  for I := 1 to 12 do
  begin
    TOTMES := 0;
    for J := 1 to 4 do
      TOTMES := TOTMES + MAT[I,J];
    writeLn('Total do mes ',I,': ',TOTMES:3:2);
    TOTANO := TOTANO + TOTMES;
  end;
  for J := 1 to 4 do
  begin
    TOTSEM := 0;
    for I := 1 to 12 do
      TOTSEM := TOTSEM + MAT[I,J];
    writeLn('Total da semana ',J,': ',TOTSEM:3:2);
  end;
  writeLn('Total do ano: ',TOTANO:3:2);
end.
```

```
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);
```

25) Criar um algoritmo que leia valores para uma matriz $M_{2x2}$. 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.

```
algoritmo L5P25;
var
  inteiro: I, J;
  real:    DET, M[1..2,1..2];
início
  para I de 1 até 2 faça
    para J de 1 até 2 faça
      imprima("Elemento (",I,",",J,"): ");
      leia(M[I,J]);
    fim-para
  fim-para
  DET <- M[1,1] * M[2,2] - M[1,2] * M[2,1];
  imprima("Determinante: ",DET);
fim
```

```
program l5p25;
var
  I, J: integer;
  DET: real;
  M: array [1..2,1..2] of real;
begin
  for I := 1 to 2 do
    for J := 1 to 2 do
    begin
      write('Elemento (',I,',',J,'): ');
      readLn(M[I,J]);
    end;
  DET := M[1,1] * M[2,2] - M[1,2] * M[2,1];
  writeLn('Determinante: ',DET:5:4);
end.
```

```
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);
```

30) Criar um algoritmo que receba duas matrizes $A_{CxD}$ e $B_{ExF}$ (C, D, E e F ≤ 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 $G_{CxF}$ resultado.

```
algoritmo L5P30;
var
  inteiro: I, J, K;
  inteiro: C, D, E, F;
  real:    ELEM, A[1..6,1..6], B[1..6,1..6], G[1..6,1..6];
início
  imprima("Qual a ordem da Matriz A (c x d): ");
  leia(C,D);
  imprima("Qual a ordem da Matriz B (e x f): ");
  leia(E,F);
  { entrada de dados - matriz A }
  para I de 1 até C faça
    para J de 1 até D faça
      imprima("Elemento (",I,",",J,"): ");
      leia(A[I,J]);
    fim-para
  fim-para
  { entrada de dados - matriz B }
  para I de 1 até E faça
    para J de 1 até F faça
      imprima("Elemento (",I,",",J,"): ");
      leia(B[I,J]);
    fim-para
  fim-para
  se ( D = E ) então
    { calculo do produto matricial }
    para I de 1 até C faça
      para J de 1 até F faça
        ELEM <- 0;
        para K de 1 até D faça
          ELEM <- ELEM + A[I,K] * B[K,J];
        fim-para
        G[I,J] <- ELEM;
      fim-para
    fim-para
    { saida de dados - matriz G }
    para I de 1 até C faça
      para J de 1 até F faça
        imprima("Elemento (",I,",",J,"): ",G[I,J]);
      fim-para
    fim-para
  senão
    imprima("Impossível calcular o produto matricial entre A e B!");
    imprima("O número de colunas de A (d) deve ser igual ao número de linhas de B (e)!");
  fim-se
fim
```

```
program l5p30;
var
  I, J, K: integer;
  C, D, E, F: integer;
  ELEM: real;
  A, B, G: array [1..6,1..6] of real;
begin
  write('Qual a ordem da Matriz A (c x d): ');
  readLn(C,D);
  write('Qual a ordem da Matriz B (e x f): ');
  readLn(E,F);
  { entrada de dados - matriz A }
  for I := 1 to C do
    for J := 1 to D do
    begin
      write('Elemento (',I,',',J,'): ');
      readLn(A[I,J]);
    end;
  { entrada de dados - matriz B }
  for I := 1 to E do
    for J := 1 to F do
    begin
      write('Elemento (',I,',',J,'): ');
      readLn(B[I,J]);
    end;
  if ( D = E ) then
  begin
    { calculo do produto matricial }
    for I := 1 to C do
      for J := 1 to F do
      begin
        ELEM := 0;
        for K := 1 to D do
          ELEM := ELEM + A[I,K] * B[K,J];
        G[I,J] := ELEM;
      end;
    { saida de dados - matriz G }
    for I := 1 to C do
    begin
      for J := 1 to F do
        write(G[I,J]:7:4,chr(9));
      writeLn('');
    end;
  end
  else
  begin
    writeLn('Impossivel calcular o produto matricial entre A e B!');
    writeLn('O numero de colunas de A (d) deve ser igual ao numero de linhas de B (e)!');
  end;
end.
```

```
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
```